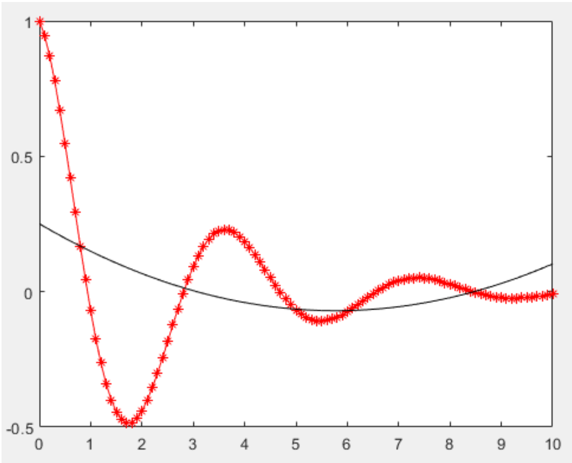
Question 1:

Best fit for f1(t):



Coefficients =
  0.250922262575243
 -0.111345919449174
  0.009651643345421

Question 2:

The only root was found to be =

  0.666664252057672

which is close to 2/3 due to numerical method.

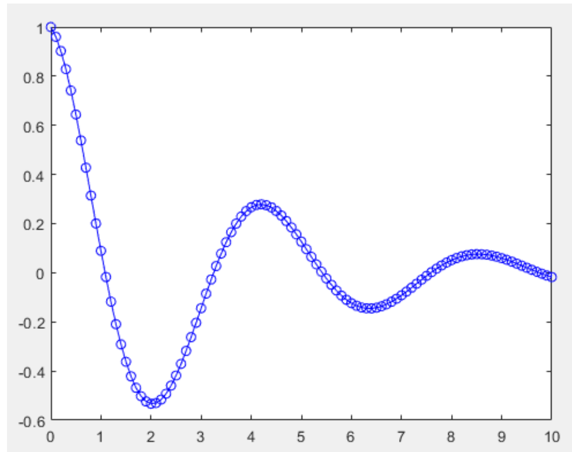
Question 3:

The frequency was found to be =

  0.230777157544303 Hz

using the secant method, the time between two successive periods was found. Then we invert the mean to find the average frequency.

Signal was plotted.



Question 4:

The operating frequency was found to be =

 52.499999999991473 kHz

Which satisfied both equations by taking the derivative of the energy equation and equating it to the power equation.

Appendix for Code:

Code for question 1 and 3 was developed in slides

Question 1:

Signal was copy pasted on MATLAB. Omitted from report for conciseness

```
signal = [
     ];

%first column of signal is t
t = signal(:,1);

%second column of signal is y
y = signal(:,2);

%based on a0 + a1*t + a2*t^2

%matrix formed by concatenating
columns
%column 1 is t power 0 coefficient
col1 = ones(size(t));

%merge the columns to make A
A = [col1, t, (t.^2)];

%forming normal matrix A, transpose
multiplied by A
NormA = A' *A;

%forming normal B, transpose
multiplied by y
NormB = A' *y;

%solving for matrix using inverse of
NormA multiplied by NormB
Coef = inv(NormA)*NormB

Zest= Coef(1) + Coef(2).*t +
Coef(3).*t.^2;
plot(t, y, 'b-o' ,t , Zest, 'k')
```

Question 2:

```
%input a and b such that f(a)*f(b)<0
a  = 0; %lower limit
b = 5; %upper limit

%tolerance e
e = 10^-8;

%bisection algorithm
while abs(b-a) >= e
    %calculate the midpoint
    c = (a+b)/2;
    f_c = c^3 - 2*c^2 + (4/3)*c -
8/27;
    f_a = a^3 - 2*a^2 + (4/3)*a -
8/27;
    if f_c*f_a > 0
       %solution is [c,b]
       a = c;
    else
       %solution is [a,c]
       b = c;
    end
end
%display the root
display(c);
```

Question 3:

Signal was copy pasted on MATLAB. Omitted from report for conciseness

```
signal = [
     ];

 %first column of signal is x
 x = signal(:, 1);

 %second column of signal is y
 y = signal(:, 2);

 %plot the signal
 plot(x, y, 'b-o');

 %based on the graph, 5 roots
 Roots = ones(5,1);

 %secant method algorithm
 Roots(1)  = x(12)-(y(12)*((x(12)-
x(11))/(y(12)-y(11))));
 Roots(2)  = x(34)-(y(34)*((x(34)-
x(33))/(y(34)-y(33))));
 Roots(3)  = x(56)-(y(56)*((x(56)-
x(55))/(y(56)-y(55))));
 Roots(4)  = x(77)-(y(77)*((x(77)-
x(76))/(y(77)-y(76))));
 Roots(5)  = x(99)-(y(99)*((x(99)-
x(98))/(y(99)-y(98))));

 %initialize period data
 Period = ones(2,1);

 Period31 = Roots(3)  - Roots(1);
 Period53 = Roots(5)  - Roots(3);
 Period(1)  = Period31;
 Period(2)  = Period53;

 frequency = 1/mean(Period);
 display(frequency);
```

Question 4:

```
%initial guess arbitrary
x0 = 10;
%tolerance
e = 10^(-8);

f = @(x)  (100 - 100*exp(-0.56*x)  -
2*x + 5);
f_prime = @(x)  56*(exp(-0.56*x))  -
2;

%initialize flag convergence
convergence = false;

%max number of iterations
maxIter = 100;

for n = 1 : maxIter
    y = f(x0);
    y_prime = f_prime(x0);

    %newton's method
    x1 = x0 - y/y_prime;

    if(abs(x1 - x0)  <= e * abs(x1))
        convergence = true;
        break;
    end
    x0 = x1;
end

if(convergence)
    disp(x1);
else
    disp("did not converge");
end
```