ELECTRONIC ENGINEERING DEPARTMENT

MATH 214 NUMERICAL METHODS

2020 – 2021 FALL

PROJECT 3

Student ID :1901022039

Name Surname :Alperen Arslan

Preperation date: 16.12.2020

Upload date: 16.12.2020

1. Problem Definition and Formulation

The problem is to first determine the power values according to given measured data set and then determining the stored energy by time with using composite midpoint integration rule, composite trapezoidal integration rule, composite Simpson's rule. After this, the stored energy by using voltage-current relation must be determined and methods must be compared between each other.

The formula of the power in inductor is written as follows:

$$p(t) = v(t)i(t) \quad (1)$$

The stored energy can be found as follows:

$$w(t) = \int_0^t p(t)dt = \int_0^t v(t)i(t)dt \quad (2)$$

Using the voltage-current relation, the stored energy can also be found as:

$$w(t) = \frac{1}{2}Li^2(t) \quad (3)$$

We can use three different methods to calculate the integral given in equation (2),

i. Composite Midpoint Integration rule

For applying midpoint rule, basically a rectangle area under the curve that wanted to integrate must be chosen. Then, the multiplication of the midpoint of the rectangle and the width of the rectangle will give the area under the curve.

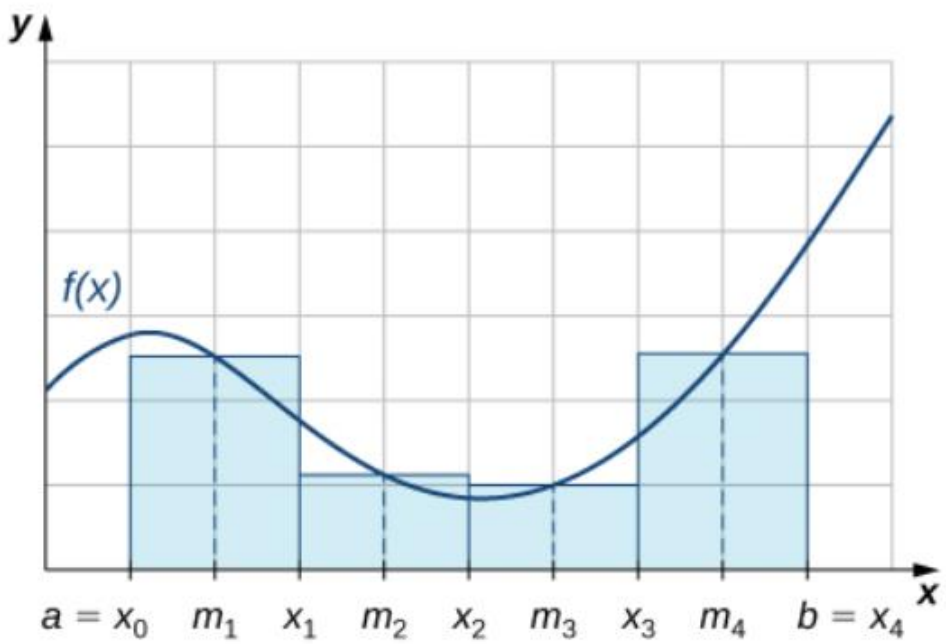

Figure 1. composite midpoint rule

For example, in Figure 1, the integration between x_0 to x_1 is:

$$\int_{x_0}^{x_1} f(x)dx = f(m_1)(x_1 - x_0)$$

And the integration between x_0 to x_4 is:

$$\int_{x_0}^{x_4} f(x)dx = f(m_1)(x_1 - x_0) + f(m_2)(x_2 - x_1) + f(m_3)(x_3 - x_2) + f(m_4)(x_4 - x_3)$$

The general formula of the composite midpoint integration rule is:

$$\int_a^b f(x) = \sum_{i=2k}^{n} f(m_i)(x_i - x_{i-1}) \ , k = 1,2,3 \ldots \tag{4}$$

Midpoint rule is only works for even number of subintervals.

ii. Composite Trapezoidal Integration Rule

For applying trapezoidal rule, first a trapezoid under the curve must be chosen. And then the area of the trapezoid can be found such as follows:

$$\int_a^b \sum_{i=2}^{b} \frac{[f(i-1) + f(i)]}{2} (\, \Delta x) \, , \Delta x = x_i - x_{i-1} \tag{5}$$

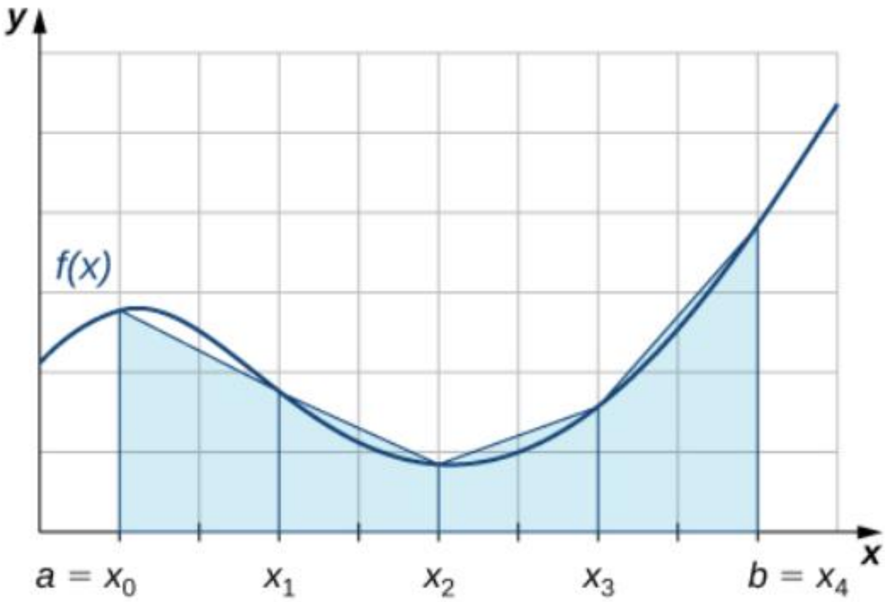

Figure 2. composite trapezoidal rule

For example, in Figure 2, the integral from x_0 to x_1 is:

$$\int_{x_0}^{x_1} f(x)dx = \frac{[f(x_0) + f(x_1)]}{2} (\, \Delta x), \qquad \Delta x = x_1 - x_0$$

iii. Composite Simpson's Rule

Simpson's rule is an algorithmic application of the 1-4-1 quadratic approximation. For applying this method, first subintervals must be defined.

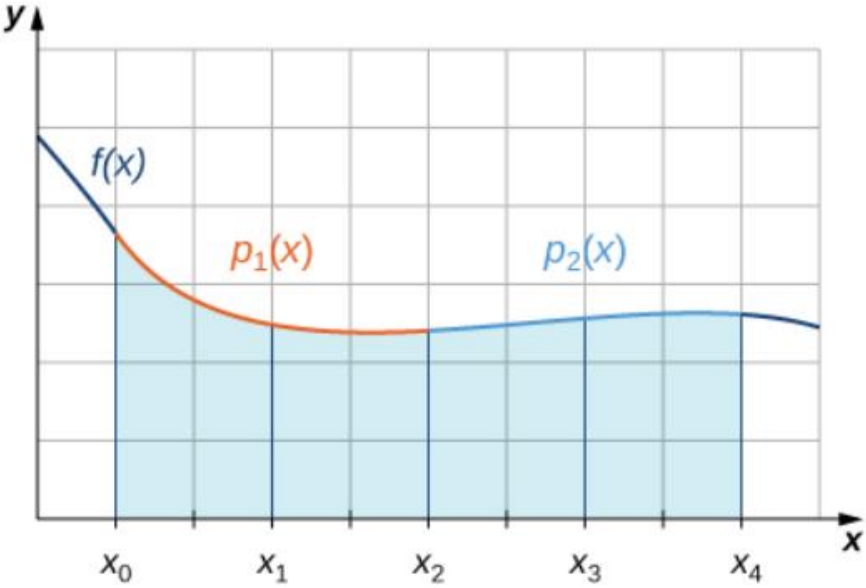

Figure 3. Simpson's rule

For example, in Figure 3, the integration between x_0 to x_1 is:

$$\int_{x_0}^{x_1} f(x)dx = (x_0 + 4x_1 + x_2)(\Delta x) \quad , \Delta x = (x_1 - x_0)$$

And the general formula of composite Simpson's rule is:

$$\int_a^b f(x)dx = \sum_{i=2k}^{b} \frac{(x_{i-1} + 4x_i + x_{i+1})(\Delta x)}{3} \quad , k = 1,2,3 \dots , \qquad \Delta x = (x_i - x_{i-1}) \qquad (6)$$

Simpson's rule is only works for even number of subintervals.

2. Code and Inputs

All results and figures were generated using code which given in appendix section. The input data pr3data is imported and then, the derivative of current is taken and stored inside of derivative() array. Then the derivatives of the current and voltage values are compared using a graph. Then, all power values are calculated by using equation (1) for all time values and the power-time graph is plotted. After that voltage versus time and current versus time graphs are plotted.

Next, the stored energy related to the time is calculated with using midpoint method which given in equation (4), trapezoidal rule which given in equation (5), Simpson's rule which given in equation (6) and then with using equation (3). All of the integration rules are compared with the results found in equation (3) since the equation (3) is provide us the real values of stored energy. And then all of the stored energy versus time graphs are plotted for comparing.

After that, an error analysis is made for estimating which of those three integral methods is the most accurate one.

In our problem, the intervals are predefined and we cannot change the number of subintervals. But in Simpson's rule and midpoint rule needs even number of subintervals for working so for calculating the odd number of subintervals, we can make some changes.

For example, in Figure 3, the integral between x_0 to x_3 cannot be calculated by using neither midpoint method nor Simpson's method. But we can make some modification for solving this problem. We can calculate the values between x_0 to x_2 since there are even number of subintervals. And for finding x_2 to x_3 subinterval, we can use Riemann over and Riemann under estimations.

$$riemann\ sum = \frac{f(x-1) \times h + f(x) \times h}{2} \tag{7}$$

With this method, we can find the integral at odd number of subintervals too.

3. Project results and discussions

A) Plotting the Current I(t) and Voltage v(t)

In this subsection, the derivative of current and the voltage plot is given in Figure 4. It can be seen in Figure 4 that the derivative of current and the voltage graph is consistent, that is because the Ohm's law of inductor formula is:

$$v = L\frac{di}{dt} \tag{8}$$

In our situation L is constant and 0.1 H therefore we expect to see a linear graph.

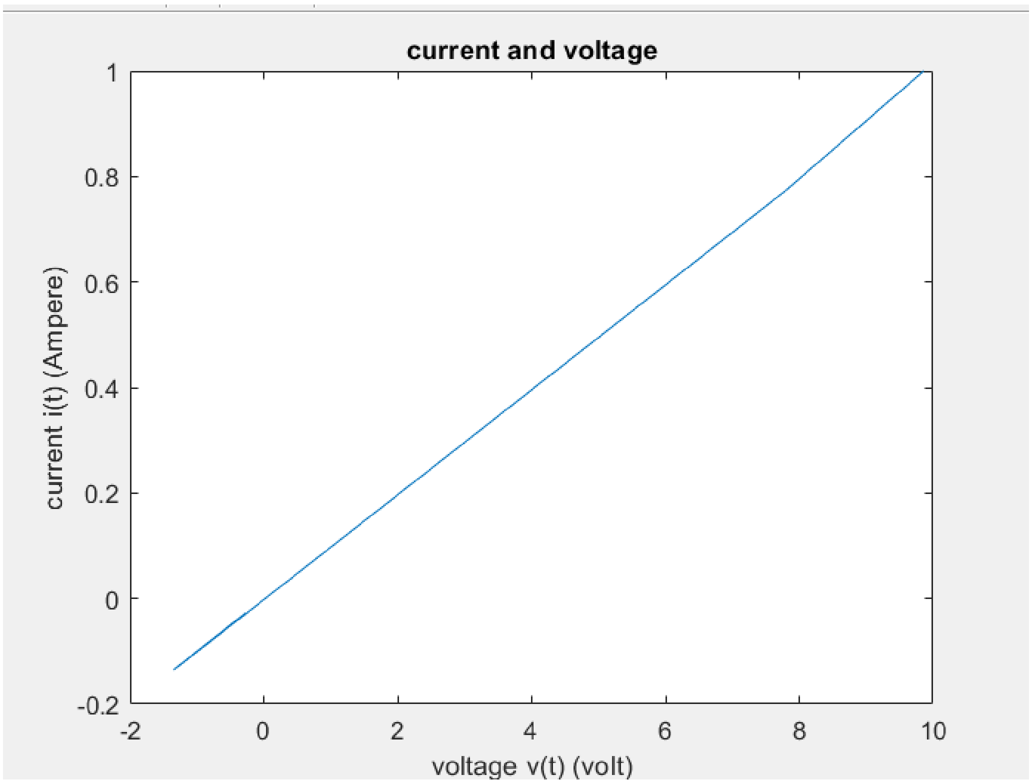

Figure 4. derivative of current versus voltage graph

B) Plotting Power Graph

In this subsection, the time versus power graph is given in Figure 5, the power can be calculated by using equation (1). This graph is going to use for calculating the stored energy therefore the stored energy must be reach it's maximum value where t=0.2 s since t=2 s is the point that power graph is started to go under zero.

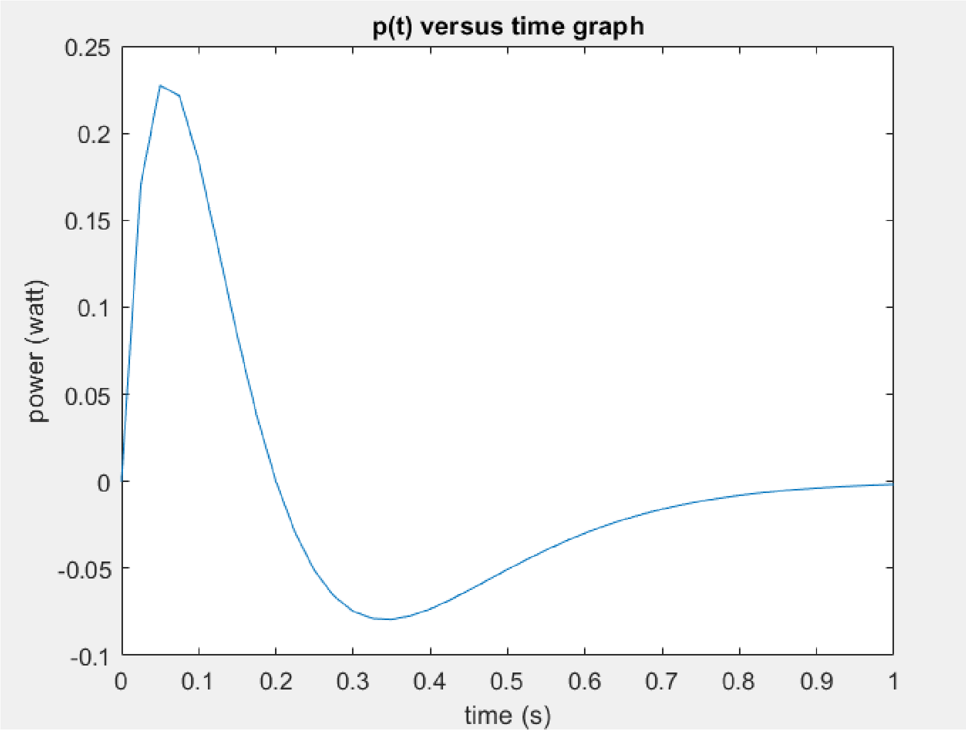

Figure 5. power versus time graph

C) Calculating the Stored Energy

In this subsection, stored energy graphs are given below. The graphs of Stored energy is reached it's maximum value at t=0.2 seconds as expected.

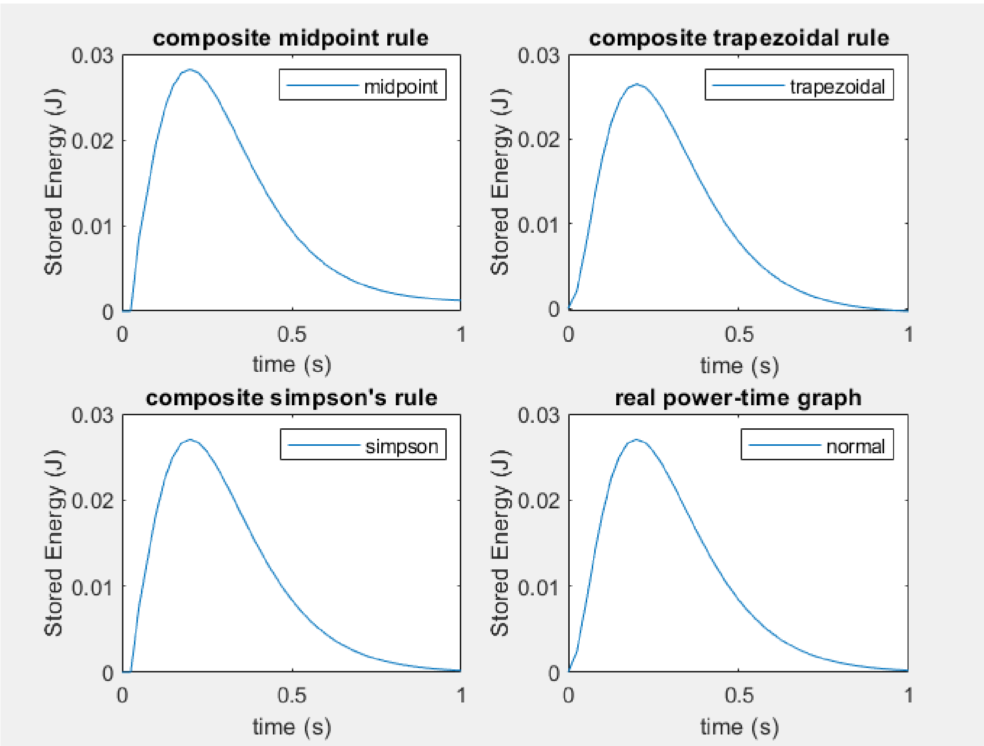

Figure 6. Stored Energy which calculated by different methods

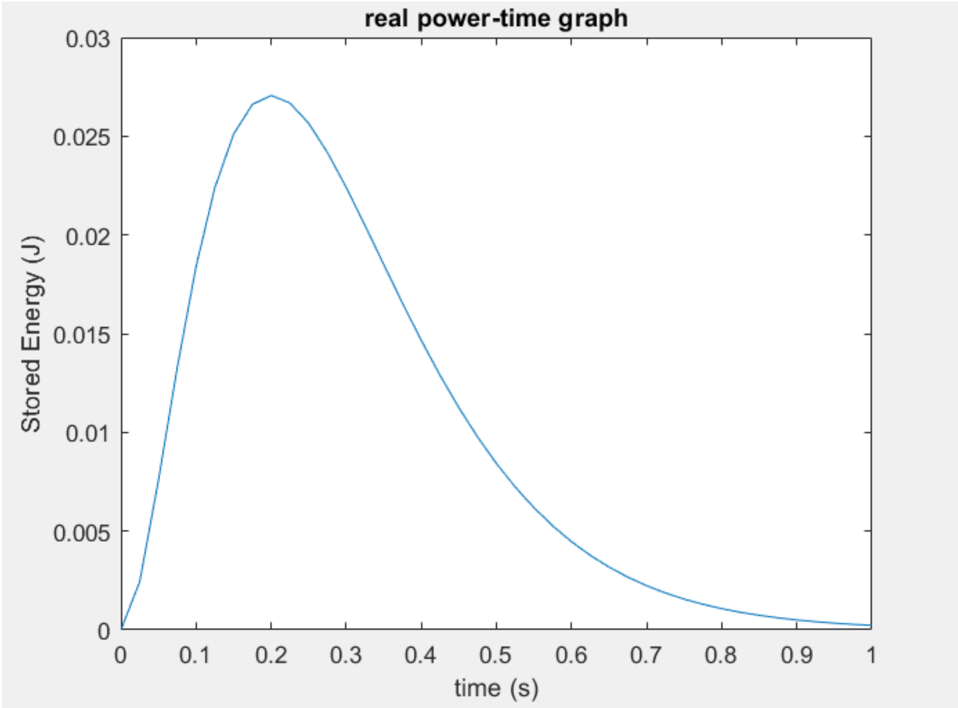

Figure 7. more detailed energy graph for equation 3

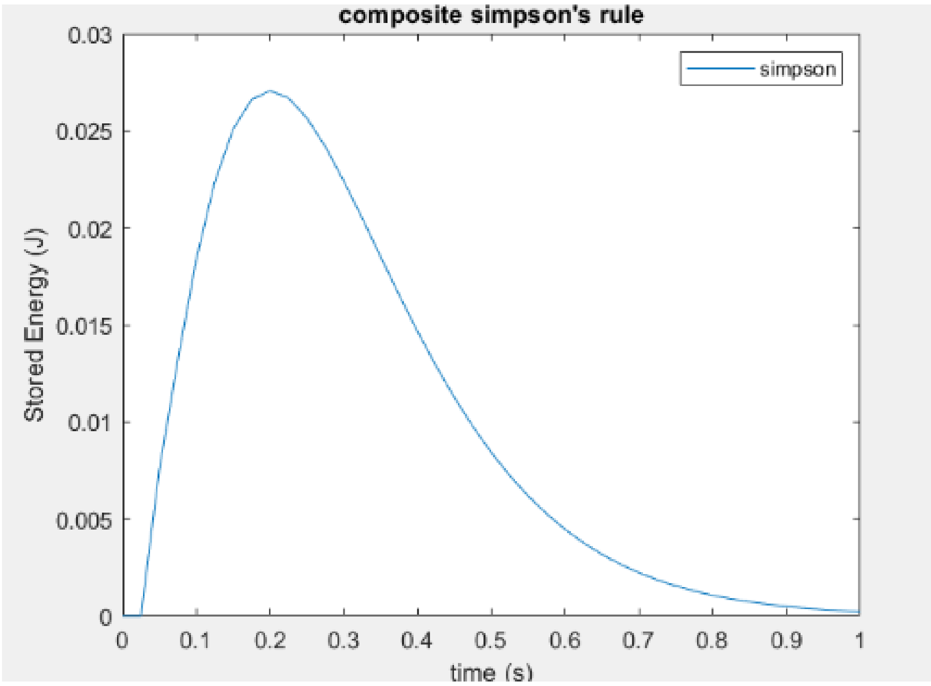

Figure 8. more detailed energy graph for Simpson’s

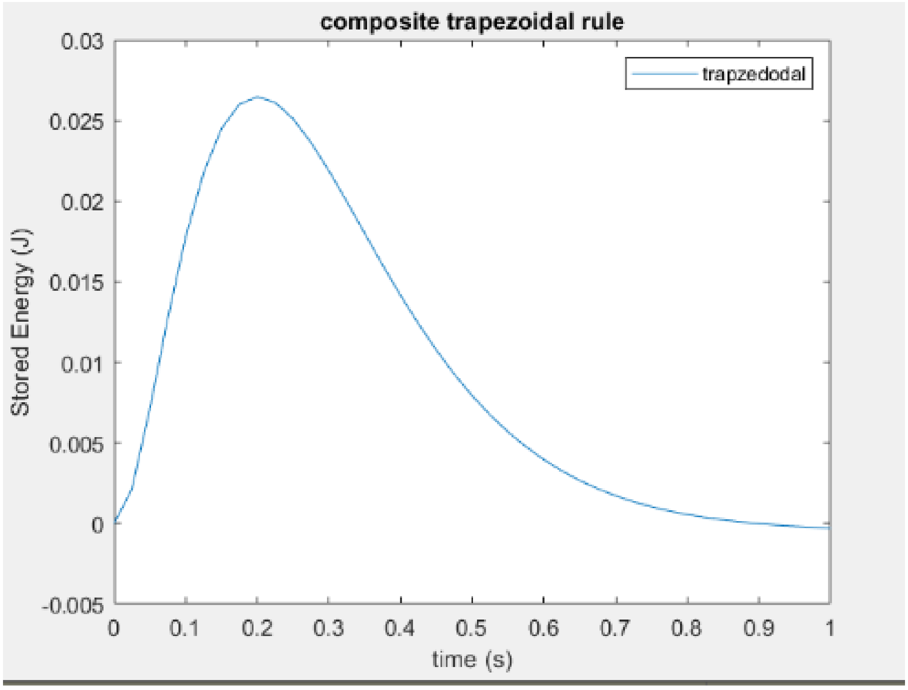

Figure 9. more detailed energy graph for trapezoidal

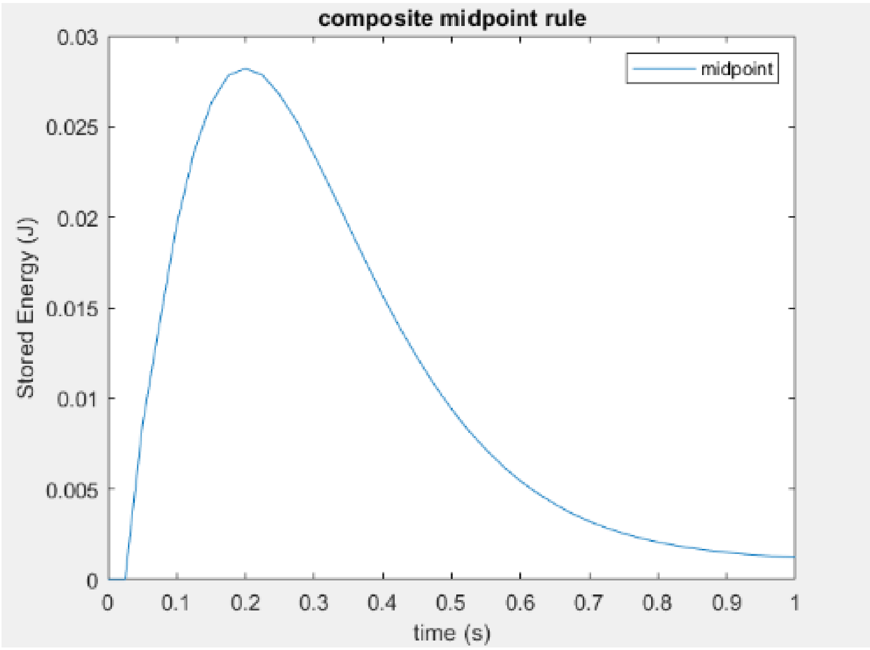

Figure 10. more detailed energy graph for midpoint

D) Error Analysis

The formula of error analysis is:

$$error = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left[E_{real} - E_{experimental}\right]^2} \tag{9}$$

With using equation (9), the errors for each of three integration method can be found:

error in composite midpoint rule is : 0.001065
error in composite trapezoidal rule is : 0.000519
error in composite Simpson's rule is : 0.000381

When four of these methods are plotted in the same figure, the accuracy of these methods can be seen in Figure 11.

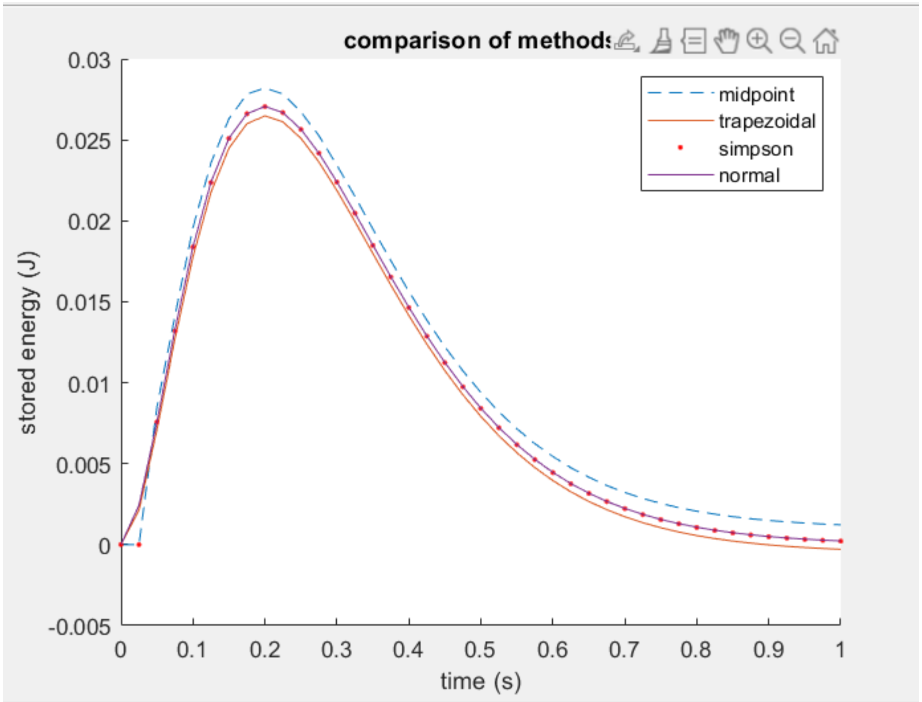

Figure 11. comparison of methods

4. Conclusion

It can be concluded that Simpson's rule gives the most accurate results comparing to other methods, and midpoint rule gives the least accurate results. Also it can be seen that the integral takes it's maximum value where t=0.2 s. and it starts to drop because the power graph is started to go below zero.

5. Appendix

The code is given below:

```matlab
clear all;clc;

load pr3data.dat;
%%
h=0.025;
derivative(1)=(1/(2*h))*(-3*pr3data(1,2)+4*pr3data(2,2)-pr3data(3,2));

for i=2:40
    derivative(i)=(1/(2*h))*(pr3data(i+1,2)-pr3data(i-1,2));
end
derivative(41)=(1/(2*h))*(+3*pr3data(41,2)-4*pr3data(40,2)+pr3data(39,2));

figure
plot(derivative,pr3data(:,3));
title("current and voltage");
xlabel("voltage v(t)  (volt)");
ylabel("current i(t)  (Ampere)");

p(41)=0;
for i=1:41
   p(i) = pr3data(i,2) * pr3data(i,3);
end
figure
plot(pr3data(:,1),p);
title("p(t) versus time graph");
xlabel("time (s)");
ylabel("power (watt)");
hold off;

%% composite midpoint integration
E_midpoint(41)=0;
for i=3:2:41
    for k=2:2:i
        E_midpoint(i)=E_midpoint(i)+p(k)*0.05;
    end
end

for i=4:2:41
   E_midpoint(i)=E_midpoint(i-1) + 0.025*( (p(i)+p(i-1))*0.5 );
end

figure('name','composite midpoint rule');
plot(pr3data(:,1),E_midpoint);
title("composite midpoint rule");
hold off;
xlabel("time (s)");
ylabel("Stored Energy (J)");
legend("midpoint");
%% composite trapezoidal integration
E_trapezoidal(41)=0;
for i=2:1:41
   for k=2:1:i
      E_trapezoidal(i) = E_trapezoidal(i) + 0.025*(p(k-1)+p(k))/2;
   end
end
figure('name','composite trapzedoidal rule');
plot(pr3data(:,1),E_trapezoidal);
title("composite trapezoidal rule");
xlabel("time (s)");
ylabel("Stored Energy (J)");
legend("trapzedodal");
%% composite simpson's rule
```

```
E_simpson(41)=0;
for i=3:2:41
   for k=3:2:i
   E_simpson(i) = E_simpson(i) + 0.025/3*(p(k-2)+4*p(k-1)+p(k));
   end
end
for i=4:2:40
   E_simpson(i)=E_simpson(i-1)  + 0.025*(p(i)+p(i-1))/2;
end
figure('name','simpson');
plot(pr3data(:,1),E_simpson);
title("composite simpson's rule");
xlabel("time (s)");
ylabel("Stored Energy (J)");
legend("simpson");
%% normal
for i=1:41
   E_normal(i) = 0.5*0.1*pr3data(i,2)*pr3data(i,2);
end
figure('name','voltage current relation');
plot(pr3data(:,1),E_normal);
title("real power-time graph");
xlabel("time (s)");
ylabel("Stored Energy (J)");;

figure;
subplot(2,2,1);
plot(pr3data(:,1),E_midpoint);
title("composite midpoint rule");
legend("midpoint");
hold off;
xlabel("time (s)");
ylabel("Stored Energy (J)");

subplot(2,2,2);
plot(pr3data(:,1),E_trapezoidal);
title("composite trapezoidal rule");
xlabel("time (s)");
ylabel("Stored Energy (J)");
legend("trapezoidal");

subplot(2,2,3);
plot(pr3data(:,1),E_simpson);
title("composite simpson's rule");
xlabel("time (s)");
ylabel("Stored Energy (J)");
legend("simpson");

subplot(2,2,4);
plot(pr3data(:,1),E_normal);
title("real power-time graph");
xlabel("time (s)");
ylabel("Stored Energy (J)");
legend("normal");
 %% analysis
 figure;
 hold on;
 plot(pr3data(:,1),E_midpoint,'--');
 plot(pr3data(:,1),E_trapezoidal,'-');
plot(pr3data(:,1),E_simpson,'.r');
plot(pr3data(:,1),E_normal);
xlabel("time (s)");
```

```
ylabel("stored energy (J)");
title("comparison of methods");
hold off;
legend("midpoint","trapezoidal","simpson","normal");
fprintf("error in composite midpoint rule is : %f\n",norm(E_midpoint-
E_normal) / sqrt(41));
fprintf("error in composite trapzedoidal rule is :
%f\n",norm(E_trapezoidal-E_normal) / sqrt(41));
fprintf("error in composite simpson'S rule is : %f\n",norm(E_simpson-
E_normal) / sqrt(41));
```