GEBZE TEKNİK ÜNİVERSİTESİ

DEPARTMENT OF ELECTRONIC ENGINEERING

MATH214 NUMERICAL METHODS

2020 – 2021 FALL TERM

Project 3

1801022001
Ömer Sarımeşe

Preparation date	Upload date
14.12.2020	16.12.2020

I. Problem Definition and Formulation

The main goal is finding stored energy in the inductor by numerical integration methods. Voltages and current values change by time and given by data file. Power of inductor can be measured by

$$p(t) = i(t).v(t) \tag{1}$$

Equation(1). Power values are changing by alteration of current and voltage. Stored energy in inductor is shown as follows:

$$w(t) = \int_0^t p(t).dt = \int_0^t v(t).i(t).dt \tag{2}$$

This equation(2) will be useful during numerical integration. Stored energy can be shown like:

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

The equation(3) will give the exact stored energy value of inductor. Applying first and last current values in equation(3) will give stored enegy in inductor when reached the certain time.

For calculate the stored energy in inductor, There are some numerical integration methods. In this case, composite simpson rule is the most common method and it will use alongside composite trapezoidal rule and composite midpoint rule. Trapezoidal rule is

$$\int_a^b f(x)dx = \frac{h}{2}[f(a) + 2\sum_{j=1}^{n-1} f(x_j) + f(b)] - \frac{b-a}{6}h^2 f''(\mu) \tag{4}$$

$$h = \frac{b-a}{n} \quad , \quad x_j = a + j.h$$

shown as equation(4). 'n' value can be even or odd. Error part will not used in calculations because lack of second derivative. Other numerical integration method is composite midpoint rule:

$$\int_a^b f(x)dx = 2h\sum_{j=0}^{\frac{n}{2}} f(x_{2j}) + \frac{b-a}{6}h^2 f''(\mu) \tag{5}$$

$$h = \frac{b-a}{n+2} \quad , \quad x_j = a + (j+1)h$$

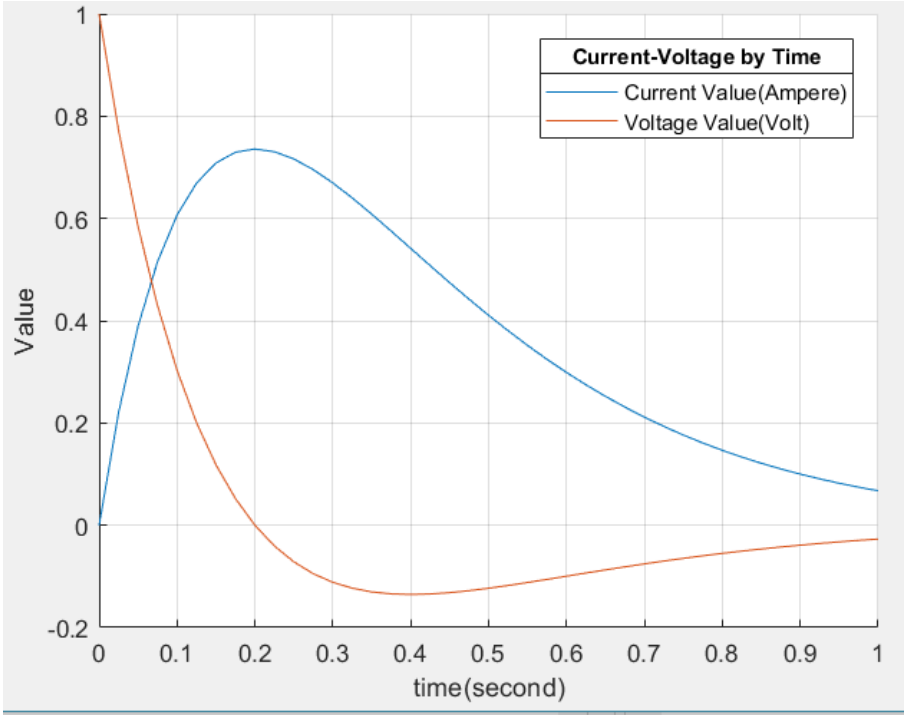
Equation(5) and Equation(4) have same error order. 'n' must be even. Their results should be close each other. Last but not least method is composite simpson rule:

$$\int_a^b f(x)dx = \frac{h}{3}[f(a) + 2\sum_{j=1}^{(\frac{n}{2}-1)} f(x_{2j}) + 4\sum_{j=1}^{\frac{n}{2}} f(x_{2j-1}) + f(b)] - \frac{b-a}{180}h^4 f^{(4)}(\mu) \tag{6}$$

$$h = \frac{b-a}{n} \quad , \quad x_j = a + j.h$$

Equation(6) has $O(h^4)$ and for that reason, composite simpson rule is should be much more accurate than other methods. Like composite midpoint rule 'n' must be even.

II. Current and Voltage data

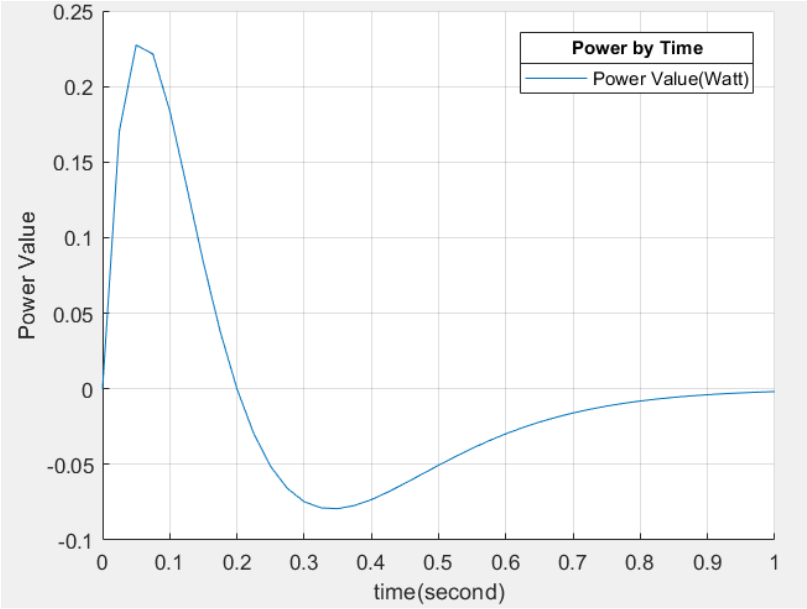


Graph-1: Voltage-Current Change by time

Graph-1 visualize change of voltage and current by time. Lenz's law explain that voltage will be opposite purpose to the current because voltage try to manage current unchanged. Always act like opposite side to keep current at old value. This graph-1 shows us between 0-0.2 second, voltage deliberately decrease to prevent current increase. Between 0.2 to 1 second range, they are getting closer and both of their rate of converge are much smaller. Graph-1 verify that the data conform Lenz's law and consistent.

III. Power of Inductor

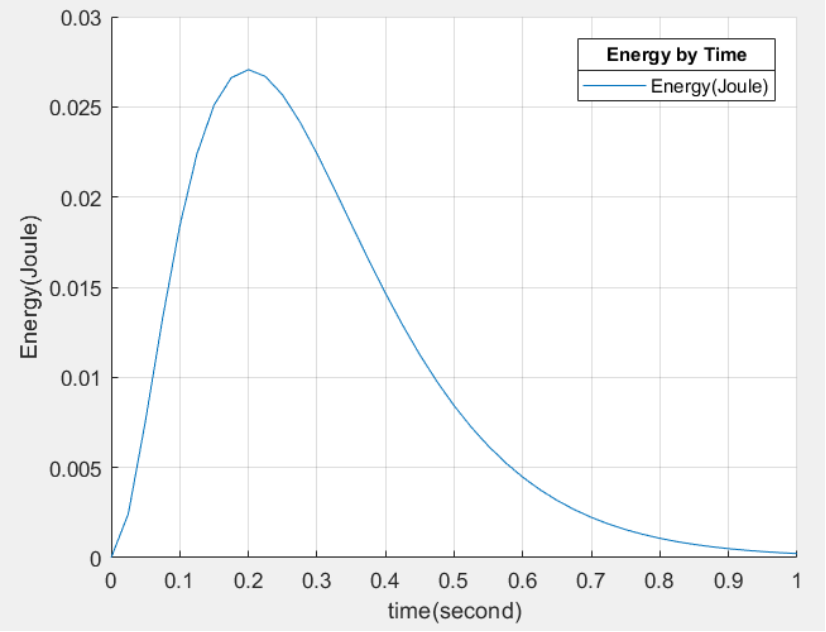
Graph-2 is under this segment of comments on power. İf all dissipated and supplied power gathered, the power value is being -0.00363 Watt. Equation(1) has been used to calculate power of inductance. Negative power value means this circuit element supplied both current and voltage. This result makes us consider the stored energy should be positive because in numerical integration part, we will integrate power by time to find stored energy.



Graph-2: Power – time graph

IV. Stored Energy in Inductor

Equation(3) requires inductance, current values by time. So stored energy is related by current and plot of stored energy shows that clearly.



Graph-3: Stored energy in inductor

Graph-3 shows the stored energy at exact time. So difference between energy value when time goes to 0 to 1, gives stored energy at this range of time. Last and first current values are applied to equation(3) and their difference is found 2.27e-04 joule. This is exact value of stored energy in inductor.

As clearly can be seen, times between 0 to 0.2 second, inductor gathering so much energy and it is totally related to fast converge current at these moments. For the exactly same reason, between 0.2 to 1 second time energy dropping down so quickly just like the current. In the end, it looks like there is not much change on energy between start and finish of the case. 2.27e-04 joule is a positive value, that means inductor has more energy in last time compare to start.

V. Matlab Code

The data file named and declared as ‘f’. Between lines 6-16 program just plotting imported volt and ampere values at y-axis, time steps at x-axis. In lines between 16-33, power of inductor is calculated by multiplying current and voltage values which are in the same time row. Then plotting power by time. In stored energy part, code working with current values and saving values in Energy_D array. Then plotting it by time. Lastly, the exact value of stored energy is found in line 46, using equation(3). Remaining lines are for the numerical integration methods. Firstly, composit trapezoidal method is using between lines 47-56. ‘n’ value is set length of file. ‘h’ value is calculated by extracting last time value and first time value on data file, then program divide that result by ‘n’. For loop is for calculate series sum. The loop starts from 1 and finishes at n-1. Inside the loop, Voltage and current values are multiplied and added series sum. In line 54 Composite trapezoidal formula is applied. Between lines 56-65 composit midpoint method is working. ‘s4’ is variable for series sum and it is set to current times voltage which is in first row in data file. ‘n’ is length of file again. ‘h’ is now finding with divide n+2 not n. The for loop is stating from i=1 to i= n/2. Inside the loop program multiply current and voltages which are in double rows because in formula, series open like that. Then program adding this value to ‘s4’ series sum variable. After the loop finish, result of composite midpoint rule is calculating with this line 63: result_M=2*h*s4. Between lines 65-87 code has two for loops for composite simpson rule. First loop basically adding even rows’ power value and the second loop adding odd rows’ power values. In this section of code, n is length of file plus one. ‘s1’ variable for odd sum series, ‘s2’ variable for even sum series. Loops are totally working same as formula’s series. After these two loop code is running line 76 wihch is calculating composite simpson’s result, using sum of two series with some coefficients and initial,last power values. Then program will finish her tasks.

VI. Result and analysis of Methods

Composite trapezoidal rule gave -2.827e-04, composite midpoint rule gave 0.01 and -7.993e-04 is given by composite simpson rule. To remind again, exact value fort his problem is 2.27e-04. Clearly, composite midpoint rule is totally wrong and there must be some miscalculations on program. Composite trapezoidal rule gave the smallest error and closest to exact value. Composite simpson rule gave us quite acceptable result However, it should have been the most accurate one. Probably the series in formulas are expressed in a wrong way. When some of the first or the latest terms do not adding sum, the result is became different and causes huge difference. But generally, the composite simpson rule is the most accurate one because of the order it has. Midpoint should be the less accurate one, the reason is there is peek at 0.1 second time in power graph. Midpoint can be miss incomparably big numbers at this peek part of the function. Composite trapezoidal rule is gave us more accurate approximation of change compare to composite simpson rule. All of the integration methods work ok except composite midpoint. Taking different subintervals have to solve the problem, if there is no mistake in code.

To conclude, composite trapezoidal and composite simpson rule are close to exact stored energy value. Somehow composite midpoint rule is not gave an accurate result. Methods understandably has some errors and reason why the errors occured is given previous paragraph. If we look at the inductor's current-voltage graph(1), current and voltage act consistently as it is supposed to be. The inductor store more energy compare to initial values and gathering power from circuit.

MATLAB CODE

```
clear all;
close all;
format long;
L=0.1;
f=importdata('pr3data.dat');
%-------current-voltage graph-------------------------
figure
grid on;
hold on;
plot(f(:,1),f(:,2));
plot(f(:,1),f(:,3));
ylabel('Value');
xlabel('time(second)');
a=legend('Current Value(Ampere)','Voltage Value(Volt)');
title(a,'Current-Voltage by Time');
%-------Power graph--------------------------------
power=zeros(1,length(f));
total=0;
for i=1:1:length(f)
    power(i)=f(i,2)*f(i,3);
    total=power(i)+power(i);
end
display("power of inductance:");
```

```
display(total);
figure
grid on;
hold on;
plot(f(:,1),power(:));
ylabel('Power Value');
xlabel('time(second)');
a=legend('Power Value(Watt)');
title(a,'Power by Time');
%----------Stored Energy by data----------------------
Energy_D=zeros(1,length(f));
for i=1:1:length(f)
    Energy_D(i)=(1/2)*L*((f(i,2))^2);
end
figure
grid on;
hold on;
plot(f(:,1),Energy_D(:));
ylabel('Energy(Joule)');
xlabel('time(second)');
a=legend('Energy(Joule)');
title(a,'Energy by Time');
EXACT_VALUE=(0.5)*(0.1)*((f(length(f),2)-f(1,2))^2); % stored energy
%----------composite trapezoidal----------------------
s3=0;
n=length(f);
h=((f(length(f),1)-f(1,1))/n);
for i=1:1:n-1
    s3=s3+(f(i,2)*f(i,3));
end
result_T=(h/2)*((f(1,2)*f(1,3))+(f(length(f),2)*f(length(f),3))+(2*s3));
display(result_T);
%----------composite midpoint-------------------------
s4=(f(1,2)*f(1,3)); % i=0 için sonuç
n=length(f)-1;
h=((f(length(f),1)-f(1,1))/n+2);
for i=1:1:n/2
    s4=s4+(f((2*i),2)*f((2*i),3));
end
result_M=2*h*s4;
display(result_M);
%--------composite simpson---------------------------
s1=0;
s2=0;
n=length(f)+1;
h=((f(length(f),1)-f(1,1))/n);
for i=1:1:(n/2)-1 % 2-40 arasi çiftler
    s1=s1+(f(2*i,2)*f(2*i,3));
end
for i=1:1:(n/2) % 1-41 arasi tekler
    s2=s2+(f((2*i)-1,2)*f((2*i)-1,3));
end
result_S=(h/3)*((f(1,2)*f(1,3))+(f(length(f),2)*f(length(f),3))+(2*s1)+(4*s
2));
display(result_S);
%-----------------------------------------------------
```