NUMERİCAL METHODS/MATH214

PROJECT 3

Numerically Integrating with

A) Composite Midpoint Rule,
B) Composite Trapezoidal Rule,
C) Composite Simpson's Rule.

Muhammed Cemal Eryiğit
1801022024

Deadline 16/12/2020 – 17.00

Preparation Time 11-16/12/2020

1- Problem definition and introduction formulas

The problem is to calculate the work done in time t for given voltage and current data. In order to calculate the work, firstly, the power value is calculated by using the voltage and current values. With this formula;

$$\boldsymbol{p(t) = v(t) * i(t)}$$

The integral of the calculated power value in a particular time interval gives the energy stored by the indicator. The mentioned integral calculation;

$$\boldsymbol{w(t) = \int_a^b p(t)dt = \int_a^b v(t)i(t)\,dt}$$

Integral to be calculated; It is found using the Composite Midpoint rule, the Composite Simpson's rule and the Composite Trapezoidal rule.

A- **Composite Midpoint rule** is calculated as the sum of rectangular areas between points in a given interval. Composite midpoint rule for n+2 subintervals can be written with its error term as;

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

Here " h = (b - a)/n " is the value of the interval, "n" is the number of intervals, [a-b] is the intervals of integral. The total symbol increases by a multiple of two. $f''(\delta)$ error term

B- In **Composite Simpson's Rule**, we use parabolas to approximate each part of the curve. Simpson's rule is calculated with n intervals between two points. Composite Simpson's rule for n subintervals can be written with its error term as

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

Here " h = (b - a)/n " is the value of the interval, "n" is the number of intervals, [a-b] is the intervals of integral, $f^{(4)}(\delta)$ error term. Note that the error term for the compound Simpson's rule is O (h4), whereas O (h5) for the standard Simpson's rule. However, these rates are not comparable because we fixed the standard Simpson's rule at h = (b - a) / 2, but Composite Simpson's. The rule is that for n even integers we have h = (b - a) / n. This allows us to significantly reduce the value of h using the compound Simpson's rule. In this way this proves to be very efficient as it is generally more accurate than other numerical methods.

C- **Composite Trapezoidal Rule** give a way to approximate the integral [a, b] in a given interval. We divide the interval [a, b] into n sub-intervals, where h = (b - a)/n. Thus, the trapezoidal rule applied to the interval is

$$\int_a^b f(x)dx = \frac{h}{2}\,[\,f(a) + 2\sum_{j=1}^{n-1} f(x_j) + f(b)] - \frac{b-a}{12} h^2 f^{(2)}(\delta)$$

Here " h = (b - a)/n " is the value of the interval, "n" is the number of intervals, [a-b] is the intervals of integral, $f^{(2)}(\delta)$ error term.

2- Code description and inputs

All result and figures are generated using the code, which is given at the appendix section. The input data given by pr3data.dat[t,i,v] are imported and columns of the imported data are assigned to [:,t], [:,i], [:,v] respectively. The inductance value is assigned to L as 0.1 [H]. To plotted at 0.025 s intervals in the time interval [0,1] so that the values can be seen (figure-1) (figure-2). To control the inductance current, take the numerical derivation of the voltage and checked it (figure-3). The power of the inductance was calculated by multiplying the current and voltage values and indicated in the plot (figure-4).

The symbol of sum in the formula of the midpoint rule is calculated (using the values h, n, cvc). Interval values into sub-intervals is divided to calculate the midpoint rule. X_{-1} and X_{n+1} also attend this sub-intervals.

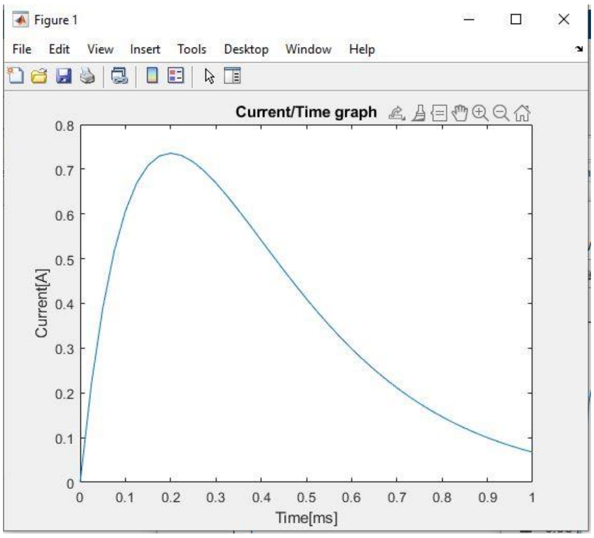
The symbol of sum in the formula of the Simpson's rule is calculated (using the values h, n, cx1, cx2 f(a), f(b)). Interval values into sub-intervals is divided to calculate the Simpson's rule because intervals are not adequate n=n*2.

The symbol of sum in the formula of the trapezoidal rule is calculated (using the values h, n, cxc, f(a), f(b)).

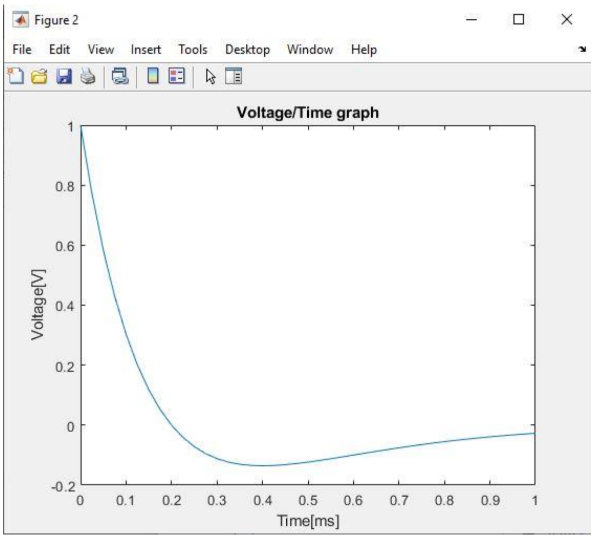
The stored energy is calculated using this formula;

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

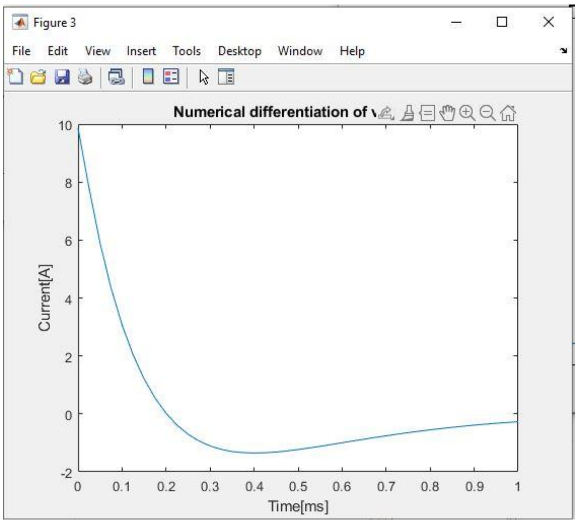
The calculated work value is equal to the work done in 1 second.



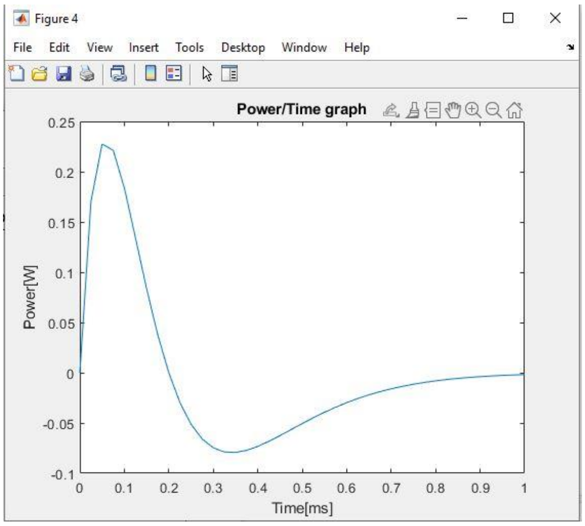
(figure-1)



(figure-2)



(figure-3)



(figure-4)

3- Project Result and Analysis

Comment on error rates and comparisons between methods will be evaluate in this section.

A- Error Rate

The composite Simpson's rule is an obvious choice if you want to minimize the computation because it converges faster than other methods. When comparing the methods, Simpson's rule requires less computation, so it contains less rounding error than the trapezoidal and midpoint rule.

The error for the trapezoidal rule applied to a intervals is given by $\frac{b-a}{12}h^2 f^{(2)}(\delta)$ If we are applying the composite trapezoidal rule to n intervals, each of width h = (b - a)/n, the error for the composite-trapezoidal rule is the sum of the errors on each of the individual intervals.

To simplify this expression, we determine that the sum approximates the average value of $f^{(2)}(\delta)$ so with this equation the error rate is found

The error for the Simpson's rule applied to a intervals is given by $\frac{b-a}{180}h^4 f^{(4)}(\delta)$ If we are applying the composite Simpson's rule to 2n intervals because the interval n does not allow us to obtain the required result in the given data.

To simplify this expression, we determine that the sum approximates the average value of $f^{(4)}(\delta)$ so with this equation the error rate is found

The error for the midpoint rule applied to a intervals is given by $\frac{b-a}{6}h^2 f^{(2)}(\delta)$ If we are applying the composite midpoint rule to n+2 intervals, each of width h = (b - a)/n, the error for the composite-midpoint rule is the sum of the errors on each of the individual intervals but in order to calculate midpoint correctly the intervals are Δt/2 instead of Δt thus the value of n doubled.

To simplify this expression, we determine that the sum approximates the average value of $f^{(2)}(\delta)$ so with this equation the error rate is found

Note: The values used in the formula were used as I*V, not P (P=I*V) for minimize rounding errors.

B- Analysis

The work value (stored energy) was calculated as 2.269996488124243e-04 between 1 to 0 seconds. When the required numerical integration calculations were calculate, it was observed that the closest value (1.071098271928057e-04) to this value was Simpson's rule. Then observed that the trapezoidal rule value (-2.898579253196828e-04) converged. Finally, the midpoint rule value (-3.049345270252929e-04) observed to converge least.

4- Appendix Section

Numerical Analysis Richard L. Burden - J. Douglas Faires book was used.

Code

```
clear all
clear
clc

load pr3data.dat

t=pr3data(:,1)'; i=pr3data(:,2)'; v=pr3data(:,3)';
%Current graph
figure
plot(t,i);
xlabel('Time[ms]')
ylabel('Current[A]')
title('Current/Time graph');
%voltage graph
figure
plot(t,v);
xlabel('Time[ms]')
ylabel('Voltage[V]')
title('Voltage/Time graph');
%Derivative of Current
for k=2 : length(t)-1
    h=pr3data(2,1);
    deriv_c(1,1)=(((-3*i(1,1))+4*i(1,2)-i(1,3))/(2*h));
    deriv_c(1,41)=(((3*i(1,41))-4*i(1,40)+i(1,39))/(2*h));
    deriv_c(1,k)=((i(1,(k+1))-i(1,k-1))/(2*h));
end
figure
plot(t,deriv_c)
xlabel('Time[ms]')
ylabel('Current[A]');
title('Numerical differentiation of voltage')
%Power graph
for k=1 :length(i)
    p(:,k)=i(:,k)*v(:,k);
end
figure
plot(t,p)
xlabel('Time[ms]')
```

```matlab
ylabel('Power[W]')
title('Power/Time graph');
%definition
a=0; %lower limit
b=1; %upper limit
L=0.1;

%trapezoidal rule
n=length(t)-1;
h=(b-a)/n;
cxc=0;
for k=2:40
    cxc=cxc+i(1,k)*v(1,k);
end
pptrapp=(h/2)*(i(1,1)*v(1,1)+(2*cxc)+i(1,41)*v(1,41));

%simpsons rule
n=80;
h=(b-a)/n;
cx1=0;
cx2=0;
for j=3:2:n/2-1
    cx1=cx1+i(1,j)*v(1,j);
end
for k=2:2:n/2
    cx2=cx2+i(1,k)*v(1,k);
end
ppsimpp=(2*cx1+4*cx2+i(1,1)*v(1,1)+v(1,41)*i(1,41))*h/3;

%midpoint rule
n=80+2;
h=(b-a)/n;
cvc=0;
for j=1:1:n/2
    cvc=cvc+i(1,j)*v(1,j);
end
ppmidpp=2*h*cvc;
%work
w=1/2*L*(i(1,41)^2);
```