GEBZE TEKNİK ÜNİVERSİTESİ

DEPARTMENT OF ELECTRONIC ENGINEERING

MATH214 NUMERICAL METHODS

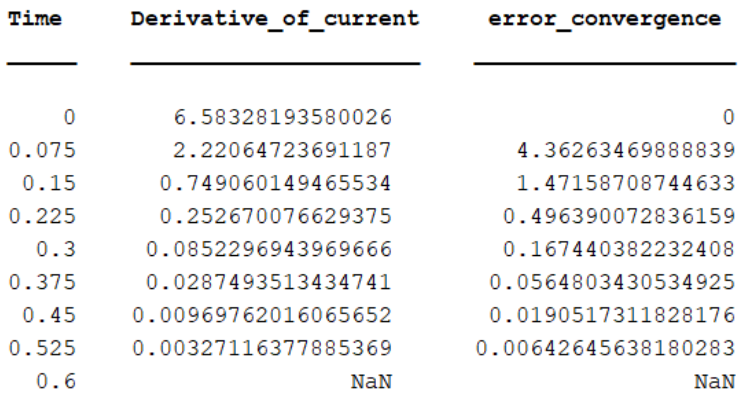
2020 – 2021 FALL TERM

Project 2

1801022001	Upload date
Ömer Sarımeşe	30.11.2020

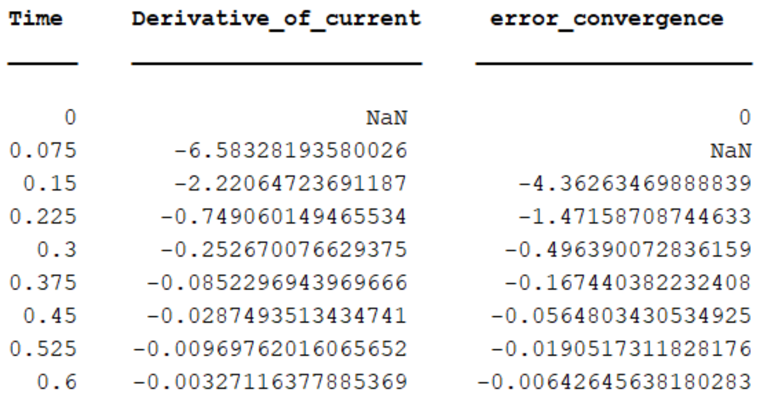
I. Derivative of Current

Forward difference, backward difference and three point endpoint formulas are used finding derivatives from the given data. Gap between current values is represented by “h”. When h>0, Forward Difference formula can be applied. Which is $\frac{f(x+h)-f(x)}{h}+0(h)\approx f'(x)$. If h<0, then Backward difference method is used. $\frac{f(x)-f(x-h)}{h}+0(h)\approx f'(x)$ is the backward difference formula. These two formulas requires both previous and next point to calculate derivative and for that reason one of the endpoint cannot be used. Derivative for first point cannot be found in backward difference method in the same way Derivative of last point in forward difference method. Three point endpoint method has three different formulas. $f'(x_0)\approx {}^{1}/_{2h}\,[-3f(x_0)+4f(x_0+h)-f(x_0+2h)]$ can be used for the first point. $f'(x_0)\approx {}^{1}/_{2h}\,[f(x_0+h)-f(x_0-h)]$ formula can be used for midpoints. $f'(x_0)\approx {}^{1}/_{-2h}\,[-3f(x_0)+4f(x_0-h)-f(x_0-2h)]$ can be used for last point. Reason why there is two different end point formula because of the last point does not have any next point, so i multiple both inside of functions and h value with -1. There for, derivative of last term can be calculated with using previous points values.



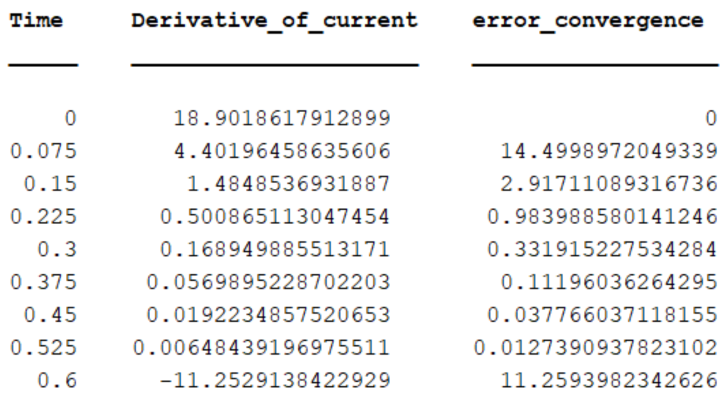
```
Time     Derivative_of_current     error_convergence
_____    _____________________    ___________________

    0         6.58328193580026                      0
0.075         2.22064723691187       4.36263469888839
 0.15        0.749060149465534       1.47158708744633
0.225        0.252670076629375      0.496390072836159
  0.3       0.0852296943969666      0.167440382232408
0.375       0.0287493513434741     0.0564803430534925
 0.45      0.00969762016065652     0.0190517311828176
0.525      0.00327116377885369    0.00642645638180283
  0.6                      NaN                    NaN
```

Table 1:Derivative-Forward difference method-case1



```
Time     Derivative_of_current     error_convergence
_____    _____________________    ___________________

    0                      NaN                      0
0.075        -6.58328193580026                    NaN
 0.15        -2.22064723691187      -4.36263469888839
0.225       -0.749060149465534      -1.47158708744633
  0.3       -0.252670076629375     -0.496390072836159
0.375      -0.0852296943969666     -0.167440382232408
 0.45      -0.0287493513434741    -0.0564803430534925
0.525     -0.00969762016065652    -0.0190517311828176
  0.6     -0.00327116377885369   -0.00642645638180283
```

Table 2:Derivative-Backward difference method-case1



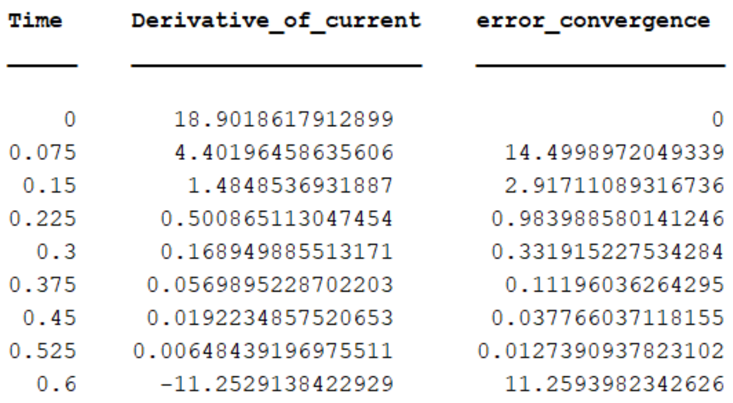
```
Time      Derivative_of_current     error_convergence
______    _____________________     __________________

     0       18.9018617912899                       0
 0.075       4.40196458635606        14.4998972049339
  0.15        1.4848536931887        2.91711089316736
 0.225      0.500865113047454       0.983988580141246
   0.3      0.168949885513171       0.331915227534284
 0.375     0.0569895228702203        0.11196036264295
  0.45     0.0192234857520653       0.037766037118155
 0.525    0.00648439196975511      0.0127390937823102
   0.6      -11.2529138422929        11.2593982342626
```

Table 3:Derivative-Three point midpoint method-case1

As clearly seen by Table 1,Table 2 and Table 3 ,after seven iterations Three point midpoint method has less error than other methods. In Three point midpoint method, end points have huge derivative results because of the end point formulas caused problems . Probably reason is irrelevant coefficients of functions in formula. İf we look closely there is no difference between forward method and backward method except forward difference method does not give derivative for last term, Bacward difference method doesn't give at first. As expected The three point midpoint formula converge much better rate compare to others. İt is because Three point midpoint formula has O(h^2) which means second order. At the same time Forward and backward methods has O(h) first order. İt makes Three point midpoint method more converge and accurate.

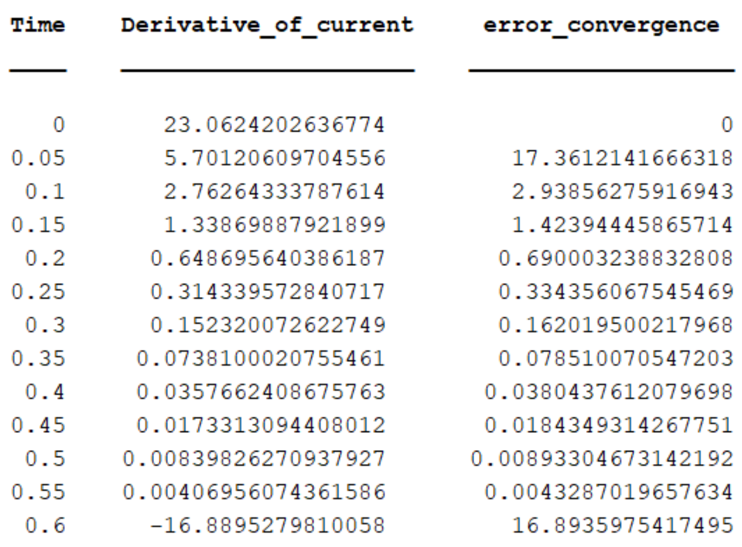
II. Time step size (Δt)

Gap between one point to other is h. İn this problem we are estimating that less h or Δt will give us more accurate results. To be understand that we need to compare cases which has same method and different time step sizes. Three point mid point formula is the most accurate one so i will use that method.



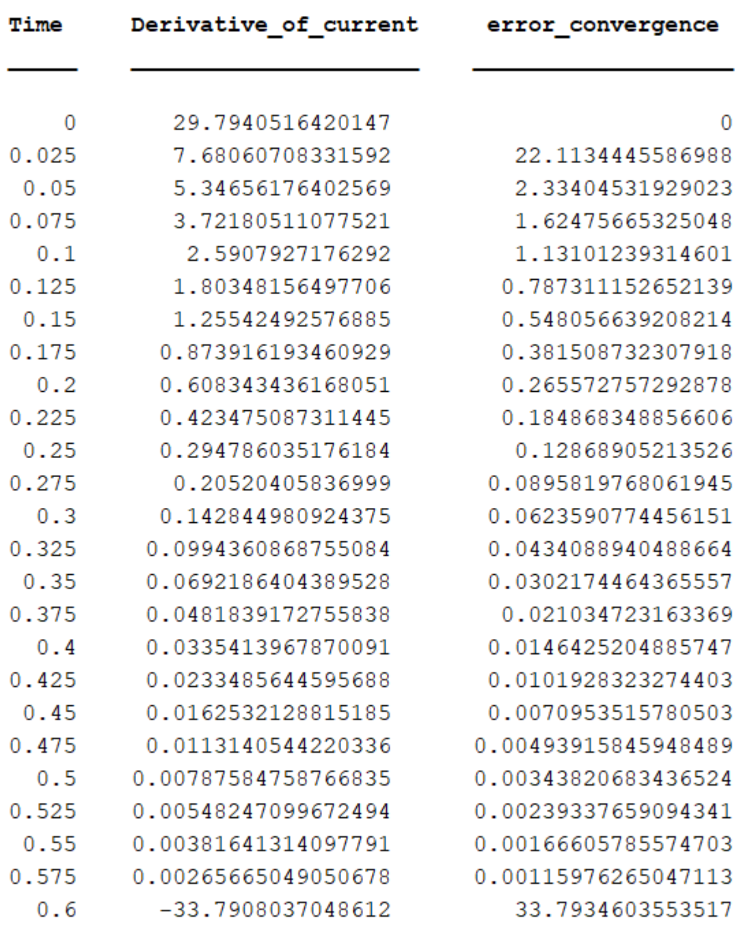
```
Time      Derivative_of_current     error_convergence
______    _____________________     __________________

     0       18.9018617912899                       0
 0.075       4.40196458635606        14.4998972049339
  0.15        1.4848536931887        2.91711089316736
 0.225      0.500865113047454       0.983988580141246
   0.3      0.168949885513171       0.331915227534284
 0.375     0.0569895228702203        0.11196036264295
  0.45     0.0192234857520653       0.037766037118155
 0.525    0.00648439196975511      0.0127390937823102
   0.6      -11.2529138422929        11.2593982342626
```

Table 4:Derivative-Three point midpoint method-case1($\Delta t = 0.075$)



Time	Derivative_of_current	error_convergence
0	23.0624202636774	0
0.05	5.70120609704556	17.3612141666318
0.1	2.76264333787614	2.93856275916943
0.15	1.33869887921899	1.42394445865714
0.2	0.648695640386187	0.690003238832808
0.25	0.314339572840717	0.334356067545469
0.3	0.152320072622749	0.162019500217968
0.35	0.0738100020755461	0.078510070547203
0.4	0.0357662408675763	0.0380437612079698
0.45	0.0173313094408012	0.0184349314267751
0.5	0.00839826270937927	0.00893304673142192
0.55	0.00406956074361586	0.0043287019657634
0.6	-16.8895279810058	16.8935975417495

Table 5:Derivative-Three point midpoint method-case2($\Delta t = 0.050$)



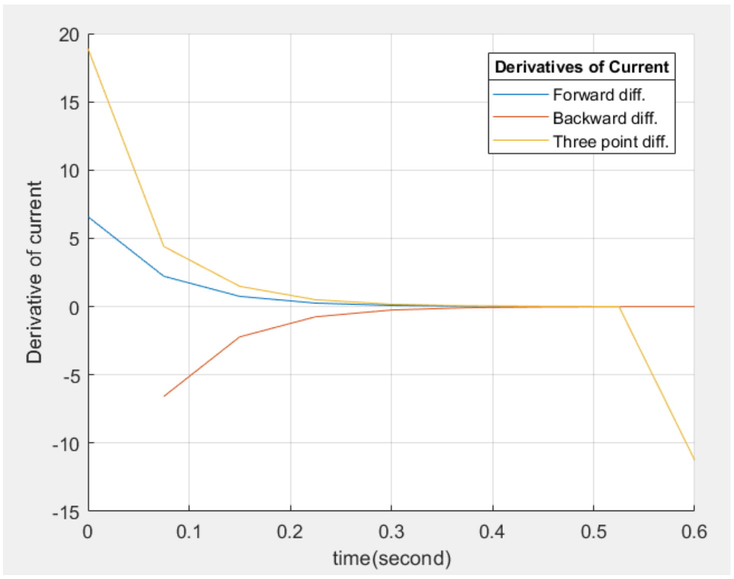
Time	Derivative_of_current	error_convergence
0	29.7940516420147	0
0.025	7.68060708331592	22.1134445586988
0.05	5.34656176402569	2.33404531929023
0.075	3.72180511077521	1.62475665325048
0.1	2.5907927176292	1.13101239314601
0.125	1.80348156497706	0.787311152652139
0.15	1.25542492576885	0.548056639208214
0.175	0.873916193460929	0.381508732307918
0.2	0.608343436168051	0.265572757292878
0.225	0.423475087311445	0.184868348856606
0.25	0.294786035176184	0.12868905213526
0.275	0.20520405836999	0.0895819768061945
0.3	0.142844980924375	0.0623590774456151
0.325	0.0994360868755084	0.0434088940488664
0.35	0.0692186404389528	0.0302174464365557
0.375	0.0481839172755838	0.021034723163369
0.4	0.0335413967870091	0.0146425204885747
0.425	0.0233485644595688	0.0101928323274403
0.45	0.0162532128815185	0.0070953515780503
0.475	0.0113140544220336	0.00493915845948489
0.5	0.00787584758766835	0.00343820683436524
0.525	0.00548247099672494	0.00239337659094341
0.55	0.00381641314097791	0.00166605785574703
0.575	0.00265665049050678	0.00115976265047113
0.6	-33.7908037048612	33.7934603553517

Table 6:Derivative-Three point midpoint method-case3($\Delta t = 0.025$)

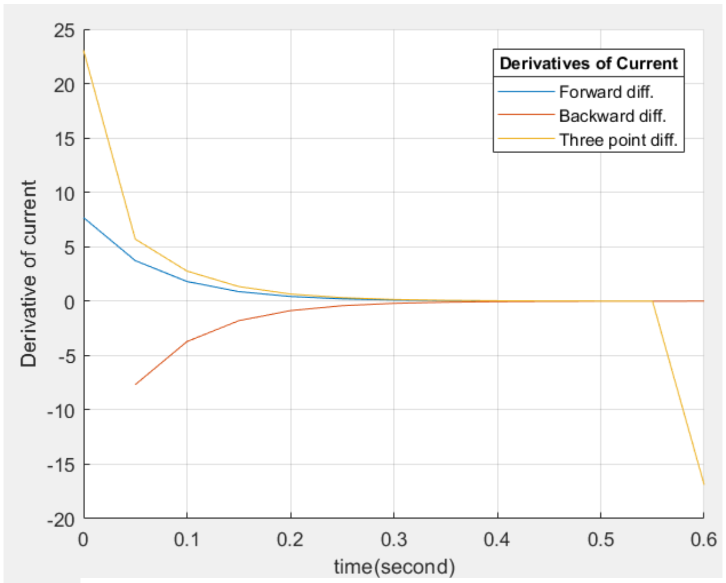
Table 4 has $\Delta t = 0.075$ second time step size, Table 5 has $\Delta t = 0.050$ second time step size, Table 6 has $\Delta t = 0.025$ second time step size. As expected, converge results in table 6 has most accurate answer. Obviously it has more iteration but if we look at first seven iteration in every case, the less time step size is improve converge rate even it was slightly. I can not put there table for the case 4 which has time step $\Delta t = 0.01$, because it is too long to capture.

III. Derivative Graphs case by case

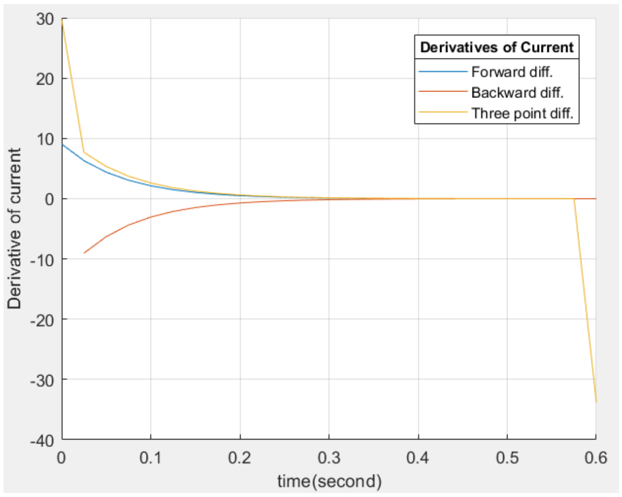
This section has similar results and comments with section one, Only difference is this section will Show us derivative graphs meanings unlikely to table data shown section one. All four case were used plotting graphs.



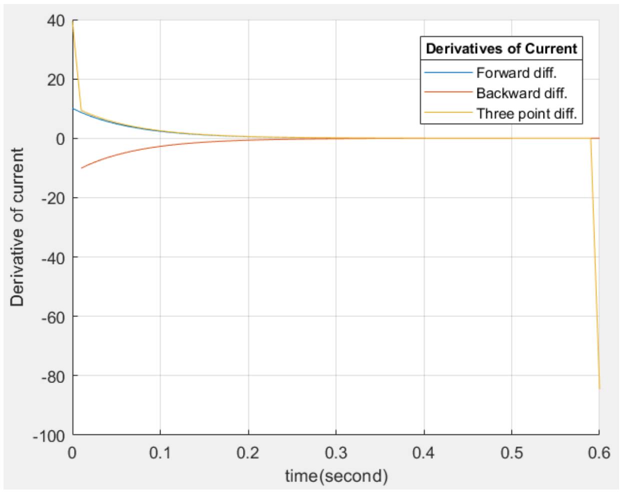
Graph 1: Derivative – time graph Case1($\Delta t = 0.075$)



Graph 2: Derivative – time graph Case2($\Delta t = 0.050$)



Graph 3: Derivative – time graph Case3($\Delta t = 0.025$)



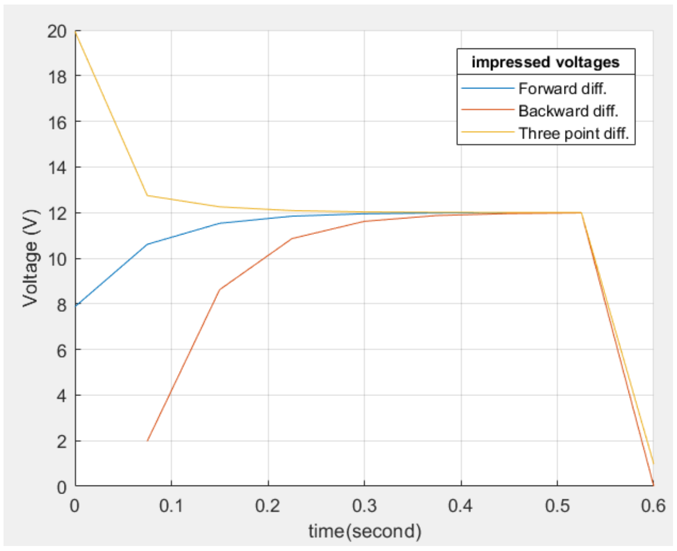
Graph 4: Derivative – time graph Case4($\Delta t = 0.010$)

All four graphs, graph 1, graph 2, graph 3, graph 4, are located above. From graph 1 we can say that three point midpoint method's line has enormous initial value and the last point also diving through below so fast. Forward method converges little bit more than backward method according to graph 1. There might be misunderstanding because it looks like forward method converges faster than three point midpoint method. However, end point formulas are made three point method's initial derivative value wrong. If we skip that point and bring the midpoint method's graph close to y-axis, clearly midpoint converges better. According to my observations Midpoint method is the best convergence method and forward method is slightly better than backward method. I explained the details in section one.

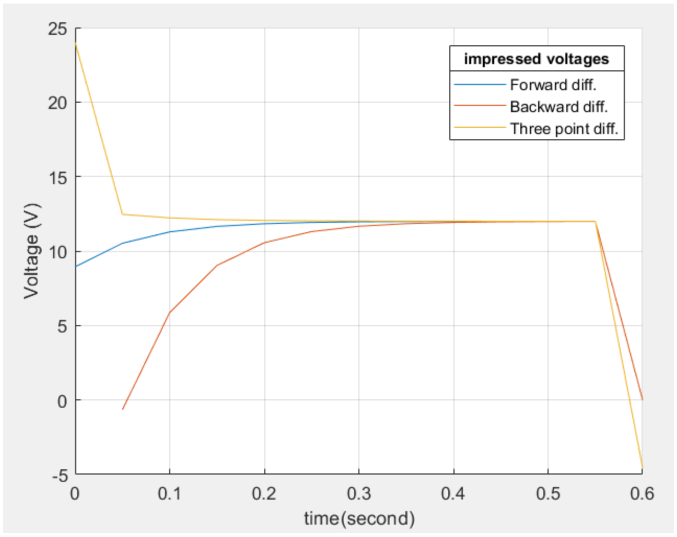
IV. Impressed Voltage Graphs case by case

We want to find most accurate way to calculate impressed voltage with just knowing Current values by time , Inductance and resistance value. $\varepsilon_{(t)} = I.R + I'.L$ is the basic form of the solution. We already calculated first derivative of current. Only term that is chanced is current. I expecting similar results like derivativeof current by time graphs. There is five graphs located below.

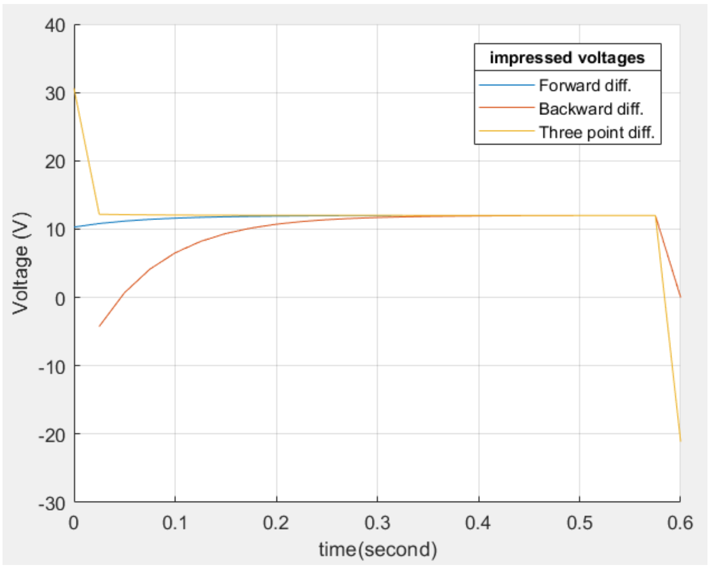
Graph 5: Voltage – time graph Case1($\Delta t = 0.075$)



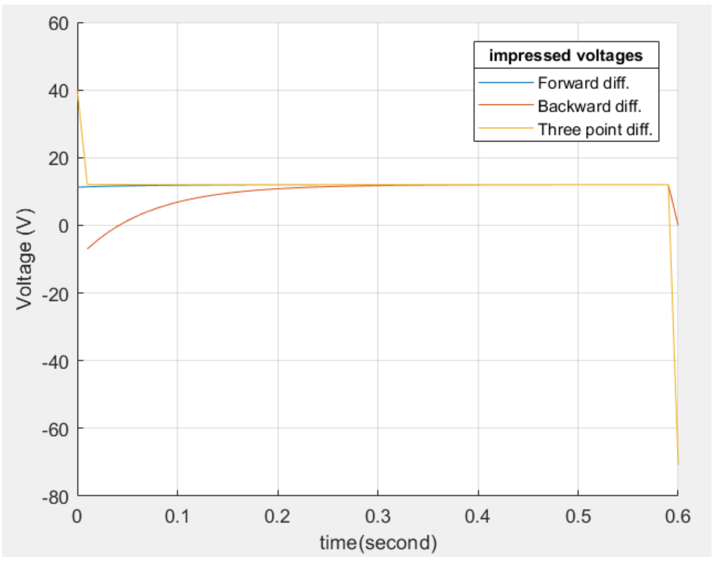
Graph 6: Voltage – time graph Case2($\Delta t = 0.050$)



Graph 7: Voltage – time graph Case3($\Delta t = 0.025$)



Graph 8: Voltage – time graph Case4($\Delta t = 0.010$)



The graph 5 is show us three point midpoint method converges very fast and obviously better converge rate than other methods. Forward method is converges much more than backward method. Every result so far is similar to expected. Exact impressed voltage value should be around twelve volt according to graphs.

V. Explaining of Matlab code

C1,C2,C3,C4 variables used for imported time and current data (line5-8). D_F(1,2,3,4) is used for keep derivatives of current by Forward method. D_B(1,2,3,4) is used for keep derivatives of current by backward method. D_T(1,2,3,4) is used for keep derivatives of current by threepoint method. Forward difference method algorithm is between lines 14-37. In line 16, program assign non-value to last element of array because of the endpoint's derivative cannot be taken at this point. At this part of code, Every derivative is keeping by an array. Δt or h value is found by extraction between time values to prevent errors. Backward method algorithm is between lines 39-62. İt is same as forward method but this time program makes first element empty. h value is negative too. Three point midpoint algorithm is between lines 63-110. In this part program firstly finds h value. then in the loop, Program decide where to use which endpoint formula by conrolling i value. If its not one of the endpoints then it apply midpoint formula. Between lines 114-166, program calculating impressed voltage by already prepared derivative and current arrays. In this part program calculate voltage values for each method and keeping them in an array too. Between lines 166 to 191 program plotting derivative of current and impressed voltage graphics. Program creates two different figure. İt has label,title,legend,axis limits and grids. Last part is for output table which means data of derivatives and convergence rates. Table part is between lines 192-232. Graphs are from all four cases and showing converges of different methods. Tables are showing error values basically. That's all we have.

VI. Analysis and Assumptions

To consider all of these data and graphs, I can guess impressed voltage value as 12 volt because all voltage plot lines goes to this value approximately. If we want to create a function for impressed voltage, $\varepsilon_{(t)} = I.R + I'.L$ should be used. Cause its all we have to. I cannot say exact function about impressed voltage its too difficulty to me to find.

In this report, I observe all my expectations in both way correctly and similarly. Three point end point formula caused some errors. But except from that, all methods work fine and close to expected. Three point midpoint method is the most accurately converges one because it is second order method unlikely to forward and backward methods. Forward difference method is little bit different than backward difference method. Unused endpoints are different but rest are the same. In all methods, converges rates are quite slow at first iterations but then they are getting better convergences. If end point not used, midpoint method will be much more accurate totally. This end point formula do nothing except harmful calculations.

Matlab Code

```matlab
clear all;
close all;
format long;

C1=importdata('current1.dat');
C2=importdata('current2.dat');
C3=importdata('current3.dat');
C4=importdata('current4.dat');
L=0.98;
R=14.2;

%---------------Derivatives of current--------------------

%---------------Forward difference-C1---------------------
D_F1=zeros(1,length(C1)-1);
D_F1(length(C1))=nan;
for i=1:1:length(C1)-1
   D_F1(i)=((C1(i+1,2))-(C1(i,2)))/((C1(i+1,1))-(C1(i,1)));
end
%---------------Forward difference-C2---------------------
D_F2=zeros(1,length(C2)-1);
D_F2(length(C2))=nan;
for i=1:1:length(C2)-1
    D_F2(i)=((C2(i+1,2))-(C2(i,2)))/((C2(i+1,1))-(C2(i,1)));
end
%---------------Forward difference-C3---------------------
D_F3=zeros(1,length(C3)-1);
D_F3(length(C3))=nan;
for i=1:1:length(C3)-1
    D_F3(i)=((C3(i+1,2))-(C3(i,2)))/((C3(i+1,1))-(C3(i,1)));
end
%---------------Forward difference-C4---------------------
D_F4=zeros(1,length(C4)-1);
D_F4(length(C4))=nan;
for i=1:1:length(C4)-1
    D_F4(i)=((C4(i+1,2))-(C4(i,2)))/((C4(i+1,1))-(C4(i,1)));
end

%--------------Backward difference-C1---------------------
D_B1=zeros(1,length(C1));
D_B1(1)=nan;
for i=1:1:length(C1)-1
    D_B1(i+1)=((C1(i+1,2))-(C1(i,2)))/((C1(i,1))-(C1(i+1,1)));
end
%--------------Backward difference-C2---------------------
D_B2=zeros(1,length(C2));
D_B2(1)=nan;
for i=1:1:length(C2)-1
    D_B2(i+1)=((C2(i+1,2))-(C2(i,2)))/((C2(i,1))-(C2(i+1,1)));
end
%--------------Backward difference-C3---------------------
D_B3=zeros(1,length(C3));
D_B3(1)=nan;
for i=1:1:length(C3)-1
    D_B3(i+1)=((C3(i+1,2))-(C3(i,2)))/((C3(i,1))-(C3(i+1,1)));
end
%--------------Backward difference-C4---------------------
D_B4=zeros(1,length(C4));
D_B4(1)=nan;
```

```
for i=1:1:length(C4)-1
    D_B4(i+1)=((C4(i+1,2))-(C4(i,2)))/((C4(i,1))-(C4(i+1,1)));
end
%-------------Three point-C1-------------------------------
D_T1=zeros(1,length(C1));
for i=1:1:length(C1)
h=C1(2,1)-C1(1,1);
    if i==1
       D_T1(i)=(1/(2*h))*(((-3)*C1(i,2))+(4*C1(i+1,2))-((-1)*C1(i+2,2)));
    elseif i==length(C1)
       D_T1(i)=(1/(-2*h))*(((-3)*C1(i,2))+(4*C1(i-1,2))-((-1)*C1(i-2,2)));
    else
        D_T1(i)=(1/(2*h))*(C1(i+1,2)-C1(i-1,2));
    end
end
%-------------Three point-C2-------------------------------
D_T2=zeros(1,length(C2));
for i=1:1:length(C2)
h=C2(2,1)-C2(1,1);
    if i==1
       D_T2(i)=(1/(2*h))*(((-3)*C2(i,2))+(4*C2(i+1,2))-((-1)*C2(i+2,2)));
    elseif i==length(C2)
       D_T2(i)=(1/(-2*h))*(((-3)*C2(i,2))+(4*C2(i-1,2))-((-1)*C2(i-2,2)));
    else
        D_T2(i)=(1/(2*h))*(C2(i+1,2)-C2(i-1,2));
    end
end
%-------------Three point-C3-------------------------------
D_T3=zeros(1,length(C3));
for i=1:1:length(C3)
h=C3(2,1)-C3(1,1);
    if i==1
       D_T3(i)=(1/(2*h))*(((-3)*C3(i,2))+(4*C3(i+1,2))-((-1)*C3(i+2,2)));
    elseif i==length(C3)
       D_T3(i)=(1/(-2*h))*(((-3)*C3(i,2))+(4*C3(i-1,2))-((-1)*C3(i-2,2)));
    else
        D_T3(i)=(1/(2*h))*(C3(i+1,2)-C3(i-1,2));
    end
end
%-------------Three point-C4-------------------------------
D_T4=zeros(1,length(C4));
for i=1:1:length(C4)
h=C4(2,1)-C4(1,1);
    if i==1
       D_T4(i)=(1/(2*h))*(((-3)*C4(i,2))+(4*C4(i+1,2))-((-1)*C4(i+2,2)));
    elseif i==length(C4)
       D_T4(i)=(1/(-2*h))*(((-3)*C4(i,2))+(4*C4(i-1,2))-((-1)*C4(i-2,2)));
    else
        D_T4(i)=(1/(2*h))*(C4(i+1,2)-C4(i-1,2));
    end
end

%------------Voltages------------------------------

%-----------Voltage-C1-----------------------------
V_F1=zeros(1,length(C1));
for i=1:1:length(C1)
    V_F1(i)=(C1(i,2)*R)+(D_F1(i)*L);
end
V_B1=zeros(1,length(C1));
for i=1:1:length(C1)-1
```

```
    V_B1(i)=(C1(i,2)*R)+(D_B1(i)*L);
end
V_T1=zeros(1,length(C1));
for i=1:1:length(C1)
    V_T1(i)=(C1(i,2)*R)+(D_T1(i)*L);
end
%-----------Voltage-C2------------------------------
V_F2=zeros(1,length(C2));
for i=1:1:length(C2)
    V_F2(i)=(C2(i,2)*R)+(D_F2(i)*L);
end
V_B2=zeros(1,length(C2));
for i=1:1:length(C2)-1
    V_B2(i)=(C2(i,2)*R)+(D_B2(i)*L);
end
V_T2=zeros(1,length(C2));
for i=1:1:length(C2)
    V_T2(i)=(C2(i,2)*R)+(D_T2(i)*L);
end
%-----------Voltage-C3------------------------------
V_F3=zeros(1,length(C3));
for i=1:1:length(C3)
    V_F3(i)=(C3(i,2)*R)+(D_F3(i)*L);
end
V_B3=zeros(1,length(C3));
for i=1:1:length(C3)-1
    V_B3(i)=(C3(i,2)*R)+(D_B3(i)*L);
end
V_T3=zeros(1,length(C3));
for i=1:1:length(C3)
    V_T3(i)=(C3(i,2)*R)+(D_T3(i)*L);
end
%-----------Voltage-C4------------------------------
V_F4=zeros(1,length(C4));
for i=1:1:length(C4)
    V_F4(i)=(C4(i,2)*R)+(D_F4(i)*L);
end
V_B4=zeros(1,length(C4));
for i=1:1:length(C4)-1
    V_B4(i)=(C4(i,2)*R)+(D_B4(i)*L);
end
V_T4=zeros(1,length(C4));
for i=1:1:length(C4)
    V_T4(i)=(C4(i,2)*R)+(D_T4(i)*L);
end
%-----------------Graphs--------------------------
figure
grid on;
hold on;
plot(C1(:,1),D_F1);
plot(C1(:,1),D_B1);
plot(C1(:,1),D_T1);
ylabel('Derivative of current');
xlabel('time(second)');
xlim auto
ylim auto
a=legend('Forward diff.','Backward diff.','Three point diff.');
title(a,'Derivatives of Current');

figure
grid on;
```

```
hold on;
plot(C1(:,1),V_F1);
plot(C1(:,1),V_B1);
plot(C1(:,1),V_T1);
ylabel('Voltage (V)');
xlabel('time(second)');
xlim auto
ylim auto
a=legend('Forward diff.','Backward diff.','Three point diff.');
title(a,'impressed voltages');
%---------------Tables-------------------------------
%--------------Forward Difference Method-------------
T=zeros(length(C4),3);
for i=1:1:length(C4)
    T(i,1)=C4(i,1);
    T(i,2)=D_F4(i);
    if i>1
        T(i,3)=D_F4(i-1)-D_F4(i);
    end
end
Forward_method=array2table(T);
Forward_method.Properties.VariableNames={'Time' 'Derivative_of_current'
'error_convergence'};
display('Forward Difference Method Converge Table');
display(Forward_method);
%--------------Backward Difference Method-------------
T2=zeros(length(C4),3);
for i=1:1:length(C4)
    T2(i,1)=C4(i,1);
    T2(i,2)=D_B4(i);
    if i>1
        T2(i,3)=D_B4(i-1)-D_B4(i);
    end
end
Backward_method=array2table(T2);
Backward_method.Properties.VariableNames={'Time' 'Derivative_of_current'
'error_convergence'};
display('Backward Difference Method Converge Table');
display(Backward_method);
%-------------Three point mid/end point Method----------
T3=zeros(length(C4),3);
for i=1:1:length(C4)
    T3(i,1)=C4(i,1);
    T3(i,2)=D_T4(i);
    if i>1
        T3(i,3)=D_T4(i-1)-D_T4(i);
    end
end
Three_point_method=array2table(T3);
Three_point_method.Properties.VariableNames={'Time' 'Derivative_of_current'
'error_convergence'};
display('Three point mid/end point Method Converge Table');
display(Three_point_method);
%-----------------------------------------------------
```