Using symbolic math tool box and laplace's transform to solve real life problems

Parmeet Singh(2K19/EP/066) and Pankaj Gupta(2K19/EP/064)

November 11, 2020

1 Prerequisites- software's used

1.MATLAB
2.MATLAB app designer
3.Symbolic math tool box
4.Latex

2 Symbolic Math tool Box

Mathematical equations can be found everywhere in real world from simple pendulums to heat flow equations. They require a lot of algebra, calculus and differential equations. Solving them using hand can be tedious work and that is where MATLAB comes in. But even in MATLAB, we need different tool box's for easier and better use. Symbolic math tool box helps a great deal.
It can be used to solve equations(ordinary differential and algebraic) perform differentiation, integration, simplification, transforms.

2.1 some functions that we will be using

1.subs(s,old,new) - return a copy of s with the old variables replaced with new ones.
2.syms a b - creates symbolic variables a and b.
3.diff(a,b) - calculates the difference between adjacent elements of a , b number of times.
4.solve(eqn,var)-solves the equations for these defined variables.
5.laplace(y,s,t)-gives the laplace transform of y.
6.ilaplace(y,s,t)-gives the inverse laplace transform of y.

3 Laplace transform for solving differential equations

Laplace transform is defined as

$$F(s) = \int_0^\infty f(t)e^{-st}\,dx \tag{1}$$

it can be used to solve differential equations.
Let's say we have a differential equation:

$$\frac{dy}{dx} = Ky(x) + C \tag{2}$$

where K and C are both two constants.
Taking Laplace transform on both the sides we get,

$$L(\frac{dy}{dx}) = K.L(y) + L(C) \tag{3}$$

and we know that

$$L(\frac{dy}{dx}) = L(y') = L(y)s - y(0)$$

by substituting the value of $L(y')$ in the equation number (3) we will get the value of $L(y)$ on which we can then use inverse laplace transform for getting the value of y.
We can use this method to solve the differential equation inside of MATLAB too.

4 GUI used in our program

Graphical user interfaces (GUIs), which is also called apps. It eliminates the need for other to learn MATLAB language in order to run the program. We have created an interactive app using MATLAB App Designer to simulate various real life problems which can be solved using laplace transform.

We have used **callback functions** in our app.A callback is a function that executes when a user interacts with a UI component in your app.A callback can be created using several ways.

We have used the right click method, Right-click a component in the canvas, Component Browser, or App Layout pane, and select Callbacks > Add (callback property) callback.

5 Applications of Laplace Transform in real life scenarios

Differential equations can be seen in a lot of real life situations. Here we will be using MATLAB and its symbolic math toolbox along with the above mentioned concept of Laqplace transform for solving three of such physical situations. These three situations are Bacteria decay, population growth and solving the RLC circuits found in electrical problems.

5.1 Increase in population

In 1798 British scientist Thomas Robert Malthus proposed a model for the population growth.It reflects the exponential growth in the population, it can be stated as

$$\frac{dN}{dt} = kN \tag{4}$$

where k is know as the growth constant(Malthusian Parameter).
We will be first solving the equation and then plotting the graph.

5.2 Radioactive decay

Let's say we have a radioactive element and it is decaying then it's decay can be expressed as a differential equation

$$\frac{dR}{dt} = -kR \tag{5}$$

this equation was derived by F.Soddy and E.Rutherford. Where R is the amount of Radioactive element and k is the radioactive decay constant.The negative sign shows that R is decreasing over time. We can also find the **half life** of the radioactive element it is given by

$$T = \frac{1}{k} ln(2)$$

Just like the example of Increase in population we will be doing the same thing by solving the equation and then showing the graph. We will also show the half life of the radioactive compound.

5.2.1 GUI based application using MATLAB app designer - Part A

```
classdef LAPLACETRANSFORM < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        UIFigure                     matlab.ui.Figure
        GridLayout                   matlab.ui.container.GridLayout
        axis                         matlab.ui.control.UIAxes
```

```
    ChoicesButtonGroup            matlab.ui.container.ButtonGroup
    PopulationGrowthButton        matlab.ui.control.RadioButton
    RadioactiveDecayButton        matlab.ui.control.RadioButton
    KEditField                    matlab.ui.control.NumericEditField
    KEditFieldLabel               matlab.ui.control.Label
    InitialPopulationEditField  matlab.ui.control.NumericEditField
    InitialPopulationEditFieldLabel  matlab.ui.control.Label
    PlotButton                    matlab.ui.control.Button
    ContextMenu                   matlab.ui.container.ContextMenu
    Menu                          matlab.ui.container.Menu
    Menu2                         matlab.ui.container.Menu
    ContextMenu2                  matlab.ui.container.ContextMenu
    Menu_2                        matlab.ui.container.Menu
    Menu2_2                       matlab.ui.container.Menu
end


% Callbacks that handle component events
methods (Access = private)

    % Button pushed function: PlotButton
    function PlotButtonPushed(app, event)
        % solving the Population Growth Problem
        % equation using the laplace transfrom method to solve the following
        % differential equation
        selectedButton = app.ChoicesButtonGroup.SelectedObject;
        if app.PopulationGrowthButton == selectedButton
            % taking inputs from user
            a = app.KEditField.Value;
            b = app.InitialPopulationEditField.Value;

            %just defining the symbolic variables
            syms N(t) s t k

            %Now we define the differential equation dN/dt=KN
            dN= diff(N,t);
            eqn1= dN == k*N;

            %defining the initial condition
             cond1 = (N(0)==0);

            %calculting the laplace transform of the eqn
             eqn1LT=laplace(eqn1,t,s);
```

```
                %this will break itself into smaller laplace differential to normal differne
                syms N_lt
                eqn1LT = subs(eqn1LT,[laplace(N,t,s)],[N_lt]);

                %solving the above equations
                N_lt=solve(eqn1LT,N_lt)

                %we will get the inverse laplace transform of the above equation
                N_soln=ilaplace(N_lt,s,t);
                N_soln=simplify(N_soln);
                vars = [k N(0)];
                values = [a b];
                N_soln = subs(N_soln,vars,values)
                 % we are now gonna plot the function that we want
                 fplot(app.axis,N_soln,[1 10],'w*')

           elseif app.RadioactiveDecayButton == selectedButton
                % taking inputs from user
               a = app.KEditField.Value;
               b = app.InitialPopulationEditField.Value;

                %just defining the symbolic variables
               syms N(t) s t k
               dN= diff(N,t);
               eqn1= dN==(-1)*k*N;
%defining the initial condition
cond1 = (N(0)==0);
%calculting the laplace transform of the eqn
eqn1LT=laplace(eqn1,t,s);
%this will break itself into smaller laplace differential to normal differnetial
syms N_lt
eqn1LT = subs(eqn1LT,[laplace(N,t,s)],[N_lt]);
%solving the above equations
N_lt=solve(eqn1LT,N_lt)
%we will get the inverse laplace transform of the above equation
N_soln=ilaplace(N_lt,s,t);
N_soln=simplify(N_soln);
vars = [k N(0)];
values = [a b];
N_soln = subs(N_soln,vars,values)
% we are now gonna plot the function that we want
fplot(app.axis,N_soln,[1 10],'wo')
end
end

% Callback function
```

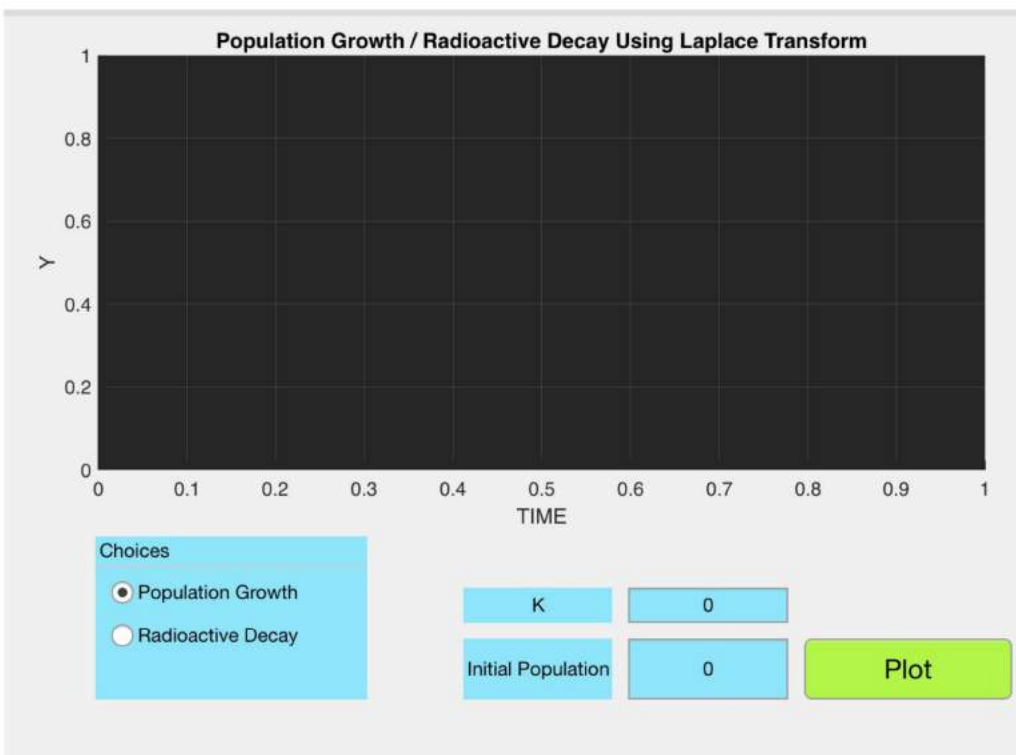
```
function ChoiceButtonGroupSelectionChanged(app, event)
selectedButton = app.ChoicesButtonGroup.SelectedObject;
end
end

% Component initialization
methods (Access = private)

% Create UIFigure and components
function createComponents(app)

% Create UIFigure and hide until all components are created
app.UIFigure = uifigure('Visible', 'off');
app.UIFigure.Position = [100 100 640 480];
app.UIFigure.Name = 'MATLAB App';

% Create GridLayout
app.GridLayout = uigridlayout(app.UIFigure);
app.GridLayout.ColumnWidth = {'1x', 109, '1.39x', 41, '2.53x', 100, 130};
app.GridLayout.RowHeight = {'13.59x', '1x', 22, 38, 33};

% Create axis
app.axis = uiaxes(app.GridLayout);
title(app.axis, 'Population Growth / Radioactive Decay Using Laplace Transform')
xlabel(app.axis, 'TIME')
ylabel(app.axis, 'Y')
zlabel(app.axis, 'Z')
app.axis.PlotBoxAspectRatio = [2.13991769547325 1 1];
app.axis.Color = [0.149 0.149 0.149];
app.axis.XGrid = 'on';
app.axis.YGrid = 'on';
app.axis.GridColor = [0.9412 0.9412 0.9412];
app.axis.Layout.Row = 1;
app.axis.Layout.Column = [1 7];

% Create ChoicesButtonGroup
app.ChoicesButtonGroup = uibuttongroup(app.GridLayout);
app.ChoicesButtonGroup.BorderType = 'none';
app.ChoicesButtonGroup.Title = 'Choices ';
app.ChoicesButtonGroup.BackgroundColor = [0.5608 0.902 0.9804];
app.ChoicesButtonGroup.Layout.Row = [2 4];
app.ChoicesButtonGroup.Layout.Column = [2 3];

% Create PopulationGrowthButton
app.PopulationGrowthButton = uiradiobutton(app.ChoicesButtonGroup);
app.PopulationGrowthButton.Text = 'Population Growth';
```

```
app.PopulationGrowthButton.Position = [11 57 122 22];
app.PopulationGrowthButton.Value = true;

% Create RadioactiveDecayButton
app.RadioactiveDecayButton = uiradiobutton(app.ChoicesButtonGroup);
app.RadioactiveDecayButton.Text = 'Radioactive Decay';
app.RadioactiveDecayButton.Position = [11 30 122 22];

% Create KEditField
app.KEditField = uieditfield(app.GridLayout, 'numeric');
app.KEditField.ValueDisplayFormat = '%19.4g';
app.KEditField.HorizontalAlignment = 'center';
app.KEditField.BackgroundColor = [0.5608 0.902 0.9804];
app.KEditField.Layout.Row = 3;
app.KEditField.Layout.Column = 6;

% Create KEditFieldLabel
app.KEditFieldLabel = uilabel(app.GridLayout);
app.KEditFieldLabel.BackgroundColor = [0.5608 0.902 0.9804];
app.KEditFieldLabel.HorizontalAlignment = 'center';
app.KEditFieldLabel.Layout.Row = 3;
app.KEditFieldLabel.Layout.Column = 5;
app.KEditFieldLabel.Text = 'K';

% Create InitialPopulationEditField
app.InitialPopulationEditField = uieditfield(app.GridLayout, 'numeric');
app.InitialPopulationEditField.ValueDisplayFormat = '%19.4g';
app.InitialPopulationEditField.HorizontalAlignment = 'center';
app.InitialPopulationEditField.BackgroundColor = [0.5608 0.902 0.9804];
app.InitialPopulationEditField.Layout.Row = 4;
app.InitialPopulationEditField.Layout.Column = 6;

% Create InitialPopulationEditFieldLabel
app.InitialPopulationEditFieldLabel = uilabel(app.GridLayout);
app.InitialPopulationEditFieldLabel.BackgroundColor = [0.5608 0.902 0.9804];
app.InitialPopulationEditFieldLabel.HorizontalAlignment = 'center';
app.InitialPopulationEditFieldLabel.Layout.Row = 4;
app.InitialPopulationEditFieldLabel.Layout.Column = 5;
app.InitialPopulationEditFieldLabel.Text = 'Initial Population';

% Create PlotButton
app.PlotButton = uibutton(app.GridLayout, 'push');
app.PlotButton.ButtonPushedFcn = createCallbackFcn(app, @PlotButtonPushed, true);
app.PlotButton.BackgroundColor = [0.702 0.9608 0.2863];
app.PlotButton.FontSize = 17;
app.PlotButton.Layout.Row = 4;
```

```
app.PlotButton.Layout.Column = 7;
app.PlotButton.Text = 'Plot';

% Create ContextMenu
app.ContextMenu = uicontextmenu(app.UIFigure);
% Assign app.ContextMenu

app.PopulationGrowthButton.ContextMenu = app.ContextMenu;

% Create Menu
app.Menu = uimenu(app.ContextMenu);
app.Menu.Text = 'Menu';

% Create Menu2
app.Menu2 = uimenu(app.ContextMenu);
app.Menu2.Text = 'Menu2';

% Create ContextMenu2
app.ContextMenu2 = uicontextmenu(app.UIFigure);

% Create Menu_2
app.Menu_2 = uimenu(app.ContextMenu2);
app.Menu_2.Text = 'Menu';

% Create Menu2_2
app.Menu2_2 = uimenu(app.ContextMenu2);
app.Menu2_2.Text = 'Menu2';

% Show the figure after all components are created
app.UIFigure.Visible = 'on';
end
end

% App creation and deletion
methods (Access = public)


% Construct app

function app = LAPLACETRANSFORM

% Create UIFigure and components
createComponents(app)

% Register the app with App Designer
registerApp(app, app.UIFigure)
```

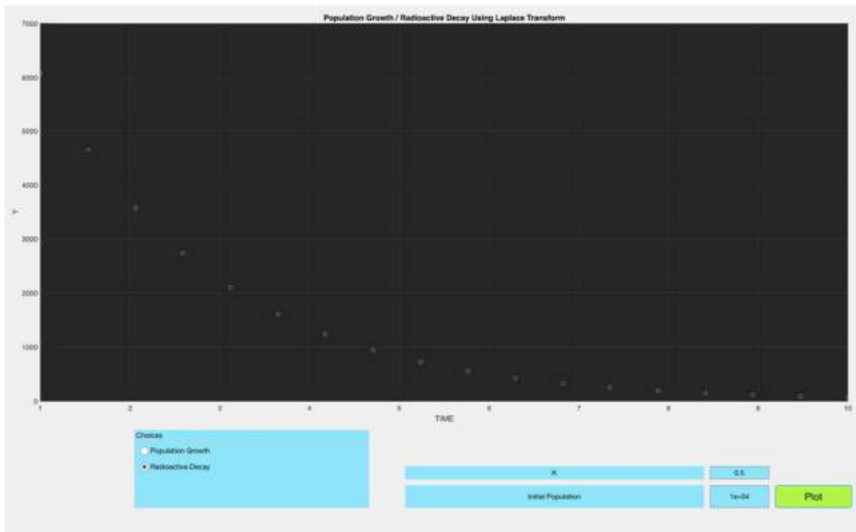
```
if nargout == 0
clear app
end

end

% Code that executes before app deletion
function delete(app)

% Delete UIFigure when app is deleted
delete(app.UIFigure)
end
end
end
```

5.2.2 Preview of our GUI based application - Part A

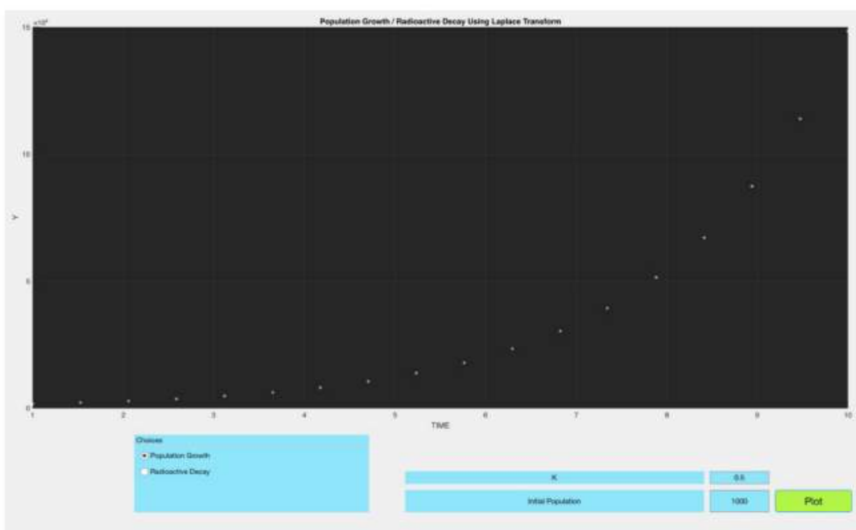


5.2.3 Final output for our app- Part A

Radioacitve Decay



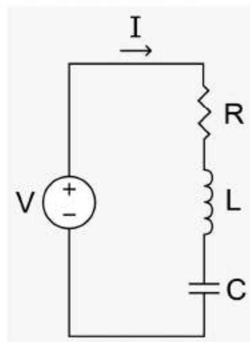
Population growth



6 RLC Circuit

We can also solve the differential equations that we get for the RLC circuit. We can get these equations using Kirchioff's Current and Potential law.

We will show one example here-



The differential equation that we get for the following circuit is-

$$V - IR - L\frac{dI}{dt} - \frac{Q}{C} = O \tag{6}$$

and

$$\frac{dQ}{dt} = I \tag{7}$$

now we will be using MATLAB again to solve these equations.

6.1 GUI based application using app designer - Part B

```
classdef LCR < matlab.apps.AppBase

   % Properties that correspond to app components
   properties (Access = public)
       UIFigure                      matlab.ui.Figure
       ResistanceEditFieldLabel      matlab.ui.control.Label
       ResistanceEditField           matlab.ui.control.NumericEditField
       CapacitanceEditFieldLabel     matlab.ui.control.Label
       CapacitanceEditField          matlab.ui.control.NumericEditField
       ImpedenceEditFieldLabel       matlab.ui.control.Label
       ImpedenceEditField            matlab.ui.control.NumericEditField
       VoltageEditFieldLabel         matlab.ui.control.Label
       VoltageEditField              matlab.ui.control.NumericEditField
       CurrentEditFieldLabel         matlab.ui.control.Label
       CurrentEditField              matlab.ui.control.NumericEditField
       ChargeEditFieldLabel          matlab.ui.control.Label
       ChargeEditField               matlab.ui.control.NumericEditField
       PlotButton                    matlab.ui.control.Button
       Axis1                         matlab.ui.control.UIAxes
```

```
    Axis2                       matlab.ui.control.UIAxes
end

% Callbacks that handle component events
methods (Access = private)

    % Button pushed function: PlotButton
    function PlotButtonPushed(app, event)

        %now we are trying to do solve an RLC circuit using the same method


        %taking the input from the user

        a= app.ResistanceEditField.Value;
        b= app.CapacitanceEditField.Value;
        c= app.ImpedenceEditField.Value;
        d= app.VoltageEditField.Value;
        e= app.CurrentEditField.Value;
        f= app.ChargeEditField.Value;

        %defining the variables

        syms L C I(t) Q(t) s V R

        %defining the equation

        dI = diff(I,t);
        dQ = diff(Q,t);
        eq1 = (V-I*R-L*dI-Q/C == 0)
        eq2 = (dQ == I);


        %defining the initial conditions

        c1 = (I(0)  == 0)
        c2 = (Q(0) == 0)

        %calculate the laplace transform

        eq1lt = laplace(eq1,t,s);
        eq2lt = laplace(eq2,t,s);

        %substituting the equations

        syms I_LT Q_LT
```

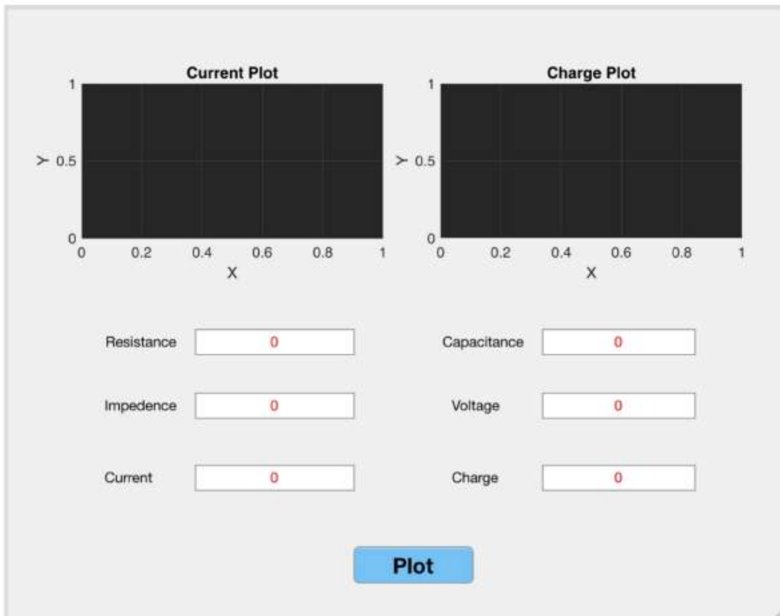
```
            eq1lt = subs(eq1lt,[laplace(I,t,s) laplace(Q,t,s)],[I_LT Q_LT])
            eq2lt = subs(eq2lt,[laplace(I,t,s) laplace(Q,t,s)],[I_LT Q_LT])

            %further solving

            eqns = [eq1lt eq2lt];
            var = [I_LT Q_LT];
            [I_LT, Q_LT] = solve(eqns,var);

            % calculating the inverse laplace and then simplifying

            I_solution = ilaplace(I_LT,s,t);
            Q_solution = ilaplace(Q_LT,s,t);
            I_solution = simplify(I_solution);
            Q_solution = simplify(Q_solution);

%giving the values to all the variables when we are solving further
var = [R L C I(0) Q(0) V];
value = [a b c e f d ];
I_solution = subs(I_solution,var,value);
Q_solution = subs(Q_solution,var,value);
%we will now plot the values for I(t) and Q(t)
fplot(app.Axis2,Q_solution,[1 100],'w*')
fplot(app.Axis1,I_solution,[1 100],'wo')
end
end

% Component initialization
methods (Access = private)

% Create UIFigure and components
function createComponents(app)


app.UIFigure.Position = [100 100 651 512];
app.UIFigure.Name = 'MATLAB App';

% Create ResistanceEditFieldLabel
app.ResistanceEditFieldLabel = uilabel(app.UIFigure);
app.ResistanceEditFieldLabel.HorizontalAlignment = 'right';
app.ResistanceEditFieldLabel.Position = [77 225 65 22];
app.ResistanceEditFieldLabel.Text = 'Resistance';

% Create ResistanceEditField
app.ResistanceEditField = uieditfield(app.UIFigure, 'numeric');
app.ResistanceEditField.HorizontalAlignment = 'center';
```

```
app.ResistanceEditField.FontColor = [1 0 0];
app.ResistanceEditField.Position = [157 225 133 22];

% Create CapacitanceEditFieldLabel
app.CapacitanceEditFieldLabel = uilabel(app.UIFigure);
app.CapacitanceEditFieldLabel.HorizontalAlignment = 'right';
app.CapacitanceEditFieldLabel.Position = [358 225 73 22];
app.CapacitanceEditFieldLabel.Text = 'Capacitance';

% Create CapacitanceEditField
app.CapacitanceEditField = uieditfield(app.UIFigure, 'numeric');
app.CapacitanceEditField.HorizontalAlignment = 'center';
app.CapacitanceEditField.FontColor = [1 0 0];
app.CapacitanceEditField.Position = [446 225 128 22];

% Create ImpedenceEditFieldLabel
app.ImpedenceEditFieldLabel = uilabel(app.UIFigure);
app.ImpedenceEditFieldLabel.HorizontalAlignment = 'right';
app.ImpedenceEditFieldLabel.Position = [76 172 66 22];
app.ImpedenceEditFieldLabel.Text = 'Impedence';

% Create ImpedenceEditField
app.ImpedenceEditField = uieditfield(app.UIFigure, 'numeric');
app.ImpedenceEditField.HorizontalAlignment = 'center';
app.ImpedenceEditField.FontColor = [1 0 0];
app.ImpedenceEditField.Position = [157 172 133 22];

% Create VoltageEditFieldLabel
app.VoltageEditFieldLabel = uilabel(app.UIFigure);
app.VoltageEditFieldLabel.HorizontalAlignment = 'right';
app.VoltageEditFieldLabel.Position = [366 172 45 22];
app.VoltageEditFieldLabel.Text = 'Voltage';

% Create VoltageEditField
app.VoltageEditField = uieditfield(app.UIFigure, 'numeric');
app.VoltageEditField.HorizontalAlignment = 'center';
app.VoltageEditField.FontColor = [1 0 0];
app.VoltageEditField.Position = [446 172 128 22];

% Create CurrentEditFieldLabel
app.CurrentEditFieldLabel = uilabel(app.UIFigure);
app.CurrentEditFieldLabel.HorizontalAlignment = 'right';
app.CurrentEditFieldLabel.Position = [76 112 46 22];
app.CurrentEditFieldLabel.Text = 'Current';

% Create CurrentEditField
```

```
app.CurrentEditField = uieditfield(app.UIFigure, 'numeric');
app.CurrentEditField.HorizontalAlignment = 'center';
app.CurrentEditField.FontColor = [1 0 0];
app.CurrentEditField.Position = [157 112 133 22];

% Create ChargeEditFieldLabel
app.ChargeEditFieldLabel = uilabel(app.UIFigure);
app.ChargeEditFieldLabel.HorizontalAlignment = 'right';
app.ChargeEditFieldLabel.Position = [366 112 44 22];
app.ChargeEditFieldLabel.Text = 'Charge';

% Create ChargeEditField
app.ChargeEditField = uieditfield(app.UIFigure, 'numeric');
app.ChargeEditField.HorizontalAlignment = 'center';
app.ChargeEditField.FontColor = [1 0 0];
app.ChargeEditField.Position = [446 112 128 22];

% Create PlotButton
app.PlotButton = uibutton(app.UIFigure, 'push');
app.PlotButton.ButtonPushedFcn = createCallbackFcn(app, @PlotButtonPushed, true);
app.PlotButton.IconAlignment = 'center';
app.PlotButton.WordWrap = 'on';
app.PlotButton.BackgroundColor = [0.4627 0.7608 0.9608];
app.PlotButton.FontSize = 18;
app.PlotButton.FontWeight = 'bold';
app.PlotButton.Position = [289 35 100 31];
app.PlotButton.Text = 'Plot';

% Create Axis1
app.Axis1 = uiaxes(app.UIFigure);
title(app.Axis1, 'Current Plot')
xlabel(app.Axis1, 'X')
ylabel(app.Axis1, 'Y')
zlabel(app.Axis1, 'Z')
app.Axis1.PlotBoxAspectRatio = [1.94573643410853 1 1];
app.Axis1.Layer = 'top';
app.Axis1.Color = [0.149 0.149 0.149];
app.Axis1.XGrid = 'on';
app.Axis1.YGrid = 'on';
app.Axis1.ZGrid = 'on';
app.Axis1.GridColor = [0.9412 0.9412 0.9412];
app.Axis1.Position = [22 286 300 185];

% Create Axis2
app.Axis2 = uiaxes(app.UIFigure);
title(app.Axis2, 'Charge Plot')
```

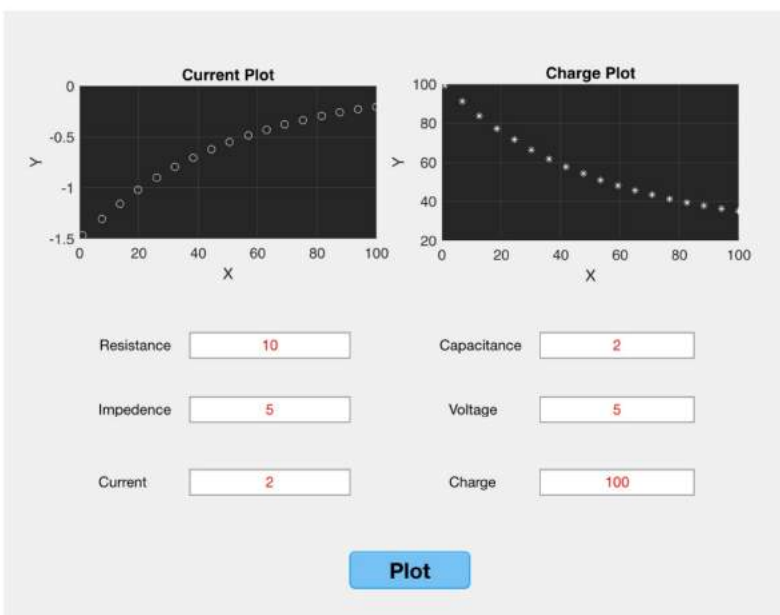
```
xlabel(app.Axis2, 'X')
ylabel(app.Axis2, 'Y')
zlabel(app.Axis2, 'Z')
app.Axis2.Color = [0.149 0.149 0.149];
app.Axis2.XGrid = 'on';
app.Axis2.YGrid = 'on';
app.Axis2.GridColor = [0.9412 0.9412 0.9412];
app.Axis2.Position = [321 286 300 185];

% Show the figure after all components are created
app.UIFigure.Visible = 'on';
end
end

% App creation and deletion
methods (Access = public)

% Construct app
function app = LCR

% Create UIFigure and components
createComponents(app)

% Register the app with App Designer
registerApp(app, app.UIFigure)

if nargout == 0
clear app
end
end

% Code that executes before app deletion
function delete(app)

% Delete UIFigure when app is deleted
delete(app.UIFigure)
end
end
end
```

6.1.1 Preview of our GUI based application - Part B



6.1.2 Final output for our app- Part B

RLC circuit



7 Further applications of Laplace Transform

Note- due to lack of experience and time constraints we have restricted our applications to these three only but, Laplace Transformation can be further used for these-

1. Analysis of electronic circuits: Laplace Transform is widely used by electronic engineers to solve quickly differential equations occurring in the analysis of electronic circuits.

2. System modeling: Laplace Transform is used to simplify calculations in system modeling, where large number of differential equations are used.

3. Digital signal processing: One can not imagine solving digital signal processing problems without employing Laplace Transform.

4. Nuclear Physics: In order to get the true form of radioactive decay a Laplace Transform is used. It makes easy to study analytic part of Nuclear physics possible.

5. Process Control: Laplace Transform is used for process controls. It helps to analyze the variables which when altered, produce desired manipulations in the result.

8 Bibliography

1.www.Mathworks.com(app designer section)
2.https://in.mathworks.com/help/matlab/creating_guis/callback_creation.png
3.https://www.google.com/url?sa=i&url=https%3A%2F%2Fen.wikiversity.org%2Fwiki%2FRLC_circuit&
4.https://in.mathworks.com/discovery/matlab-gui.html
5.https://in.mathworks.com/help/matlab/creating_guis/write-callbacks-for-gui-in-app-designer
6.https://in.mathworks.com/help/symbolic/laplace.html
7.https://in.mathworks.com/help/symbolic/solve-differential-equations-using-laplace-transfor