Virtual Capacitor

Team GIANTS capacitor design tool

Description

This MATLAB program is used to model capacitors. The user enters several values into a graphical-user-interface, which are then used to calculate specific capacitor parameters.

A capacitor is a fundamental electrical component that stores energy via the accumulation of charge. There are many varieties of capacitors, but this tool is currently limited to modeling two: the parallel plate capacitor, and the spherical capacitor.

Model Creation

Upon running the tool in the MATLAB IDE, a window will appear. This window contains three editable fields – in the first, the user enters the desired capacitance value (in Farads). In the second, the user provides the surface area of the capacitor's plates, in meters^2. The third field is reserved for setting the distance between the plates – this is in meters. Below these fields a dropdown menu is used to select the capacitor's dielectric material. After the fields have been populated, and once the dielectric material has been selected, the user clicks the 'Calculate' button in the bottom right corner of the window to create an instance of the capacitor object.

Code Breakdown

The code comprises of a main GUI and many classes that the main uses, most components are modeled through objects. There is a dielectric class that creates different objects representing the different dielectric materials and their respective properties. There are also multiple capacitor classes with the superclass representing a basic parallel plate capacitor and multiple subclasses for spherical and cylindrical capacitors. The main file runs the GUI and uses a radio button to allow the user to select the shape of the capacitor, based on the selected shape a different GUI is shown and the correct object based on the shape is created.

Classes

The model incorporates classes – one for each capacitor type (parallel plate and spherical), and one for the dielectric. The capacitor classes contain public variables that represent the properties of physical capacitors, and two methods that act upon those variables. The first is a simple constructor that passes along the user-entered values from the GUI window. The other is a 'solve' function that calculates the capacitor's remaining parameters. The dielectric class contains four public properties, and these are set by passing a dielectric variable into its constructor. The dielectric variable is an enumeration – we use a 'material _bank' file that contains a list of materials, each assigned a name and several properties. This set of materials comprises the set of possible values the dielectric variable can take on. (See code example on following page)

```
clear; clc;
addpath('Class_Definitions')
[bank_names, bank] = material_bank;

GUI(bank_names, bank)

function GUI(bank_names, bank)
    figGUI = uifigure('Name','Capacitor Design Tool','NumberTitle','off');
    fig = uigridlayout(figGUI, [6,3]);

    cap_shape = uibuttongroup(fig);
    cap_shape.Layout.Row = 1;
    cap_shape.Layout.Column = [1 3];

    p = uiradiobutton(cap_shape, "Text", "Parallel Plate", "Position", [150
10 91 22], "Value", 1, "Tag", "plate");
    s = uiradiobutton(cap_shape, "Text", "Spherical", "Position", [300 10 91
22], "Value", 0, "Tag", "sphere");
    c = uiradiobutton(cap_shape, "Text", "Cylindrical", "Position", [450 10
91 22], "Value", 0, "Tag", "cylinder");

    defaultData = struct;
    defaultData.OldValue=p;
    defaultData.NewValue=p;
    defaultData.Source=cap_shape;
    defaultData.EventName='SelectionChanged';

    shape_SelectionChangedFcn(cap_shape, defaultData);

    set(cap_shape, 'SelectionChangedFcn', @shape_SelectionChangedFcn);

    function shape_SelectionChangedFcn(hObject, eventData)
```

```
        switch get(eventData.NewValue,'Tag') % Get Tag of selected object.
            case 'plate'

                plateGUI(fig, bank_names, bank)

            case 'sphere'

                sphereGUI(fig, bank_names, bank)

            case 'cylinder'

                cylinderGUI(fig, bank_names, bank)

            otherwise

                plateGUI(fig, bank_names, bank)

        end
    end
end
```

Main File (Tool_Demo_0404_2023.m)

Class_Definitions Sub Folder Files

Capacitor.m – Basic parallel plate capacitor (class definition)

Dielectric.m – Dielectric material (class definition)

Material_bank.m – creates Dielectric objects with their respective properties (function definition)

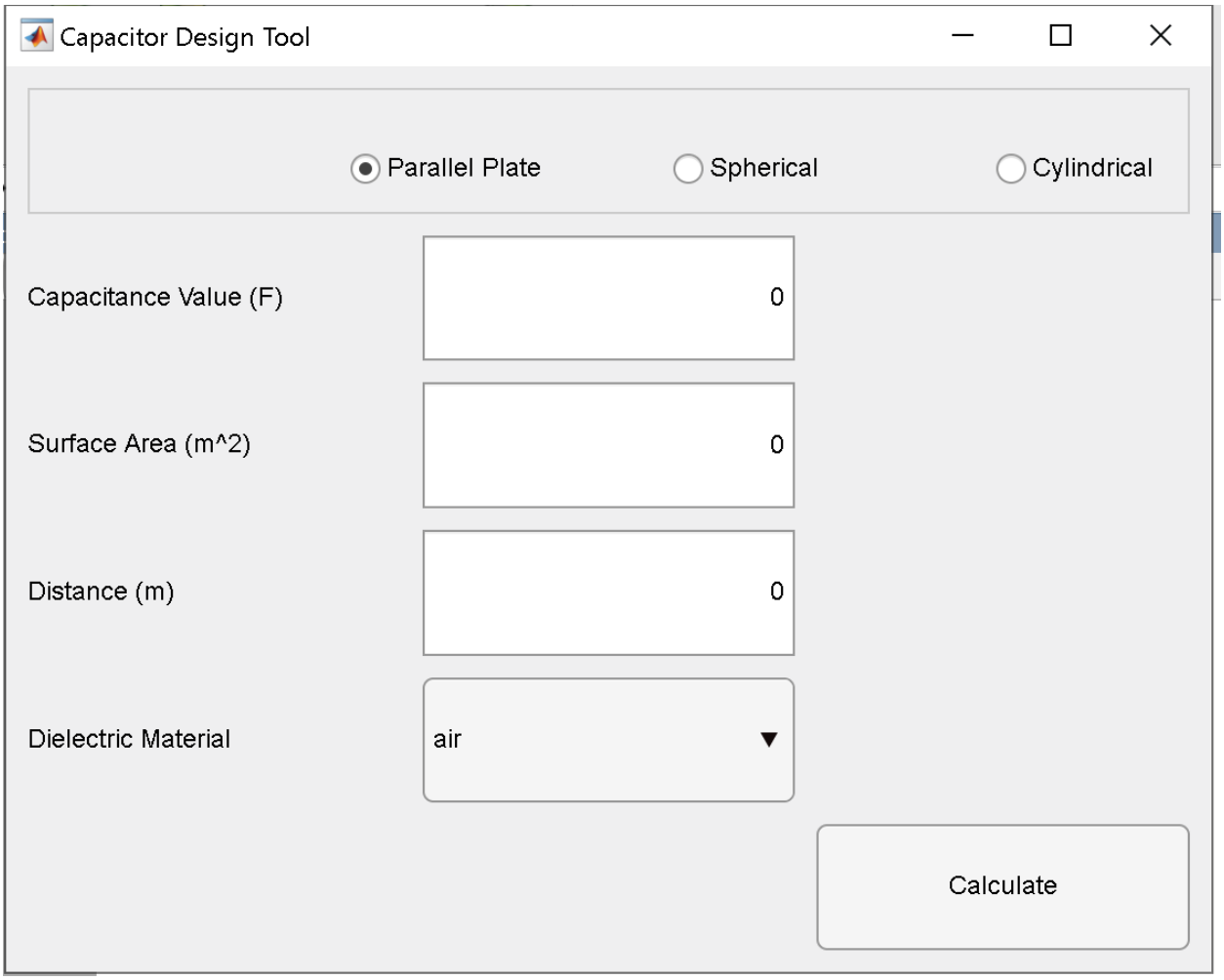
plateGUI.m – Creates GUI for parallel plate capacitor (function definition)

sphereGUI.m – Creates GUI for spherical capacitor (function definition)

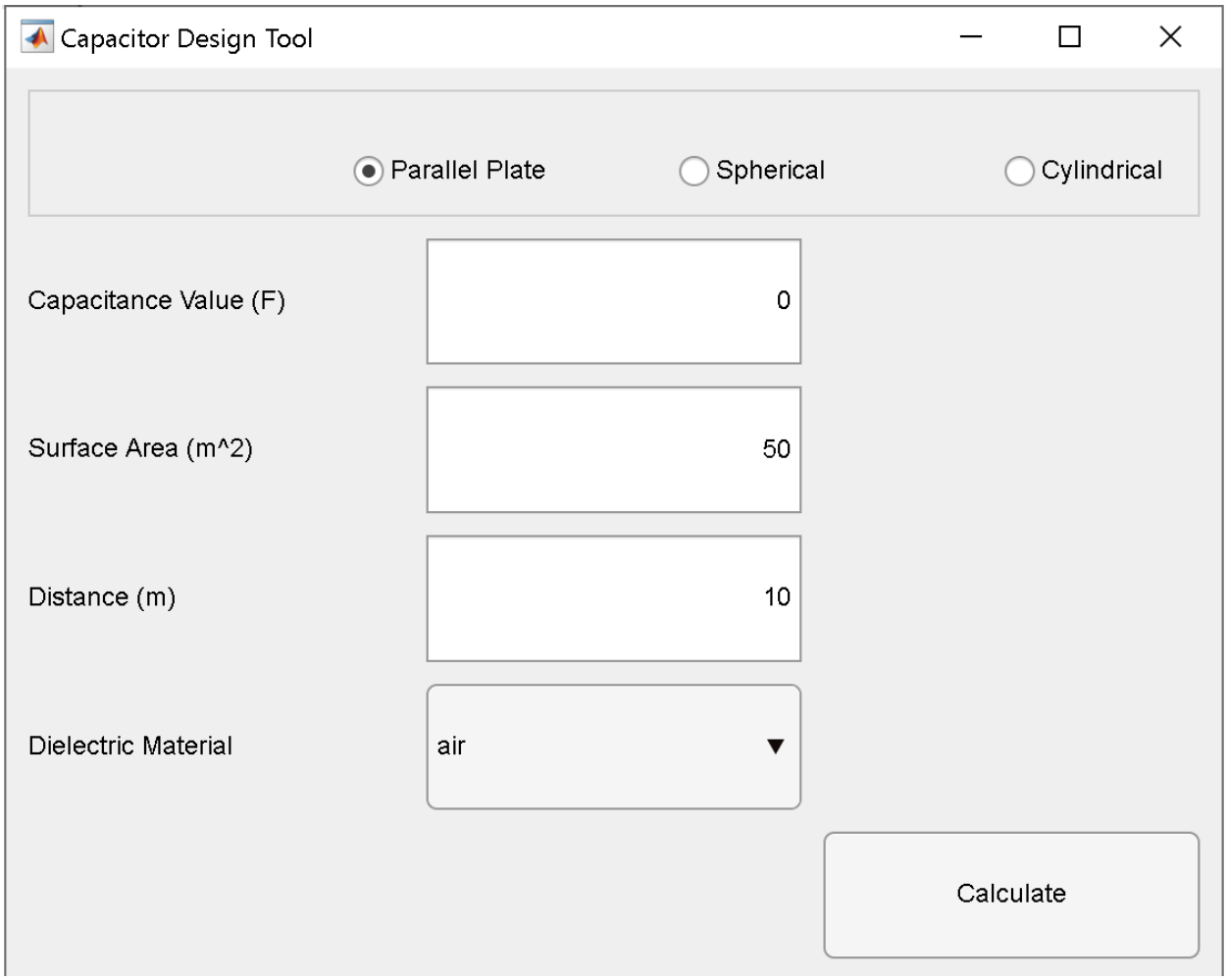
spherical_cap.m – Subclass of Capacitor class, creates spherical capacitor objects with inherited properties from Capacitor class

Main File (or Demo, in the prototype)

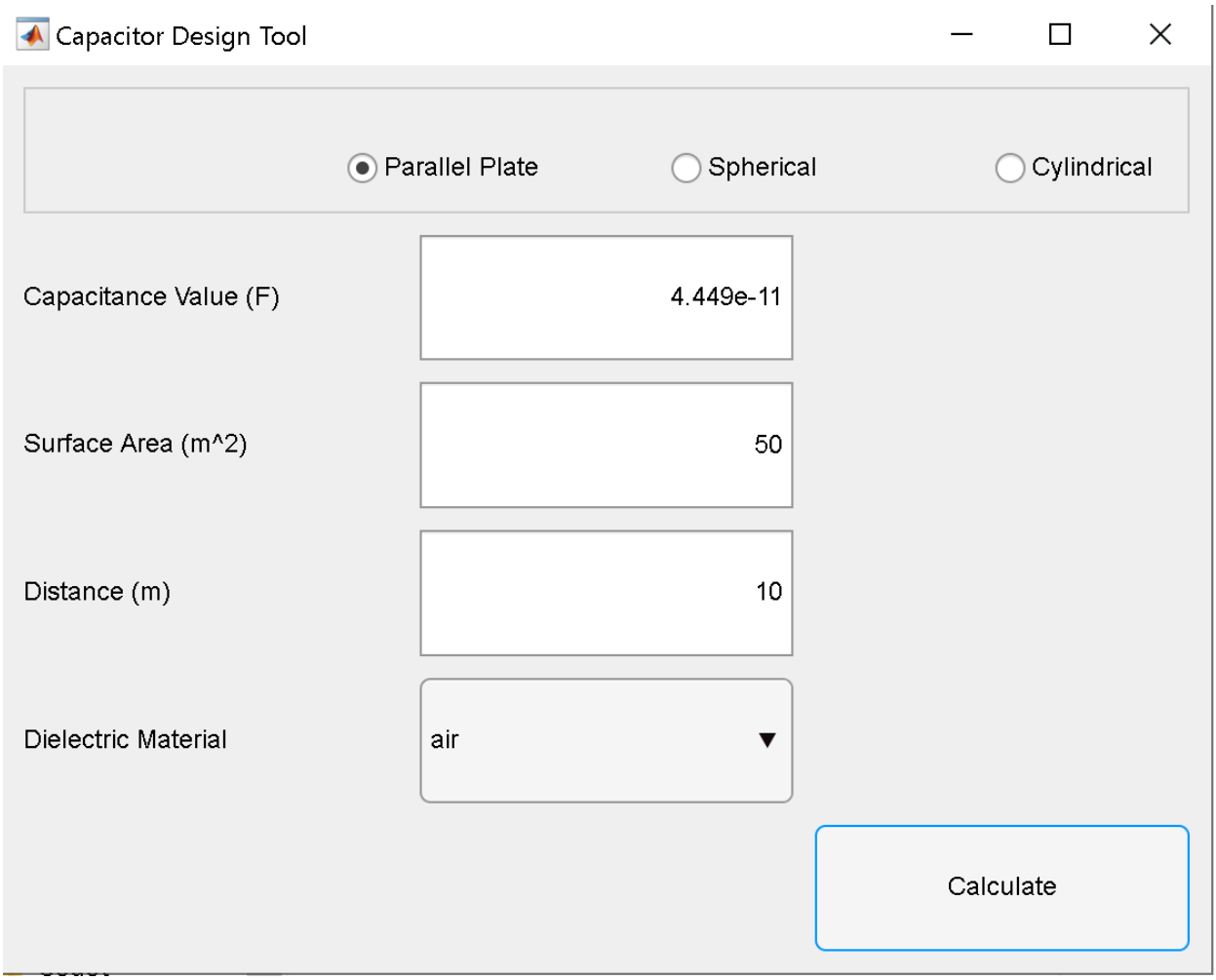
This file sets up the GUI window, and encodes its functionality. After the user has entered all of the required values into the window, the button is pressed, and the function used to generate a capacitor instance is called.



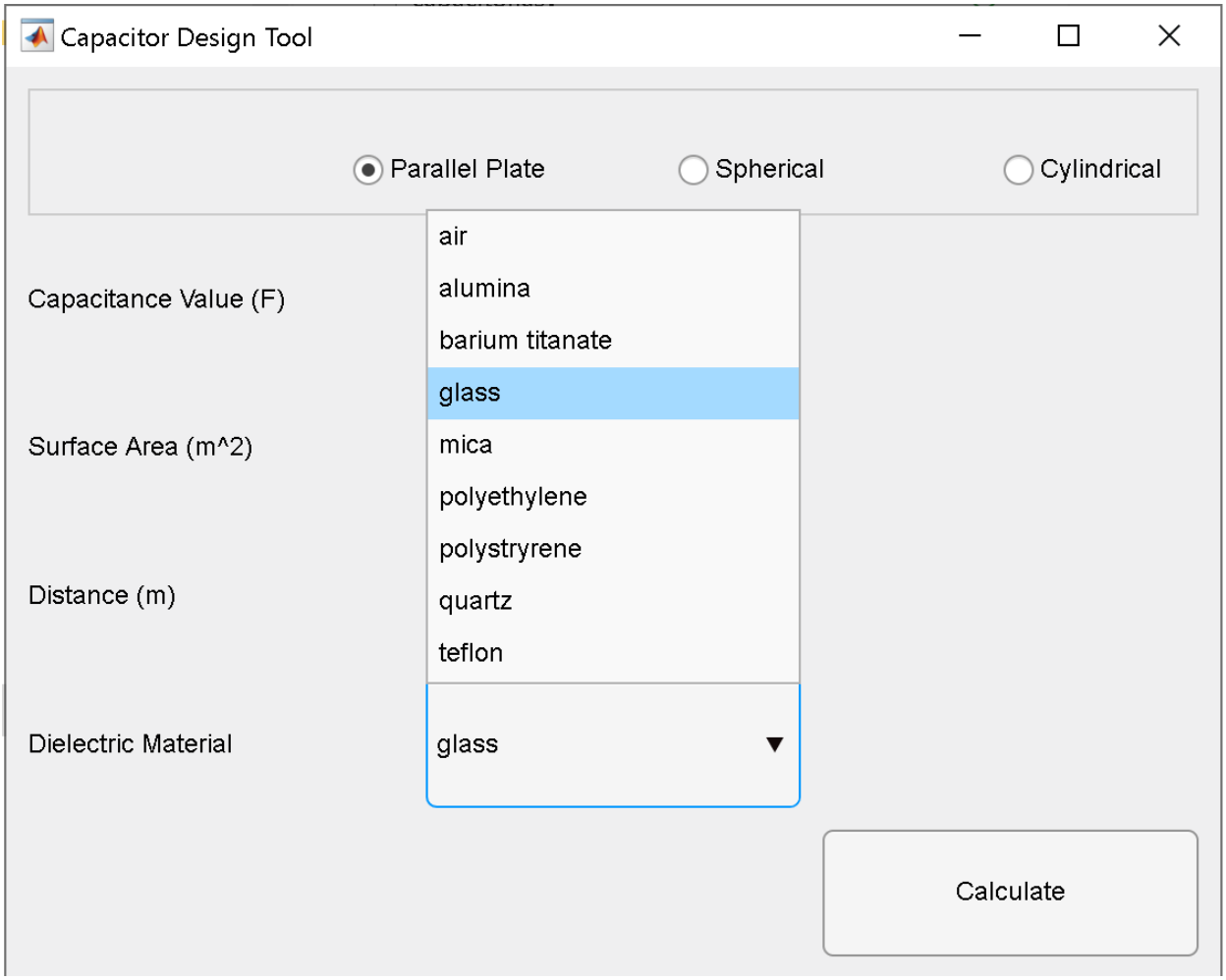
Default GUI for Parallel Plate



User Fills Three of Four Capacitor Requirements



User Presses Calculate and Empty Field Is Populated



Dielectric Menu of Selectable Materials

Future Implementations

Fixing capacitor shape GUIs

Adding circuit modeling for basic voltage divider and maybe user entered transfer function

Create a bode plot of the gain of the voltage divider and transfer functions