

MATLAB EXERCISE 2.13 Capacitance calculator and GUI for multiple structures.

Create a capacitance calculator in the form of a graphical user interface (GUI) in MATLAB to calculate and show the capacitance or per-unit-length capacitance of a coaxial cable [Fig.2.9(b) (from the book)], microstrip transmission line [Fig.2.9(e)], parallel-plate capacitor [Fig.2.9(d)], spherical capacitor [Fig.2.9(a)], and strip transmission line [Fig.2.9(f)], respectively, with the names of structures appearing in a pop-up menu. *[folder ME2_13(GUI) on IR]*

SOLUTION:

Figure S2.8 shows the GUI if, for instance, a strip line is selected in the pop-up menu.

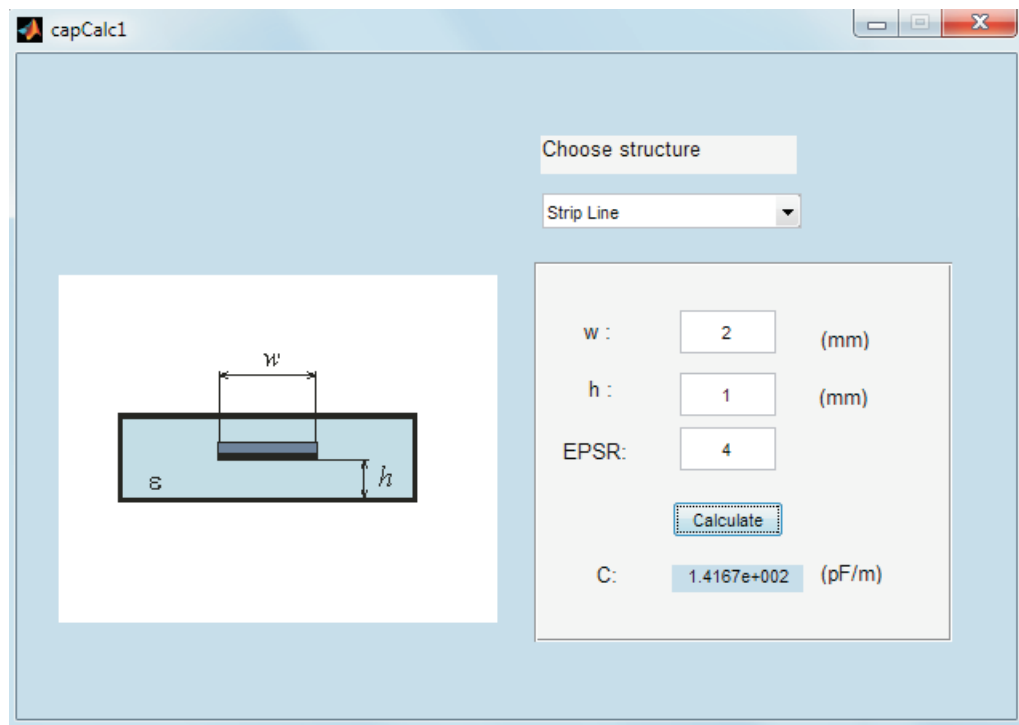


Figure S2.8 MATLAB capacitance calculator and graphical user interface for multiple structures: GUI in the case a strip line is selected in the pop-up menu; for MATLAB Exercise 2.13.

```

%
% Book: MATLAB-Based Electromagnetics (Pearson Prentice Hall)
% Author: Branislav M. Notaros
% Instructor Resources
% (c) 2011
%
% This MATLAB code or any part of it may be used only for
% educational purposes associated with the book
%
%
%
% Capacitance calculator and GUI for multiple structures

function varargout = capCalc1(varargin)
% CAPCALC1 M-file for capCalc1.fig
% CAPCALC1, by itself, creates a new CAPCALC1 or raises the existing
% singleton*.
%
% H = CAPCALC1 returns the handle to a new CAPCALC1 or the handle to
% the existing singleton*.
%
% CAPCALC1('CALLBACK',hObject,eventData,handles,...) calls the local
% function named CALLBACK in CAPCALC1.M with the given input arguments.
%
% CAPCALC1('Property','Value',...) creates a new CAPCALC1 or raises the
% existing singleton*. Starting from the left, property value pairs are
% applied to the GUI before capCalc1_OpeningFcn gets called. An
% unrecognized property name or invalid value makes property application
% stop. All inputs are passed to capCalc1_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help capCalc1

% Last Modified by GUIDE v2.5 01-Jun-2010 21:53:10

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name', mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @capCalc1_OpeningFcn, ...
    'gui_OutputFcn', @capCalc1_OutputFcn, ...
    'gui_LayoutFcn', [] , ...
    'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});

```

```

end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before capCalc1 is made visible.
function capCalc1_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin   command line arguments to capCalc1 (see VARARGIN)

% Import the figure of the current structure

handles.coaxcable = imread('coaxcable.png');
handles.microstrip = imread('microstrip.png');
handles.ppcap = imread('ppcap.png');
handles.sphcap = imread('sphcap.png');
handles.stripline = imread('stripline.png');

% Set the current data value.

set(handles.uipanel1, 'Visible', 'off');
set(handles.axes2, 'Visible', 'off');

% Choose default command line output for capCalc1
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

set(0, 'units', 'inches');
screenSize = get(0, 'ScreenSize');
set(hObject, 'Units', 'inches', 'Position', [screenSize(3)/2-(6.5/2), screenSize(4)/2-(4.3229
/2), 6.5, 4.3229]);

% UIWAIT makes capCalc1 wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = capCalc1_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

```

```

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)
% hObject      handle to popupmenu1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
% Determine the selected data set.

global i;
% Set current data to the selected data set.
switch get(handles.popupmenu1, 'Value')
case 1
    set(handles.uipanel1, 'Visible', 'off');
    cla reset; axis off;
case 2
    handles.current_data = handles.coaxcable;
    imshow(handles.current_data);
    set(handles.text1, 'String', 'a :');
    set(handles.text4, 'String', 'b :');
    set(handles.text6, 'String', '(mm)');
    set(handles.text11, 'String', '(pF/m)');
    set(handles.uipanel1, 'Visible', 'on');
    i = 1;
case 3
    handles.current_data = handles.microstrip;
    imshow(handles.current_data);
    set(handles.text1, 'String', 'w :');
    set(handles.text4, 'String', 'h :');
    set(handles.text6, 'String', '(mm)');
    set(handles.text11, 'String', '(pF/m)');
    set(handles.uipanel1, 'Visible', 'on');
    i = 2;
case 4
    handles.current_data = handles.ppcap;
    imshow(handles.current_data);
    set(handles.text1, 'String', 'S :');
    set(handles.text4, 'String', 'd :');
    set(handles.text6, 'String', '(mm^2) :');
    set(handles.text11, 'String', '(pF)');
    set(handles.uipanel1, 'Visible', 'on');
    i = 3;
case 5
    handles.current_data = handles.sphcap;
    imshow(handles.current_data);
    set(handles.text1, 'String', 'a :');
    set(handles.text4, 'String', 'b :');
    set(handles.text6, 'String', '(mm)');
    set(handles.text11, 'String', '(pF)');

```

```

    set(handles.uipanel1, 'Visible', 'on');
    i = 4;
case 6
    handles.current_data = handles.stripline;
    imshow(handles.current_data);
    set(handles.text1, 'String', 'w :');
    set(handles.text4, 'String', 'h :');
    set(handles.text6, 'String', '(mm)');
    set(handles.text11, 'String', '(pF/m)');
    set(handles.uipanel1, 'Visible', 'on');
    i = 5;

end
global ready;
global var;
ready = [0 0 0];
var = [0 0 0];
set(handles.pushbutton1, 'Enable', 'off');
set(handles.edit1, 'String', '');
set(handles.edit2, 'String', '');
set(handles.edit3, 'String', '');
set(handles.text10, 'String', '');

%Save the handles structure.
guidata(hObject, handles)

% Hints: contents = get(hObject, 'String') returns popupmenu1 contents as cell array
%         contents{get(hObject, 'Value')} returns selected item from popupmenu1

% --- Executes during object creation, after setting all properties.
function popupmenu1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'), get(
(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

global ready;
ready = [0 0 0];
global var;
var = [0 0 0];

function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
% str2double(get(hObject,'String')) returns contents of edit1 as a double

% --- Executes during object creation, after setting all properties.

handles.edit1 = str2double(get(hObject,'String'));
global ready;
global var;
if (isnan(handles.edit1));
    msgbox('Invalid input','Error');
    ready(1)= 0;
else
    ready(1)= 1;
    var(1) = handles.edit1;
end;
if (ready == [1 1 1])
    set(handles.pushbutton1,'Enable','on');
else
    set(handles.pushbutton1,'Enable','off');
end;

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)
% hObject handle to edit2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
% str2double(get(hObject,'String')) returns contents of edit2 as a double

% --- Executes during object creation, after setting all properties.

handles.edit2 = str2double(get(hObject,'String'));
global ready;
global var;

```

```

if (isnan(handles.edit2));
    msgbox('Invalid input','Error');
    ready(2)= 0;
else
    ready(2)= 1;
    var(2) = handles.edit2;
end;
if (ready == [1 1 1])
    set(handles.pushbutton1,'Enable','on');
else
    set(handles.pushbutton1,'Enable','off');
end;

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%         str2double(get(hObject,'String')) returns contents of edit3 as a double

% --- Executes during object creation, after setting all properties.

handles.edit3 = str2double(get(hObject,'String'));
global ready;
global var;
if (isnan(handles.edit3));
    msgbox('Invalid input','Error');
    ready(3)= 0;
else
    ready(3)= 1;
    var(3) = handles.edit3;
end;
if (ready == [1 1 1])
    set(handles.pushbutton1,'Enable','on');
else
    set(handles.pushbutton1,'Enable','off');
end;

```

```
end;
```

```
% --- Executes during object creation, after setting all properties.
```

```
function edit3_CreateFcn(hObject, eventdata, handles)
```

```
% hObject    handle to edit3 (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
```

```
% See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'), get(
```

```
(0,'defaultUicontrolBackgroundColor'))
```

```
    set(hObject,'BackgroundColor','white');
```

```
end
```

```
% --- Executes on button press in pushbutton1.
```

```
function pushbutton1_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to pushbutton1 (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)
```

```
% --- Executes during object creation, after setting all properties.
```

```
global var;
```

```
global i;
```

```
EPS0 = 8.8542*10^(-12);
```

```
mm2m = 10^(-3);
```

```
mmsq2msq = 10^(-6);
```

```
if i == 1;
```

```
    a = var(1)*mm2m;
```

```
    b = var(2)*mm2m;
```

```
    EPSR = var(3);
```

```
    EPS = EPS0*EPSR;
```

```
    C = capacitanceCoaxCable(EPS,a,b);
```

```
else if i == 2;
```

```
    w = var(1)*mm2m;
```

```
    h = var(2)*mm2m;
```

```
    EPSR = var(3);
```

```
    EPS = EPS0*EPSR;
```

```
    C = capacitanceMicrostrip(EPS,w,h);
```

```
else if i == 3;
```

```
    S = var(1)*mmsq2msq;
```

```
    d = var(2)*mm2m;
```

```
    EPSR = var(3);
```

```
    EPS = EPS0*EPSR;
```

```
    C = capacitancePPCapacitor(EPS,S,d);
```

```
else if i == 4;
```

```
    a = var(1)*mm2m;
```

```
    b = var(2)*mm2m;
```

```
    EPSR = var(3);
```

```
    EPS = EPS0*EPSR;
```



```
        C = capacitanceSphCapacitor(EPS,a,b);
    else
        w = var(1)*mm2m;
        h = var(2)*mm2m;
        EPSR = var(3);
        EPS = EPS0*EPSR;
        C = capacitanceStripline(EPS,w,h);
    end;
end;
end;
C = C*10^12;
set(handles.text10, 'String', num2str(C, '%.4e'));

% --- Executes during object deletion, before destroying properties.
function text6_DeleteFcn(hObject, eventdata, handles)
% hObject    handle to text6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
%  
% Book: MATLAB-Based Electromagnetics (Pearson Prentice Hall)  
% Author: Branislav M. Notaros  
% Instructor Resources  
% (c) 2011  
%  
% This MATLAB code or any part of it may be used only for  
% educational purposes associated with the book  
%  
%  
%  
% Capacitance per unit length of coaxial cable  
  
function C = capacitanceCoaxCable(EPS,a,b)  
  
C = 2*pi*EPS/(log(b/a));  
  
return
```

```
%  
% Book: MATLAB-Based Electromagnetics (Pearson Prentice Hall)  
% Author: Branislav M. Notaros  
% Instructor Resources  
% (c) 2011  
%  
% This MATLAB code or any part of it may be used only for  
% educational purposes associated with the book  
%  
%  
%  
% Capacitance per unit length of microstrip  
  
function C = capacitanceMicrostrip(EPS,w,h)  
  
C = EPS*w/h;  
  
return
```

```
%  
% Book: MATLAB-Based Electromagnetics (Pearson Prentice Hall)  
% Author: Branislav M. Notaros  
% Instructor Resources  
% (c) 2011  
%  
% This MATLAB code or any part of it may be used only for  
% educational purposes associated with the book  
%  
%  
%  
% Capacitance of parallel plate capacitor  
  
function C = capacitancePPCapacitor(EPS,S,d)  
  
C = EPS*S/d;  
  
return
```

```
%  
% Book: MATLAB-Based Electromagnetics (Pearson Prentice Hall)  
% Author: Branislav M. Notaros  
% Instructor Resources  
% (c) 2011  
%  
% This MATLAB code or any part of it may be used only for  
% educational purposes associated with the book  
%  
%  
%
```

```
% Capacitance of spherical capacitor
```

```
function C = capacitanceSphCapacitor(EPS,a,b)
```

```
C = 4*pi*EPS*a*b/(b-a);
```

```
return
```

```
%  
% Book: MATLAB-Based Electromagnetics (Pearson Prentice Hall)  
% Author: Branislav M. Notaros  
% Instructor Resources  
% (c) 2011  
%  
% This MATLAB code or any part of it may be used only for  
% educational purposes associated with the book  
%  
%  
%  
  
% Capacitance of stripline  
  
function C = capacitanceStripline(EPS,w,h)  
  
C = 2*EPS*w/h;  
  
return
```