**Creating a GUI In Matlab 5**

**For Summer 1997** [**Envision-It!**](http://physics.gac.edu/%7Ehuber/envision/) **Workshop**

One of the major comments that we hear about Matlab is "I don't know how I can use it in my classroom.  It is too hard for my students to use."  This may be a correct comment - we don't necessarily want the students to construct a model from scratch in Matlab, but would want them to use a previously prepared model.  We can do this using Input and disp statements in a M-file which is run from the Matlab window.  A much better way is to use a Graphical User Interface (GUI) where the student just "fills in the blanks" and all of the Matlab commands are completely hidden from the student.

Creating GUI's in Matlab Version 5 is significantly easier than in previous versions of the program.  In particular, the guide program allows one to easily place controls onto a model and set their properties.  In the following exercise, we will   
construct a GUI which will calculate the Black-Body temperature of a planet near the sun. As outlined in the   
 [Energy Balance Model handout](http://physics.gac.edu/%7Ehuber/envision/instruct/ebm1doc.htm) the black-body temperature is defined as

(Eqn. 1): T = [S\*SX(1-a)/4 / s] 1/4-T0.

Where the symbols are defined as:

|  |  |
| --- | --- |
| T | The Temperature of the Earth in Centigrade |
| S | Solar Constant (1370 W/m2) at surface of earth |
| SX | Solar Multiplier (Fraction of Earth's Solar Flux) |
| a | Albedo - Fraction of incident solar radiation reflected (about 0.32) |
| s | Stefan's Constant (5.6696E-8 W/m2K4) |
| T0 | Conversion from Kelvin to Centigrade (273.15) |

Now, we are going to develop a Matlab GUI which allows a person to find the black-body temperature using this equation.

1. Type "guide" in the Matlab Control Window to start the GUI editor.  You should see a Figure Window which has dotted lines in it, and the "Guide Control Panel" window (which I will abbreviate GCP window.
2. Click on the "Text" box at the bottom of the GCP window, then click at some point on the Figure window, and it will create a box - this is the box that we will want to contain the text 'Solar Multiplier'
3. Double click on the box we just created and the "Guide Property Editor" window will pop up.  This allows us to set the properites of the text box.  In the white region of the window erase the word BackgroundColor and type in String and press enter.  In the next region, it will display two single quotes ('').  Type in the words Solar Multiplier between the quotes and press enter
4. The string will now be displayed in the text box on the figure.  You may need to resize it my dragging on one of the corners of the text box.
5. Now we will want to make the box where the user will type a value for the solar multiplier.  To do this, click on the "Edit" box in the GCP to create an edit-text box (where the user can type in the value for the solar multiplier).  Place it near the text box that we just created.  This should be white with a black border
6. Double click the edit-text box.  We will want to set is String to '1'  We also need to set a "Tag" for this box (so we can read the string that is in the box in our calculation program).  I used the tag 'SX' (all caps) for this box
7. Repeat the operations above to create a text box with the string 'Albedo' and a editable-text box which contains the string '0.32' and has a tag of 'Albedo'
8. In the GCP, select Pushbutton, and place it at some location on the screen.
9. Double-click on the pushbutton and set its string to 'Calculate'
10. When we click this pushbutton, we want Matlab to run a script called ebm0calc.  To do this we set the CallBack property for this button to 'ebm0calc'  The callback property is a Matlab command (or series of commands separated by semicolons) which are executed when the button is pressed.
11. Finally, we need a place to display the final temperature.  Make two text boxes and set their BackgroundColor to Cyan.  In one box, put the string 'Blackbody Temperature' and for the other box, give it a Tag property of TFinal.  This is where we will write the final temperature.
12. At this point, our GUI is complete and should look somewhat like the figure below.
13. Under the Options menu, choose, Activate Figure and save it as ebm0gui.m  By activating the figure, we leave the GUI editor and enable Matlab to process the buttons we have defined.  The file ebm0gui.m contains the Matlab commands to generate this screen.

Now we are ready to create the file  [ebm0calc.m](file:///A:\Gustavus%20Adolphus\ebm0calc.m) which will perform the calculations and display the results on the screen.  The file below will do this:

% <pre>   
% EBM0CALC.M   
% Energy Balance Model for Eqn 1 in   
%     http://physics.gac.edu/~huber/envision/instruct/ebm1doc.htm   
% Based on Equation 3.8   
%     from 'A Climate Modeling Primer', 1st Edition, 1987   
%     A. Henderson-Sellers and K. McGuffie   
%   
% Written by Tom Huber, 14-JUL-97  for Envision-It Workshop 1997   
%   
% Requires EBM0GUI  - GUI Front end to set up values

S = 1370;                 % Solar Constant (W/m^2)   
Sigma = 5.6696E-8;        % Steffan's Constant (in W/m^2K^4)   
T0 = 273.15;              % Kelvin -> Centigrade Conversion   
    
% Get the Albedo Value (from the object with a tag "Albedo"   
% the property we want is the "String"   
%   Must do a str2num conversion to convert from the string   
%   to a number   
HH = findobj(gcf,'Tag','Albedo');   
albedo = str2num(get(HH,'String'));

HH = findobj(gcf,'Tag','SX');   
SX = str2num(get(HH,'String'));   
    
    
TFinal = (S\*SX\*(1-albedo)/4/Sigma)^.25;

Units = ' C';  % Degrees Centigrade   
TFinal = TFinal - T0; %Convert from Kelvin to Centigrade   
    
% Get the Handle for the TFinal Tag   
HH = findobj(gcf,'Tag','TFinal');

% Set the string for the object TFinal   
set(HH,'String',[num2str(TFinal) Units])

There are a couple of things that we should note:

1. We obtain the values for albedo and SX by first finding their handles using the command
2. HH = findobj('Tag','Albedo');
3. Then we read the string using the get(HH,'String') and use str2num to get the values.
4. Similarly, to set the display for TFinal, we first find its handle using HH = findobj(gcf,'Tag','TFinal');
5. Then we set the value using set(HH,'String',[num2str(TFinal) Units])

Electronic Copy: http://physics.gac.edu/~huber/envision/matgui/matgui.htm   
Created: 7-JUL-97 by [Tom Huber](http://physics.gac.edu/%7Ehuber/), [Physics Department](http://physics.gac.edu/), [Gustavus Adolphus College](http://www.gac.edu/).