UNIVERSITY OF CALIFORNIA, MERCED

BERKELEY · DAVIS · IRVINE · LOS ANGELES · MERCED · RIVERSIDE · SAN DIEGO · SAN FRANCISCO · SANTA BARBARA · SANTA CRUZ

ME 021: Engineering Computing
MATLAB® Project

Fall 2023 ME 021 Course Staff: Alex Frias (ASE), Amir Kouravand (GSI), Hansong Lee (GSI)
Hassan Jafari Mosleh (GSI), Ayush Pandey (IOR)

Preliminary Notes

- **You need to choose one project out of four problem statements** Choose a project that is closest to your interests in "engineering computing" out of the four engineering projects posted as PDF on CatCourses (also available in course Box/MATLAB® Project folder). You must denote your choice in the 2nd MATLAB® project milestone, that is, the 7th project milestone on CatCourses.
- Your solution must be exclusively submitted via CatCourses. Email submission will not be accepted. Pay attention to the posted deadline because **the system automatically stops accepting submissions when the deadline passes.** You can upload one or more `.m` or `.mlx` files for your code.
- **To receive credit, you must also demo your project to your TA, either in your lab or during office hours.** You must show your working program, be prepared to explain your code, and answer other questions that the TA asks about MATLAB® programming. You must demo before the end of your last lab of the semester (that is, before Dec. 8, 2023). **You will get a 0 if you do not demo, even if you made a submission.**
- Four milestones will be released to help you make progress towards the project. However, milestones are simply to push you in the right direction. You need to put in extra effort beyond the milestones to complete the project satisfactorily.
- A short (1-2 page) report in PDF format that consists of main parts of the code, the visualizations, and any mathematical equations must be submitted for the final project submission along with the MATLAB® script files.
- The grading rubric for all milestones, demo, and the report submission will be posted on CatCourses and it will be available for the students to read through to prepare their submissions.

Contents

ME 021: Engineering Computing
MATLAB® Project – GUI for interactive engineering analysis

Amir Kouravand, Ayush Pandey

1 Overview

You are reading this PDF in a graphical user-interface or a GUI (pronounced "gooey" or "jee-you-eye"). The GUI that you are using is likely the Box PDF interface (or Adobe Acrobat). Similarly, CatCourses is a nifty GUI. It provides a graphical, user-friendly, and easy-to-use interactive framework to manage everything to facilitate academic coursework. Without such a graphical interface, managing the course content would be much harder! Similarly, in many engineering applications, a GUI is necessary to create easy-to-use engineering design and analysis pipelines. Examples include SolidWorks for mechanical prototyping, automotive diagnostic systems to monitor vehicle properties, and many others. In this project, you will develop your own graphical user interface for an engineering application.

1.1 System description

For this project, your main objective is to design a GUI for an engineering device called a "solar hot water flat plate collector". This GUI must have features that an engineer who is designing a new system may find useful in system analysis and computation.

A solar hot water flat plate collector is used to convert solar energy to heat water. It is a sustainable alternative to traditional water heating methods with gas or electricity.

The collector uses an "absorber plate" made of a highly conductive material. The plate is coated with a selective surface that maximizes its ability to absorb solar radiations. Water or heat transfer fluid is circulated through tubes or channels within the collector. As the absorber plate absorbs solar energy, it heats up the transfer fluid. For the design of this system, various parameters can be tweaked such as tilt angle, fluid choice, panel material etc. A GUI can streamline the analysis to study the effects of these choices.

1.2 About GUIs

GUI makes computing fun! Without graphics and user interaction, all we would see is binary numbers and programming would be quite challenging (also, boring). Graphical interfaces can be of many different kinds. For example, MATLAB® is a GUI for programming. Even a simple mouse click interaction in an app would count as a GUI feature, while a complex GUI may take inputs from the mouse, the keyboard, touch screen, and provide complex output/display options. You can read more about GUIs[1] to learn about them.

[1]About GUIs: https://en.wikipedia.org/wiki/Graphical_user_interface

2 Computational tasks

The main computational task in this project is to generate a user-friendly interface that an engineering designer can use to optimize the design of a solar hot water plate collector.

2.1 The interface

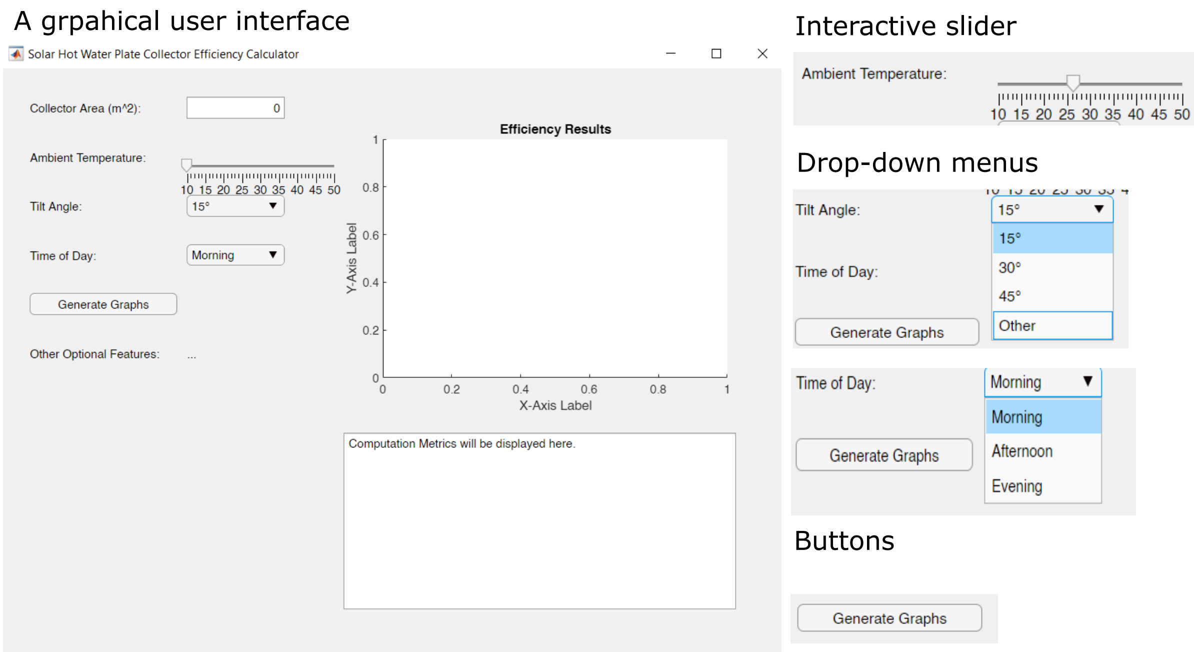

Figure 1: An illustration of a graphical user interface created in MATLAB®. Drop-down menus, sliders, and buttons are illustrated in the right panel.

You must build your GUI with the following features (see Figure 1 for reference):

1. An input text box for the collector area. The area in this box is entered in m^2 units. Your GUI must validate that a number is entered.

2. A slider for changing the ambient temperature. The ambient temperature should be varied by the slider from 10°C to 50°C.

3. A drop-down menu to input the tilt angle of the collector. The drop-down menu should have some fixed values of the tilt angle. Last item on the drop-down menu should be "Other". When other is selected, the user must be able to enter a value in an input box.

4. A drop-down menu to choose the time during the day in which the collector is operating (morning, afternoon, and evening).

5. Optional text input boxes for the azimuth of the collector, the inlet temperature, and the radiation on the solar panel.

6. Buttons for graph generation options.

An example GUI is shown in Figure 1. You may personalize your GUI in other ways but every GUI must have a slider, a drop-down menu, buttons, and a space where output(s) are displayed.

2.2 The computation of metrics using the GUI

Add an "Import data" or "Import CSV" button in your GUI. Clicking on this button, the user should be able to specify the file name that contains the data for the flat plate collector. Then, you can use the data provided along with the parameters below to find out the efficiency. The data is available on the Box folder: Temp_Radiation.csv. The file gives the values of the radiation intensity with time and varying ambient temperature.

Compute the efficiency η of a solar water plate collector using the GUI that you have designed. Given the parameters and data, use the following equations to determine the useful heat gain Q_u and the efficiency η of the collector.

The user provides some of these parameters through the GUI (Task 2.1) and the other parameters can be hard-coded into the interface.

- The useful heat gain, Q_u (in watts), is given by:

$$Q_u = A_c \left(I(\tau\alpha) - U_L(T_i - T_a) \right)$$

  where:

  - A_c is the area of the collector, which may range from 2 to 6 m^2.
  - I is the solar radiation intensity (in watts per square meter, W/m^2),
  - τ is the transmittance of the collector, which can be in the range of 80% to 95%.
  - α is the absorptance of the collector, which can be in the range of 85% to 95%.
  - U_L is the overall heat loss coefficient, which may range from 2 to 8 W/m^{2}K.
  - U_L is the overall heat loss coefficient,
  - T_i is the inlet temperature of the collector fluid, which can be considered 300 K,
  - T_a is the ambient temperature (in degrees Celsius, °C or Kelvin, K).

- The efficiency η (as a percentage) is calculated as follows:

$$\eta = \frac{Q_u}{A_c I} \times 100$$

Utilize the provided data to calculate these metrics, and document your calculations and results. Discuss any factors that might affect the efficiency of the solar water plate collector.

2.3 Visualization

Choose any two of the following graphs and provide options to generate them on your GUI.

1. Efficiency with $(T - T_a)$ for different values of I
2. Efficiency with I.
3. Efficiency with $(T - T_a)/I$ for different values of the tilt angle.
4. Efficiency with time.

3 Open-ended tasks

1. You can generate one graph at a time with your GUI, but how about allowing the user to choose multiple graphs that are plotted on the same axes? How does that feature look like in a GUI?
2. What would be your suggested settings for the knobs in your GUI to design the most efficient water plate collector.

For the open-ended problems, include justifications for each of your choices, that is, explain your reasoning and thought process. Discuss the potential trade-offs (advantages and disadvantages) and how you address them. Take the open-ended questions as an opportunity to explore data/system design/visualization that interests you the most!