

Work in Progress - Computer Problem Solving using Matlab

Michael Parke

The Ohio State University, parke.4@osu.edu

Abstract – The transition from small homework sets, which is how programming languages are traditionally introduced, to being able to use computers to solve complex problems is often a difficult challenge for students. It is a transition that is important to introduce before students become involved in more complex programming tasks in their majors and therefore fits appropriately in the second half of a first year program. This article describes an approach of using moderate complexity projects to both introduce how one approaches bigger program challenges and provide experiences for students to draw upon later.

Index Terms - Programming, Matlab

INTRODUCTION

The majority of departments in the college of Engineering at The Ohio State University have an option for students to fulfill their programming requirement by taking a course in computer problem solving with Matlab (ENGR 1221 - a two credit hour course) through the new Department of Engineering Education. The purpose of this paper is to describe the content and philosophy behind this course. Students on the regular and scholars tracks, as part of their First Year experience (FYE), are introduced to Matlab during their first semester. This course builds on that initial experience and is often taken during the second semester of the first year. Presently, over 400 students take this course each academic year. Because the disciplines served by this course include Engineering Physics, Industrial Systems Engineering, Food Agriculture and Biological, Aeronautical and Astronautical, Chemical, Civil, Environmental, and Welding, there are a wide variety of future programming needs. The only common background in programming is from what is introduced in the first semester of the FYE. To meet the differing needs of students in the departments, the content of ENGR 1221 is broken into three sections: review and extension, working with moderately large data sets, and the process of developing modestly large programs, such as used in modeling and data analysis.

Each class has 36 students and is taught by an instructional team composed of an instructor and two undergraduate teaching assistants. New teaching assistants are paired with a more experienced teaching assistant and all have previously taken the course. The format of each class is to begin with a short presentation on the concepts being

introduced, followed by in class work time. Since there are often a number of valid ways to solve a given problem, it is important to have classroom support that is able to follow a student's logic and either help them to successfully get it to work or explain why the approach needs to be modified. Tasks evolve from simpler tasks to more complex and sophisticated. The review section uses homework sets. Projects are used to introduce working with data sets and developing larger programs. Projects can last from one to five class periods and are introduced once the concepts that are needed have been introduced. Thus the course begins with a mix of review and projects, switching to only projects once the review topics are done. The nature of the projects can change over time, but the philosophy for presenting how to program does not. This paper describes the version of the course that was taught in 2015-2016 and the anticipated changes that will be implemented for the next academic year.

DATA PROJECTS

The data sets presently used are field data from an experiment on the edge of the Red Sea and ice core data from Greenland. The data projects are intended to reinforce the use of data manipulation techniques on a moderately big data set rather than the small numeric arrays typically found in programming homework sets.

The Red Sea data used in this class consist of a few days of data from a longer term project that has been collected to support the study of water vapor advection and heat over a coral-reef lagoon. Figure 1 gives an example where the students are asked to extract one day of data and plot the

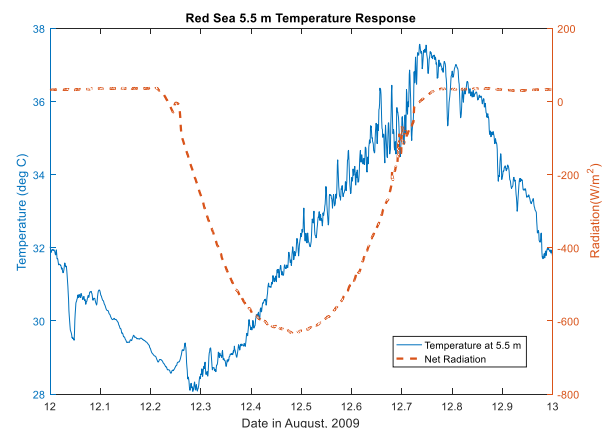


FIGURE 1
EXAMPLE OF DATA GRAPHING

temperature at 5.5 m and the net radiation as a double y-axis plot. The data include measurements of temperature, relative humidity, and water vapor pressure at two heights above the lagoon plus net radiation as measured on the shore next to the lagoon. The data set is used to practice data extraction, statistical summaries, and graphing over two class periods.

The ice core data contains continuous, annually-averaged deposition records (for years dating between 1772 and 2003) of highly toxic thallium, cadmium, and lead. Because of differences in atmospheric circulation from year to year the amounts of pollution deposited are highly variable. However, they do clearly show the onset of the industrial revolution and the later implementation of pollution controls. Because of the variability in the record this is used as a two class project to practice answering questions such as “How many years during the period of this data set have a lead concentration greater than 0.2 ng/g?” and “What is the most recent year during which lead is greater than 100 times the concentration of cadmium?”. Students are required to answer these questions both with arrays and logical expressions and with loops and conditional statements.

REPETITION PROJECT

This is a project that has students generate ever more complex models of stacking apples for a grocery store display. They start with a simple, complete square pyramid and build to modeling a display where they are given a number of apples, the diameter of the apples, base layer dimensions, the maximum number of layers that can be stacked without damaging the fruit, and the height of a display sign over the display area. The students need to combine loops and conditional statements so that development of the pyramid will stop when the program runs out of apples to include, the maximum layer height is reached, or the height of the sign is reached. Figure 2 shows a display and signage from a grocery store.



FIGURE 2
STACKED FRUIT IN A GROCERY STORE

PARACHUTE PROJECT

This is a five class project where students sequentially develop a model of a parachute drop. It begins with introducing time stepping using a model of a ball toss. The results of the time stepping solution can be compared with the known analytic solution. The time stepping code is then converted to a simple parachute drop. Students, step-by-step, add more complexity to the model including a simple model of the chute opening, realistic drag, and an atmospheric density profile. The project culminates in a team project to model the then record breaking parachute drop in 1960 by Joseph Kittinger from 31,300 meters where he reached a speed of 274.5 m/s. Figure 3 shows the results from this model. The modeled maximum speed will not

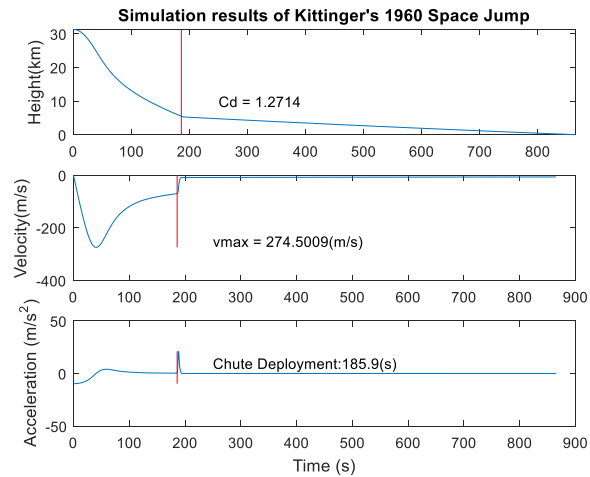


FIGURE 3
MODEL OF THE 1960 JOSEPH KITTINGER PARACHUTE DROP

match for a number of reasons, including the fact that the atmosphere on the day of the jump will not have been exactly the same as the standard model used. Students are introduced to a tunable parameter by assuming all of the error comes from the drag coefficient. They are asked to come up with an algorithm that will adjust the drag until the maximum freefall speed matches what was measured at the time.

GUI PROJECT

The course ends with an introduction to graphical user interfaces (GUIs) over four class periods. They are introduced to Matlab’s GUI development environment and begin by designing and then implementing a simple adding calculator GUI and then a more complex loan calculation GUI. For the loan calculator, the user will fill in three of the four boxes and press calculate. The program will determine which box is empty and do the appropriate calculation to fill it in. Figure 4 shows one possible layout of the loan calculator. Students are then asked to design a GUI based on any previous course assignment such as simulating an ATM or one of the parachute models. The final project is a group

project to simulate a tool that could be used for decision making at any point in a game. In this case the game is liar's dice.

The author would like to thank the Engineering Education Department for their support in developing this course.

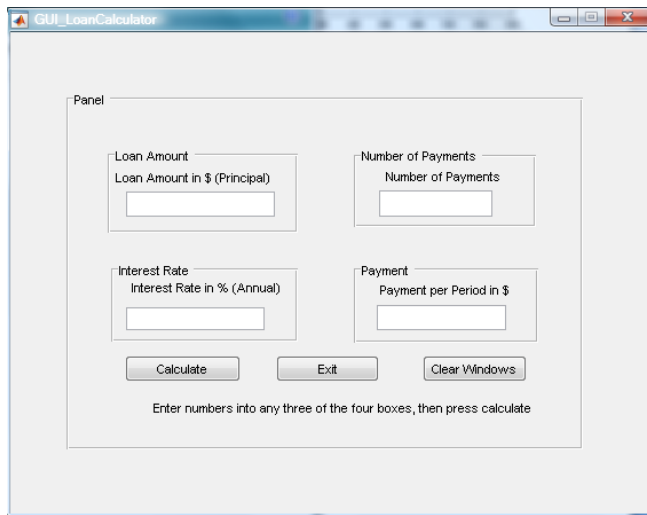


FIGURE 4
GUI LAYOUT FOR A LOAN CALCULATOR

AUTHOR INFORMATION

M Parke Assistant Professor, The Ohio State University,
parke.4@osu.edu

DISCUSSION

This course has been an evolving process, with the only constants being the parachute project and the Red Sea data set. Other final projects have been used at various times, including a tsunami travel time project and a model of a solar powered plane tied to google earth. The amount of review and extension has been increased over time because this provides a better transition to the tasks in the projects.

Feedback on course content is found by determining what tasks different majors require for their courses. This information can come directly from the relevant departments, but is generally from students. Student information comes from the undergraduate teaching assistants who have previously taken ENGR 1221, from students who are taking the relevant majors courses simultaneously with ENGR 1221, and from students who come back to either visit or seek help on their later homework.

As an example, for the next academic year a project will be added to introduce the Simulink programming environment that Matlab provides. Simulink is now being used in five departments and has specifically been asked for by six students. This will likely replace the GUI project. A model of the pilot's experience from the Virgin Galactic spacecraft disintegration will be either added or used to replace the Joseph Kittinger parachute drop.

Feedback from students has been generally positive, although from a self-selected population. An analysis of results from the 2015-2016 academic year based on gender and engineering discipline are given in an accompanying paper.

ACKNOWLEDGMENT