USF UNIVERSITY OF SOUTH FLORIDA

Department of Mechanical Engineering

COLLEGE OF ENGINEERING

EML3041: Computational Methods

Semester and Year

Spring 2021

Due Dates

Mini-project 1 - Thursday, March 4 at 2 PM
Mini-project 2 – Tuesday, April 20 at 2 PM

Title

Cooling the Aluminum Cylinder Experiment to Illustrate Use of Numerical Methods

Points

200 Points (100 points for each mini-project)

Learning Objectives

- Identify the correct procedure to solve a given problem
- Implement a programming procedure for a given problem
- Improve existing programming skills of debugging, documentation, loops and conditional statements
- Write your own numerical methods programs
- Reinforce prerequisite knowledge of programming and college mathematics
- Solve real-world problems

Background of the Experiment

A solid aluminum cylinder treated as a lumped-mass[1] system is immersed in a bath of iced water. Let us develop the mathematical model for the problem to find how the temperature of the cylinder would behave as a function of time.

When the cylinder is placed in the iced-water bath, the cylinder loses heat to its surroundings by convection.

Rate of heat loss due to convection $=hA(\theta(t)-\theta_a)$. (1)

where

$\theta(t)=$ the temperature of cylinder as a function of time t, ºC
h = the average convective cooling coefficient, W/(m^2-ºC)
A =surface area, m^2
θ_a =the ambient temperature of iced water, ºC

The energy stored in the mass is given by

Energy stored by mass at a particular time = $mC\theta(t)$ (2)

where

m = mass of the cylinder, kg
C = specific heat of the cylinder, J/(kg- ºC)

From an energy balance,

The rate at which heat is gained **—** Rate at which heat is lost
= rate at which heat is stored

gives

$$0-hA(\theta(t)-\theta_a)=mC\frac{d\theta(t)}{dt}$$

$$-hA(\theta(t)-\theta_a)=mC\frac{d\theta(t)}{dt} \quad (3)$$

The ordinary differential equation is subjected to

$$\theta(0)=\theta_0$$

where

θ_0 =the initial temperature of cylinder, ºC

Assuming that the convective cooling coefficient, h to be a constant function of temperature, the exact solution to the differential equation (3) is

$$\theta(t)=\theta_a+(\theta_0-\theta_a)e^{-\frac{hAt}{mC}} \quad (4a)$$

It can now also be written in a normalized form as

$$\frac{\theta(t)-\theta_a}{\theta_0-\theta_a}=e^{-\frac{hAt}{mC}} \quad (4b)$$

[1] It implies that the internal conduction in the trunnion is large enough that the temperature throughout the trunnion is uniform. This allows us to assume that the temperature is only a function of time and not of the location in the trunnion. This means that if a differential equation governs this physical problem, it would be an ordinary differential equation for a lumped system and a partial differential equation for a non-lumped system. In your heat transfer course, you will learn when a system can be considered lumped or non-lumped. In simplistic terms, this distinction is based on the material, geometry, and heat exchange factors of the ball with its surroundings.

What we would have done in the laboratory

1. Fill the ice-cooler with half-water and half-ice. It is better to use the water from the water-cooler, as it is cooler than the tap water. Keep stirring the ice, so that ice cubes are not stuck to each other.
2. Take the thermocouple wires and connect them properly (+ to +, - to -) to the temperature indicator. Two thermocouples are attached to illustrate the concept of a lumped system.
3. Turn the temperature indicator on and wait for a few seconds to record the initial temperature of the cylinder.
4. Record the temperature of the iced water using a temperature indicator.
5. Immerse the aluminum cylinder in a bath of iced water and start the stopwatch simultaneously. Every five to ten seconds, record the temperature of the cylinder as a function of time.

Figure 1. Cooling the Aluminum Cylinder

Data Collected

The following data for temperature vs. time was taken a few semesters ago

Time (s)	0	4	11	16	19	25	29	60
Temperature (ºC)	22.0	19.2	14.2	12.4	12.0	10.9	10.3	8.5

The temperature of iced water = 1.2ºC

Grading

This project is part of the Computer Projects grade. Your solution will be graded on the following categories:

- The merit of the conceptual portion

- The merit of the programming portion
- The format of submission

Help

If you need assistance, some resources are listed below.

- Instructor office hours
- TA office hours
- Discussion board for the project
- How do I do that in MATLAB?

Academic Dishonesty

For this project, you may not receive ANY help from anyone outside of the instructor or the TA, or the CANVAS discussion.
Refer to the academic dishonesty policy of the University of South Florida.
Also, visit the ethics resources at USF for even more information.

Formatting

- Follow the sample project format, including introduction, cell formatting, published mfile format, commenting, typed pages, etc.
- Use MATLAB to solve all the problems, unless mentioned otherwise.
- Use comments, display commands, fprintf statements, sensible variable names, and units to explain your work.
- **Use the SI system of units throughout.**

What to submit

You will be uploading two files in response to two questions in a CANVAS quiz named Computer Mini project for **each** of the two mini-projects.

Question One

A single mfile needs to be submitted in response to the first question in the quiz. Name it as lastname_firstinitial_conv_spring21_x.m, where x is *one* or *two* depending on the mini-project number. For example, if your name is Abraham Lincoln, the name of your file would be Lincoln_A_conv_spring21_one.m for mini-project 1 and Lincoln_A_conv_spring21_two.m for mini-project 2.

Question Two

In response to the second question on the quiz, the whole mini-project report needs to be submitted as a **single** pdf file (learn how to make a single pdf file). See sample submission for reference.

The single pdf file would include
1) the published mfile (learn how to publish a mfile as a pdf file),
2) any typed pages,
3) affidavit sheet, and
4) checklist.

Why do I ask for a mfile separately?

There are two reasons: 1) Sometimes, while grading your project, the grader may not follow your logic. So, the grader may need to run your mfile. 2) Your mfile is put through a plagiarism checker along with mfiles from current and previous semesters. Some overlap is expected because of the nature of the program. Each program that is flagged for plagiarism is looked at manually as well for it to progress to an academic dishonesty case.

How to approach solving problems on paper

The following is meant to help students approach engineering problems effectively and efficiently. Without the proper approach, engineering problems can be very confusing. The following guidelines are written with common correct and incorrect approaches in mind. Remembering and implementing these approaches can not only help you find a solution faster, but it can increase your understanding of the problem and its conceptual basis. Most of these guidelines are not relegated to this class; you can use them in any engineering class!

- Start with what you know. If you do not know where to start, start with what you know. It is a little bit like connecting the dots. You cannot connect the dots until you have put some down.
- Look at the information you're given.
- Look at the applicable equations. What are the restrictions on these equations?
- Be methodical in your approach.

- Often students will say, "I don't know anything about this!" Typically, this is because they do not know what they know and what they don't know. Start with what you know!
- Use dimensions as a hint.
- If you cannot find a mistake in your work, check the unit consistency in the problem.
- If you do not know how to solve a problem, determine the units of the solution, and then look to see what units you're missing in the solution.
- Do not cut corners!

How to approach programming

- Start with what you know.
- If you have trouble programming a problem, start by working through the problem on paper.
- Don't try to think up the whole program in your head and then type it out!
- When translating the problem solution into a program, display each part of the code. Fix one piece at a time.
- Avoid using ";" at the end of statements while debugging the program. You can add the ";" later when the program is finalized.
- Look at the 'How do I do that in MATLAB series.'
- Use the MATLAB help site (http://www.mathworks.com/help/matlab/) to look up error codes, syntax, etc.
- If you're looking for syntax examples, click the "example" links on the right side of MathWorks sections for a sample program.

Common mistakes in programming

- Hard coding
- Incorrect format
- Misunderstanding the conceptual (paper) solution
- Inefficient program debugging
- The published mfile is cut off
- Errors in units / no units use /SI System not used
- Professional presentation lacking
- Missing comments
- Unsuppressed lines
- Vector data not in table form
- Questions are not read correctly, and hence answered incorrectly.
- Section numbers not matching problem numbers.
- Spelling and grammar mistakes

The preceding pages are common for both mini projects – Mini project 1 and Mini project 2

Mini project 1 (100 points) – Thursday, March 4 at 2 PM

1. Type the input data of temperature vs. time we collected and the following data, even if it is not used in the project, on a separate sheet of paper using a word processor like Microsoft Word. Use tables where appropriate.

   Diameter of cylinder = 51 mm
   Length of cylinder = 100 mm
   Density of aluminum = 2700 kg/m^3
   Specific heat of aluminum = 904 J/(kg-ºC)
   Thermal conductivity of aluminum = 241 W/(m-ºC)

   Table 1. Linear coefficient of thermal expansion vs. temperature for aluminum (https://nvlpubs.nist.gov/nistpubs/Legacy/MONO/nbsmonograph29.pdf)

Temperature (K)	Linear coefficient of thermal expansion (µm/m/K)
10	0.05
30	0.9
50	3.8
70	7.4
90	10.7
120	14.6
160	17.9
200	20.0
260	22.1
273	22.5
280	22.7
293	23.0
300	23.2

2. Assign all the required input data (experimental data and other data that is needed for the mini-project one to variables as MATLAB statements at the beginning of the mfile as one section. Do not change the units of the inputs – enter them as given. Of course, fprintf/sprintf/disp the input data using the variables.
   a. Any variables that are calculated from the input variables say the surface area should be done in the problem where you need them – not here, as this section is reserved for input variables only.
   b. Any changes in the input data should not require one to change any part of the rest of the program, and that is what is called "avoiding hardcoding."
3. Change the units of input variables to the SI system in a new section. No fprintf/sprintf/disp should be used in this section.

4. **Only** using the experimental temperature vs. time data, and no other data, use second-order polynomial interpolation to
    a. display the interpolant,
    b. estimate the temperature at $t = 15s$.

    You can use any MATLAB functions to do this problem.
    Your program should, however, be choosing the proper data points with no hardcoding. Assume that the time data is always in ascending order, and that there will always be at least two data points given for strictly less than $15s$ and at least two data points strictly greater than $15s$ in the temperature vs. time data, and that $t = 15s$ would never be one of the given data points.
5. Use the polynomial (do not hardcode the polynomial) from problem 4 and the Newton-Raphson method to find when the temperature of the cylinder would be $13°\mathrm{C}$. Choose a reasonable pre-specified percentage tolerance and a suitable initial estimate of the root. Both the pre-specified tolerance and an initial guess can be hardcoded.
6. Due to temperature change, estimate the change in the volume of the aluminum cylinder at the end of 50 seconds from when the aluminum cylinder was first immersed in iced water. Assume that the value of the convection coefficient is 570 W/(m^2-°K). You can use any MATLAB functions to do this problem. Hint: Coefficient of *linear* thermal expansion means that it can be applied for a *linear* dimension such as length, height, width, radius, diameter, etc. Hint: Refer to textbook Chapter 01.01.
7. **Only** using the experimental temperature vs. time data, and no other data, find the rate of change of temperature with respect to time at $t = 15s$ by **two** different numerical methods.
    a. Your program should, however, be choosing the proper data points with no hardcoding. Assume that the time data is always in ascending order, and that there will always be at least two data points given for strictly less than $15s$ and at least two data points strictly greater than $15s$ in the temperature vs. time data, and that $t = 15s$ would never be one of the given data points.
8. Attach a signed typed affidavit sheet (Your printed name can be considered to be the signature). The affidavit sheet is given here PDF DOC.
9. Attach a completed checklist that is given here PDF DOC. Checkmark the boxes you have accommodated in your assignment. Do not checkmark without thought.

Mini project 2 (100 points) – Tuesday, April 20 at 2 PM

1. Type the input data of temperature vs. time we collected and the following data, even if it is not used in the project, on a separate sheet of paper using a word processor like Microsoft Word. Use tables where appropriate.
   Diameter of cylinder = 51 mm
   Length of cylinder = 100 mm
   Density of aluminum = 2700 kg/m^3
   Specific heat of aluminum = 904 J/(kg-°C)
   Thermal conductivity of aluminum = 241 W/(m-°C)

Table 1. Linear coefficient of thermal expansion vs. temperature for aluminum (https://nvlpubs.nist.gov/nistpubs/Legacy/MONO/nbsmonograph29.pdf)

Temperature (°K)	**Linear coefficient of thermal expansion (μm/m/°K)**
10	0.05
30	0.9
50	3.8
70	7.4
90	10.7
120	14.6
160	17.9
200	20.0
260	22.1
273	22.5
280	22.7
293	23.0
300	23.2

2. Assign all the required input data (experimental data and other data that is needed for the mini-project two to variables as MATLAB statements at the beginning of the mfile as one section. Do not change the units of the inputs – enter them as given. Of course, fprintf/sprintf/disp the input data using the variables.
   a. Any variables that are calculated from the input variables say the surface area should be done in the problem where you need them – not here, as this section is reserved for input variables only.
   b. Any changes in the input data should not require one to change any part of the rest of the program, and that is what is called "avoiding hardcoding."
3. Change the units of input variables, if needed, to the SI system in a new section. No fprintf/sprintf/disp should be used in this section.

4. Find the crude value of the convective cooling coefficient h by using the temperature at the 4th data point. Use *vpasolve* to solve for *h.*
5. Regress the temperature vs. time data to the model

   $$\theta(t) = \theta_a + (\theta_0 - \theta_a)e^{-\frac{hAt}{mC}}$$

   to find the convective cooling coefficient h of the regression model.

   You are NOT allowed to transform the data to do the problem. You should not do any part of this problem by hand.
6. Draw plots in single figure that shows
   a. the temperature vs. time data as individual data points
   b. the temperature vs. time curves using the values of convective cooling coefficient h from problem #4
   c. the temperature vs. time curves using the values of convective cooling coefficient h from problem #5

   Use axes labels with names, symbols and units, figure title, and legends.
7. Find the second-order polynomial regression model that relates the coefficient of linear thermal expansion vs. temperature that best fits the data in Table 1. To find the second-order polynomial, you are required to do this by setting up the simultaneous linear equations. You CANNOT use MATLAB regression functions such as *polyfit*. Solving simultaneous linear equations is limited to the use of either the *linsolve* MATLAB function or the A\B MATLAB command. You may use the *sum* function to do summations. Display the second-order polynomial in a format with a reasonable number of significant digits.
8. Solve exactly the ordinary differential equation given by Equation (3) using a MATLAB function. For the convective cooling coefficient *h,* use the value from problem #5. Find the following at the 4th data point.
   a. temperature from the ODE solution and
   b. the corresponding percentage absolute relative true error.
9. Attach a signed typed affidavit sheet (Your printed name can be treated as the signature). The affidavit sheet is given here PDF DOC.
10. Attach completed checklist. Checkmark the boxes you have accommodated in your assignment. Do not checkmark the checklist without thought. The checklist is given here PDF DOC.