



EML3041: Computational Methods

Semester and Year

Summer 2020

Due Dates

Mini project 1 - Wednesday, June 24 at 11 AM

Mini project 2 – Wednesday, July 8 at 11 AM

Title

Cooling the Aluminum Cylinder Experiment to Illustrate Use of Numerical Methods

Points

100 points

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-mass¹ 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.

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

where

$\theta(t)$ = the temperature of cylinder as a function of time t , °C

h = the average convective cooling coefficient, W/(m²-°C)

A = surface area, m²

θ_a = the ambient temperature of iced water, °C

The energy stored in the mass is given by

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

where

m = mass of the cylinder, kg

C = specific heat of the cylinder, J/(kg-°C)

From an energy balance,

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

gives

$$\begin{aligned} 0 - hA(\theta(t) - \theta_a) &= mC \frac{d\theta(t)}{dt} \\ -hA(\theta(t) - \theta_a) &= mC \frac{d\theta(t)}{dt} \end{aligned} \quad (3)$$

The ordinary differential equation is subjected to

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

where

θ_0 = the initial temperature of cylinder, °C

Assuming 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	5	10	16	20	25	30	60
Temperature (°C)	22.0	18.2	15.2	13.4	12.0	10.9	10.3	8.5

The temperature of iced water = 1.2°C

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

What to submit

There are **two** submissions you have to make on CANVAS for **each** mini-project. Submission one is the whole mini-project report, and Submission two is just the mfile.

Submission One

The whole mini-project report submission on CANVAS will have the following submitted as a **single** pdf file (learn [how to make a single pdf](#) file). See [sample submission](#).

1. A signed typed affidavit sheet (Your printed name can be considered to be the signature). [PDF](#) [DOC](#)
2. Published mfile. Use any format that allows you to submit it successfully as a pdf file. Click here to learn [how to publish a mfile](#).
3. Any typed pages when asked for. Each answer needs to start on a fresh page.
4. Attach completed checklist given at the end of this assignment. Check-mark the boxes you have accommodated in your assignment. Do not checkmark this blindly.

Submission Two:

The mfile needs to be submitted on CANVAS. It is a single mfile.

1. Name it as lastname_firstinitial_conv_sum20_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_sum20_one.m for mini-project 1 and Lincoln_A_conv_sum20_two.m for mini-project 2.

Why do I ask for a separate mfile submission?

There are two reasons: 1) Your mfile is put through a plagiarism checker along with mfiles from 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. 2) Sometimes, while grading your project, the grader may not follow your logic. So, the grader may need to run your mfile.

Mini project 1 (100 points) – Wednesday June 24 at 11 AM

1. Type in a word processor the data of temperature vs. time we collected and the following data even if it is not used.

Diameter of cylinder = 50 mm

Length of cylinder = 100 mm

Density of aluminum = 2700 kg/m³

Specific heat of aluminum = 904 J/(kg·°C)

Thermal conductivity of aluminum = 241 W/(m·°C)

Table 1. Coefficient of thermal expansion vs. temperature for aluminum

(<http://www.llnl.gov/tid/lof/documents/pdf/322526.pdf>)

Temperature (°C)	Coefficient of thermal expansion (μm/m/°C)
-10	5.8
77	9.3
127	13.9
177	25.5
227	32.6
277	34.1
327	36.1
377	38.9
427	39.8

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 first section 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. **Only** using the experimental temperature vs time data, and no other data, use second-order polynomial interpolation to estimate the rate at which temperature is changing with respect to time at the time corresponding to the 4th data point. You can use MATLAB functions for this question.
5. Display the second-order polynomial you got in #4.
6. Without using MATLAB functions such as *interp* or *polyfit*, repeat #4 to find the second-order polynomial interpolant directly by solving a set of simultaneous linear

- equations. It should match what you got problem #5. You are allowed to use the MATLAB `A\B` command to solve a set of simultaneous linear equations, though.
7. Use the rate at which temperature changed with respect to time from #4 and the right-hand-side of equation (3) to estimate the rate of change of heat stored in the cylinder at the time corresponding to the 4th data point.
 8. Using the polynomial from #5, find the time when the temperature of the aluminum cylinder is 13.5°C. Use Newton-Raphson method and conduct a fixed number of 40 iterations. There is NO need to calculate relative approximate errors to meet prespecified tolerance. Use a suitable initial guess that is not hardcoded.
 9. Compare your result for #8 with the *vpasolve* command.
 10. Write a typed 200-250 word essay on "The ethics of cheating in writing a computer program". Have the title of the essay at the top in bold and centered. Use Times Rmn 12 point font, and double-spacing throughout the essay.

Mini project 2 (100 points) – Wednesday, July 8, at 11 AM.

1. Type in a word processor the data of temperature vs. time collected and the following data, even if it is NOT used.

Diameter of cylinder = 50 mm

Length of cylinder = 100 mm

Density of aluminum = 2700 kg/m³

Specific heat of aluminum = 904 J/(kg·°C)

Thermal conductivity of aluminum = 241 W/(m·°C)

Table 1. Coefficient of thermal expansion vs. temperature for aluminum
(<http://www.llnl.gov/tid/lof/documents/pdf/322526.pdf>)

Temperature (°C)	Coefficient of thermal expansion (μm/m/°C)
-10	5.8
77	9.3
127	13.9
177	25.5
227	32.6
277	34.1
327	36.1
377	38.9
427	39.8

2. Assign all the required input data (experimental data and other data that is needed for the miniproject 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 first 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. 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 and should start from the sum of the square of the residuals. You should not do any part of this problem by hand.

5. Plot in one figure the temperature vs. time data that shows individual data points and the temperature vs. time curve using the value of convective cooling coefficient

- h from problem #4. Use axes labels with names, symbols and units, figure title, and legends.
6. Using the value of convective cooling coefficient h from the problem #4, find the sum of the square of the residuals for #4.
 7. Using the value of the convective cooling coefficient h from #4, use the MATLAB *dsolve* function to exactly solve the ordinary differential equation (3) for the value of the temperature at the 4th data point.
 8. Using the value of the convective cooling coefficient h from #4, use Euler's method to solve the ordinary differential equation (3) for the value of the temperature at the 4th data point. Use a step size of 0.2 seconds.
 9. Estimate the change in the volume of the aluminum cylinder at the end of 60 seconds from when the aluminum cylinder was first immersed in iced water. You can use any MATLAB functions to do this problem.

How to approach solving problems on paper

This 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 don't 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! This WILL hurt you sooner or later.

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
- Unit errors/no units/SI System not used
- Professional presentation lacking
- Missing comments
- Unsuppressed lines
- Vector data not in table form
- Questions not read properly, and hence answered incorrectly.
- Section numbers not matching problem numbers.
- Spelling and grammar mistakes

Look at the checklist on the next page that needs to be attached to the hard copy of your submissions.

Checklist for submission

Name _____

Semester _____

Project Number _____

- ☐ I submitted this submission as a SINGLE pdf file.
- ☐ I followed the general format as given in the sample project.
- ☐ I uploaded the mfile as a separate submission.
- ☐ I attached the affidavit sheet.
- ☐ I wrote the code only by myself.
- ☐ I did not show my code to anyone else.
- ☐ I attached any handwritten pages if asked for.
- ☐ I attached any typed pages if asked for.
- ☐ I followed the section format as given in the sample project.
- ☐ I published the mfile in published format.
- ☐ I wrote proper and reasonable comments.
- ☐ I put the comments on their own lines, as seen in the sample project mfile (not at the end of a code line).
- ☐ I identified my methods for each problem.
- ☐ I suppressed all statements.
- ☐ I showed input and output variables using fprintf/sprintf/disp statements for all exercises unless specified otherwise.
- ☐ I checked for cut off errors in the published file.
- ☐ I avoided all hard coding (i.e., the program should still work if ANY of the input data is changed).