Computer Programming with MATLAB – Coursework 1

# Introduction

This coursework will help you revise the MATLAB knowledge gained in lectures 1-5. This coursework is worth 20% of the module credits and will cover the following topics:

* Variables and vectors
* Solving equations
* User input and formatted output
* Loops, conditional and switch statements

Please read the Assessment Guidelines below before attempting this coursework, as well as the mark sheet given on Moodle, so you know how the marks will be awarded.

**Follow all instructions carefully and check the Assessment Guidelines below before submitting your work.**

The **submission deadline** for this coursework is: **3pm, Monday 31 March 2025**

# Assessment Guidelines

* All answers are to be submitted in a single MATLAB script file (i.e., ‘.m’). The template for this file is available on the course Moodle page, under the heading: **MATLAB Coursework 1.**
* **Put your name and e-mail address at the start of the script** preceded by the ‘%.
* For questions which require you to write text (i.e., descriptive sentences), include these as comments using the '%' character at the beginning of the line. Include comments in your script file describing what the program is doing and any features of note – you will get marks for this.
* Save your script with a name in the following format: Firstname\_Surname\_studentID.m.

For example:

**Julia\_Smith\_133444232.m**

* Save the flowchart for Question 5 in a pdf file in the following format: Firstname\_Surname\_studentid.pdf.
* Create a zipped file containing the two files, using the format Firstname\_Surname\_studentid.zip
* Upload the zipped file to Moodle in the submission box in the same section. Ensure that you upload before the submission deadline, as late submissions without exceptional circumstances will be penalised.

**A WORD OF CAUTION** – Remember this is individual work, not group work. Work handed in must be entirely your own and not copied from anyone else. Discuss the coursework with your friends if needed, but answer the questions yourself.

# Question 1 – Variables [8 MARKS]

Evaluate the following functions at :

**Hint:** Parts *f* and g can be solved without typing out the full equations.

# Question 2 – Vectors [12 MARKS]

Write a script to generate a vector called with 100 points between and . Then evaluate the following functions between and for each of the values in the vector:

Add a line to the script to do the following:

1. Calculate the mean and standard deviation of the data in array .
2. Add comment text to the script to answer the following: Why is the mean of not exactly 0.5?
3. Use the same methods as above to solve this problem concerning an electrical AC circuit. If the circuit operates with a peak current of 100 mA (see figure below, shown over one complete cycle), what would be the *average power* dissipated by a 1 kΩ resistor? You can assume this is a simple circuit featuring only a resistor, with no capacitor or inductor.

A graph of a peak current

Description automatically generated

The average power,, is given by:

where is the square root of the mean squared current and is the resistance.

# Question 3 – Solving Equations [6 Marks]

1. Write code to find the roots of the following polynomial expression:

You may use the quadratic formula:

1. The speed of light in a vacuum is given by:

,

where:

is the speed of light,

is the permittivity of free space,

is the vacuum permeability,

* 1. Write code to calculate the speed of light from these constants.
  2. The James Webb Space Telescope conducts infrared astronomy using wavelengths from 0.6 µm to 28 µm. Write some code to find the range of electromagnetic frequencies this represents, using your calculated value of the speed of light. (You many need the formula , where is wavelength, and *f* is frequency).

# Question 4 – Examples [12 marks]

Give examples of the use of the following commands and describe concisely what they do. (Hint, the last two can be found using the help command). Use the '%' character to include text comments in your .m file:

1. clear
2. who
3. a semicolon after a line
4. sound
5. roots
6. abs

# Question 5 – Program Flow [12 Marks]

A space capsule will re-enter Earth’s atmosphere from space. The computer in the capsule uses radar to check how high it is above sea level every 2 seconds. When the capsule reaches 2,700m from the sea, it deploys a parachute to safely descend.

Once the capsule hits the sea, the computer turns on a transponder beacon communicating the capsule’s position to rescue ships.

1. Deploying the parachute and turning on the transponder beacon are handled by a single program. Using boxes to represent actions, and diamonds to represent decisions, sketch the structure of this program as a flow chart. The program must contain the following boxes:

* Is the capsule at or below 2?
* Is the capsule at or below ?

You can use any program you want to do the sketch. Export it to a PDF to make sure the marker can open it. Include the PDF in your submission on Moodle with the name Firstname\_lastname\_ID.pdf.

1. Now write a Matlab script to implement your program. You will need loops and if statements. You can use comments to represent the capsule actions; e.g., % parachute deployed. Remember what you learned in class about when to use for loops and while loops.
2. Use your script to answer this question: Assuming the capsule enters the atmosphere with an initial height of 10,000 m and has an unchanging velocity of 200 m/s throughout its descent, how many seconds elapse before the parachute is deployed?

**Hint:** Define the initial height and velocity towards the start of the script, then use a while loop to progress time forward and recalculate the height of the capsule at each iteration

# Question 6 – Format and print text to screen and loops [13 marks]

This question lets you practice the *sprintf* command, which is used in Matlab and other programming languages. It is important to be able to format computer text output, especially when dealing with limited size displays.

Read parts a) and b) before proceeding. You should be able to do both together.

**Hint:** you can find instructions for the *sprintf* command by typing *doc(sprintf)* in Matlab.

1. You are tasked with writing part of the control software for a weather monitoring device. Your part of the program prints the recorded weather data to a screen. Using the *sprintf* and disp commands, create a script to print the information in the box below. You should be able to format the information exactly as below, including:

* Separate lines for data entries.
* Spaces between lines.
* Correct alignment of the data entries using tabs.
* Any special characters.

To make your script flexible, it should be able to display any data, not just those given below. Create arrays containing the time, temperature, humidity and UV data in separate arrays defined at the start of the script, each containing three entries, which are then input to your sprintf commands.

Use a loop so that the code to print each section of data is not repeated.

Data logging initiated - 27/2/2025

Location – Ningbo

Time 1300

Temperature 19 C

Humidity 55%

UV level 4

Time 1600

Temperature 20 C

Humidity 49%

UV level 2

Time 1900

Temperature 18 C

Humidity 59%

UV level 1

Data logging terminated

1. What is the main limitation of your script in its current form? Say how you would improve the program to make it more useful for the weather monitoring device. Give your answer as a comment. (But there is no need to make any further changes to the code)

# Question 7 – For loops and displaying data [16 marks]

This question lets you practice for loops and the *fprintf* command, which is used in Matlab and other programming languages.

**Hint:** you can find instructions for the *fprintf* command by typing *doc(fprintf)* in Matlab.

You work for an architect who wants to build a nice looking house. He remembers that the golden ratio can be used to designate visually pleasing structural proportions, but he can’t remember the number itself. Luckily (and because you happen to have no internet connection to search for the answer), you know it is possible to calculate the golden ratio using the well-known Fibonacci sequence. Write a program listing the Fibonacci sequence and corresponding approximations to the golden ratio.

1. Start by writing a for loop to calculate the first 50 terms of the Fibonacci sequence, *F*. You may need the formula:

- First term - Second term - *n*th term

1. Modify your for loop to also calculate an approximation to the golden ratio, *φ,* for each of the entries in *F*. *φ* is defined by:
2. Using a for loop, write each of the 50 entries of *F* and *φ* to the terminal on a new line, including an index and with the columns of numbers aligned. Add a line of text at the start of the output saying what the numbers mean (e.g., ‘Fibonacci sequence and approximations to the golden ratio’.) The architect only needs the golden ratio given to 5 decimal places, so make sure your code does this.
3. From the data in the Fibonacci output you should see that differences between your approximations of *φ* become smaller with increasing iteration *n,* so many of the entries are not really needed. Add new code to your script which does the same as part c), but stops writing the *F* and *φ* sequences when the difference between one *φ* value and previous one drops below 0.1%, therefore giving a good enough approximation for the architect to work with. This new sequence should, again, be output to the terminal. At the end of the output, write the message ‘Golden ratio converged to within 0.1%’.

**Hint:** There are several ways to achieve part d). You might want to use a for loop with an if‑else statement inside. The if statement would check if the difference is above 0.1%, and only write the entries to the terminal if that condition is met. If you use this method, the *break* command might be useful as a way of stopping the for loop once your convergence criterion is met. Marks will be awarded regardless of method as long as the program works.

What are the main benefits of using the approach in part (d) compared to part (c)? Write your answer as a comment in the Matlab script.

# Q8 – SWITCH STATEMENT [10 marks]

Write a script which will enable you to convert from GB Pounds to other currencies. It should fulfil the following specification:

* The number of GB Pounds to be converted should be user input. Your program should only continue to the conversion part of the program if a positive number has been entered.
* Use the menu function to give the options for currencies to be converted to.
* Use a switch statement to output a message with the converted currency values depending on the currency selected, eg ’10 GB Pounds is 11.70 Euros’ . Display an appropriate message if the user quit the dialog without making a selection.
* Display values to 2 decimal places.
* Use the table below for the currencies and conversion rates to be used:

|  |  |
| --- | --- |
| Currency | Conversion Rate |
| Euro | 1.17 |
| US Dollars | 1.27 |
| Chinese Yuan | 9.03 |
| Swiss Francs | 1.09 |

# Presentation and Formatting [11 Marks]

1. Presentation is neat and tidy – 5 marks
2. All codes are commented throughout – 5 marks
3. Name/email is specified at the top of the script – 1 marks