

Mini Project

Quadratic Equation Solver

The Design Problem

For this mini-Project, you will write a MATLAB program designed to perform simple calculations involving quadratic equations of the form shown below. Your program will be designed to interact with a user and process data according to their response.

$$y = ax^2 + bx + c$$

Things you will need to consider include;

- 1) Asking the user for values a, b, and c. The range for x and the step size to be considered when performing the calculations and plotting. You could start by finding the roots of the equation from the inputs.
- 2) Error checking for the above inputs e.g. are they sensible values? Are they numerical values? etc. Also, your program should decide what to do if the above inputs are not valid.
- 3) Plot the graphs of 'y against x' using the range and step size entered by the user. Include axis labels, title etc. if and when appropriate.
- 4) Find the intersection points with the x axis... taking care to ensure that your program handles different possible scenarios appropriately e.g. what if there are no intersection points (Matlab will do this without a problem, but I want you to consider the best way to handle the situation and present results to the user)
- 5) Where should the intersection points (from 4 above), be displayed? On your plot would be nice!
- 6) Calculating and plotting values for the equations is easy ... it's the error handling and user interaction aspect I want you to concentrate on!
- 7) All code should be fully self-documented i.e. you should make VERY extensive use of 'Comment statements' What you will be expected to hand-in
- 8) For those who want to take the code to the next level here are some open ended challenges: Can your code take straight and quadratic equations and find their intersections? What about two quadratic equations? If there is more the one equations, how would you like to present them graphically?

For this assignment, you will submit the following:

- **your final** MATLAB code solution to this project (i.e. the '.m' function that YOU have written)

Note that for this assignment, you have **not** been asked for a formal report however you **MUST** make sure that your MATLAB code is appropriately documented using comment statements throughout.

Also Please Note:

1. I will use MATLAB itself to help me detect plagiarised script files i.e. ones that are too similar to each other (MATLAB has excellent tools to help me do this!). I will heavily penalise plagiarised work and that includes both the originator and the submitting student. Work on your own and submit only your own work. Don't share your files with anyone else!
2. I am looking for your solution to the problem. I don't mind if it's not the most efficient or elegant solution - but I am interested to see your own approach. There are many ways to crack a nut!
3. Your MATLAB Code '.m' file must be submitted with the name: '**MPYourStudentID.m**' ... no spaces, no hyphens etc. For example, if your ID is: 4191234 then you would submit a MATLAB function file called: MP4191234.m
4. If you forget to substitute 'YourStudentID' with your actual student ID, and/or forget to include the prefix 'MP' or include any brackets or hyphens in the file names, exactly as specified here, then your mark will be downgraded!

Basis for Marking the Project

The following list gives an indication of where marks have been allocated for this assignment.

| Assessment Criteria | % | Notes |
|--|-------------|--|
| Submission guidelines followed accurately (Y/N) | 10% | Have you followed my submission guidelines to the letter? Or have I had to re-format your submission in some way? |
| Does the code perform the basic task? | 20% | Does the code plot the equations and takes in the correct inputs? |
| Can the code handle finding the roots and giving the information back to the user? | 10% | Does the code handle different root outputs? How is the information given back to the user? |
| Appropriate/relevant use of Comment Statements | 15% | Have you included complete, useful and relevant in-application documentation i.e. Comment Statements? |
| Code Layout & Design Logical/Efficient/Appropriate? | 10% | Code follows a logical layout, sensible use of variable names & easy to navigate? Code is efficient (run time will be assessed and compared to other submissions). |
| Error Checking Appropriate and Relevant | 5% | Is your error checking correct/extensive and appropriate? Can your program be fooled/tripped-up in anyway? |
| Code extended using own methods (innovation) | 30% | Does your code design show innovation and design flair above and beyond the norm? Individuality??? |
| TOTAL | 100% | |

HOW to SUBMIT YOUR PROJECT

ALL project solutions must be submitted through moodle.nottingham.ac.uk

Attachments: one '**MPYourStudentID.m**' function file and ...

Deadline: Friday 7th November (by 3pm)!!

WORKING IN THE LAB

For this assignment, lots of support is available to help you get going and to complete the task on-time and successfully. As long as you attend the sessions, ask for help when you are stuck and get engaged properly there should be no problem at all with the task (even if it seems a bit daunting at first).