MAT2409 ASSIGNMENT 3
SEMESTER 1, 2019

WEIGHT: 10% TOTAL MARKS: 20

Due date: Friday 10th May, 2019 11:55pm AEST*

Submission instructions

The assignment will be electronically submitted via Study Desk. If you cannot submit electronically please contact the Examiner as soon as possible to make alternative arrangements.

You are to submit your MATLAB code for the assignment as a **single compressed archive file** (e.g. zip, tar-gzip, tar-bzip2, 7zip, or rar). This archive file should include all appropriate files to run your code (including input files). It should also include a README text file which describes how the code implements the solution and other information necessary to run the code.

If you have trouble submitting your assignment via the Study Desk etc., please contact the Examiner, USQAssist or via phone ASAP.

Late submission of Assignments[1]

Students can apply for an extension of time to submit an assignment at any time up to the deadline. Students are advised to make a request for an extension as soon as their need becomes apparent. Delay in making a request involves the risk of losing marks if the request is refused.

The Examiner may grant a short extension of the deadline for submission of an assignment. Extensions are usually granted only in cases of Compassionate and Compelling Circumstances in accordance with the assessment of Compassionate and Compelling Circumstances Procedure. Generally, extensions will be limited to a maximum of five University Business Days. A Student requiring an extension for a period of time in excess of this should consider applying for a Deferred Assessment as per section 4.4 of the assessment procedure.

Applications for extensions must be made via email or USQAssist to the Examiner together with accompanying documentation as specified in the Assessment of Compassionate and Compelling Circumstances Procedure.

* Australian Eastern Standard Time

[1] Full assessment procedure can be found at http://policy.usq.edu.au/documents.php?id=14749PL.

An Assignment submitted after the deadline without an approved extension of time will be penalised. **The penalty for late submission without a pre-approved extension is a reduction by 5% of the maximum mark applicable for the assignment, for each University business day or part business day that the assignment is late. An assignment submitted more than ten University business days after the deadline will have a mark of zero recorded for that assignment.**

The Examiner may refuse to accept assignments for assessment purposes after marked assignments and/or feedback have been released.

Non-submission of Assignments

As per the assessment procedure outlined at `http://policy.usq.edu.au/documents.php?id=14749PL` — A student who has failed to achieve a passing final grade by 5% or less of the aggregated weighted marks, the Examiner, in agreement with the Moderator, will consider recommending to the Board of Examiners the undertaking of supplementary assessment by the student. **This offer will normally only be made if the student has undertaken all of the required summative assessment items for the Course—that is, submitted all of the assignments!**

Student Responsibilities

The assessment procedure also outlines the following student responsibilities:

- If requested, students must be capable of providing a copy of assignments submitted. Copies should be despatched to the University within 24 hours of receipt of a request being made.
- Students are responsible for submitting the correct assignment.
- Assignment submissions must contain evidence of student effort to address the requirements of the assignment. In the absence of evidence of student effort to address the requirements of the assignment, no mark will be recorded for that assessment item.
- A Student may re-submit an assignment at any time up to the deadline. A request to re-submit after the deadline is dealt with in accordance with Section 4.4 'Deferred, Supplementary and Varied Assessment and Special Consideration' of the assessment procedures.

Academic Misconduct

Academic misconduct is unacceptable and includes plagiarism, collusion and cheating.

- Plagiarism involves the use of another person's work without full and clear referencing and acknowledgement.
- Cheating involves presenting another student's work as your own.
- Collusion is a specific type of cheating, that occurs when two or more students fail to abide by directions from the examiner regarding the permitted level of collaboration on an assessment.

All are seen by the University as acts of misconduct for which you can be penalised. For further details go to: http://www.usq.edu.au/library/referencing/what-is-plagiarism.

Assignment notes

- Each function needs to be properly commented including a clear description of the purpose of the function, its parameter list and its return values.

  For example, in MATLAB this means that a user should be able use the `help` command to obtain information on how to use each function.

- A text README text file which describes each of these functions and how they are combined to address the task should also be included.

- All output data and plots must be properly labelled—including legends where required. Legends should not obscure plots.

- All output must be correctly formatted. Significant figures must reflect the convergence criterion used.

- Each function should make use of vector/matrix operations where possible.

- As this is real data you need to think about how you report any data issues you find. This should be documented in your code.

- *Functions and methods must be implemented as per the specification given in the Assignment.* Also functions *can only use methods discussed and presented in the course.* MATLAB functions that use extended methods are not allowed, and marks will deducted accordingly. However, primitive functions, such as, `rand, randn, sort, sqrt, sin` etc. are fine, while functions, such as, `fzero, roots, lsqcurvefit, linsolve`, etc. are not acceptable. If in doubt about a function, email the Examiner and ask.

- The *symbolic toolbox* **cannot be used.** If you are defining symbolic variables in your code you are using the symbolic toolbox (e.g. `syms x`). Your code should not contain any of these definitions.

- All the appropriate files (including input files) should be submitted as one compressed file (zip, tar-gzip, tar-bzip2, 7zip, or rar). This single file should be uploaded via the StudyDesk.[2]

  In Windows : the zip archive file can be created using software such as, `7-Zip`. A link to the `7-Zip` software can be found under the Resources link on the Study Desk.

  In Linux : the tar-gzip (or tar-bzip2) archive file can be created using the command `tar`.

  In MacOSX : the zip archive file can be created using the Finder.

- The `pause` command in MATLAB and `halt` command in SCILAB should be used in the code to break up the various sections of the assignment in the main driver script.

- Breakdown of marks:
  - Task 1: 2 marks
  - Task 2: 3 marks
  - Task 3: 3 marks
  - Task 4: 4 marks
  - Task 5: 4 marks
  - Task 6: 4 marks

[2] If the upload fails then the assignment can be emailed to the lecturer at `MAT2409@www.sci.usq.edu.au` with the words “MAT2409 ASSIGNMENT” in the subject — but only do this as a last resort.

Assignment Task

The challenge is to find the intersection points of two curves as a parameter p varies. Consider the following two curves

$$\begin{aligned} x^2 + 9y^2 &= 16\,, \\ y - x^2 + 2x &= p\,, \end{aligned}$$

where $0 \le p \le 2.5$.

1. Using MATLAB to plot the behaviour of the intersection points of the two curves as the parameter p is systematically changed (i.e. plot the intersection for $p = 0,\ 1$ and 2).

   Hint: Plotting an ellipse can be problematic. The simplest way if possible, is to recast the equation of the ellipse in canonical form—

   $$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1\,,$$

   and then calculate x, y using the canonical form's parametric solution–

   $$\begin{aligned} x(t) &= a\cos(t)\,, \\ y(t) &= b\sin(t)\,, \end{aligned}$$

   where $0 \le t < 2\pi$.

   Hint: When attempting to plot relations always use the parametric form wherever possible.

2. Write a MATLAB function that returns the function values and the Jacobian needed for Newton's method for the above system.

3. Write a MATLAB function to solve the above equations for a given value of p using Newton's method for non-linear systems (see Algorithm 14 from the Study Book as an example).

4. Using the Newton's method function developed in Practical 5 as a base, modify it to solve the above pair of equations for $p = 0$, $p = 1$ and $p = 2$.

   Use the estimates from Task 1 as your initial guess when $p = 0$. As p is changed use the solution for the *previous* value of p as your *initial* guess of the solution for the *new* value of p.

5. Now do something that you may find difficult.

   First, recognise that the computation method in Task 4 is not well vectorised: it does a lot of computation on scalars and on vectors with a pathetically short length only two. Write a new Newton's method function that is invoked just once and **simultaneously** computes solutions for all values of input parameter p.

   Note: instead of computing and storing the Jacobian as a 2×2 array, you will need to compute and store as a Jacobian for each value of p, say store the four elements as a $4\times n$ array where you are solving for n values of p. Then, instead of using MATLAB's linear equation solver `x=J\f`, explicitly solve all the 2×2 equations using Cramer's Rule. Assuming:

   $$J = \begin{bmatrix} J_1 & J_2 \\ J_3 & J_4 \end{bmatrix};$$

   we get

   $$x_1 = (f_1 J_4 - J_2 f_2)/\det J\,, \text{and } x_2 = (J_1 f_2 - f_1 J_3)/\det J\,,$$

   where $\det J = J_1 J_4 - J_2 J_3$.
   You will need the vector `.*` and `./` operators.

6. Plot the solutions x and y as a function of parameter p. That is, show graphically how the intersection point of the two equations varies with p.

END OF ASSIGNMENT