Project 1 – Newton-Raphson

David Balderas & Mario Rojas

Instructions:

Objectives: Find out the roots of the different equations along the path for the first problem and for the different intervals for the rest.

Method: Suggest and implement a solution to the proposed problem and delivered a single program that MATLAB completely resolved (per team) as well as a report to explain your answers.

Evaluation: The report must contain: Title / Author / Summary / Introduction / Methods / Results / Discussion and Conclusions / Attachments (here's the code) / References. The proposed solution is evaluated, the results obtained and how they were obtained (according to the document "Understanding the publishing process"). All items are weighted according to the document "Report Checklist". There are several parts for this project, each one will be evaluated separately in the methodology and should have a concrete introduction for the whole project and individual introductions if needed.

- The delivery is considered in two parts: MATLAB implementation with an individual report for the teacher. Although the code will not be evaluated it will be taken into consideration to validate your answers. If it is found out that the answers do not correspond to your answers the report will be considered invalid and it will be consider a DA on the assignment.
- The code and the report must be send to the blackboard link available. No physical documents are accepted.

Tips: Consider the accuracy of their results, whenever you need a value between two known points, /Use encapsulated numerical methods in functions / Make use of secondary functions to sort your final program / Be clear in your report and program, doing otherwise is the easiest way to lose points and offer solutions incomplete/ Add comments to your code, and adequately report.

Expected results: This time the numerical results of the problem are not included, however some considerations do in this regard:

- The answers to the questions raised in the project do not validate your solution, the qualification will be given accordingly to the methodology used and the results obtained, not whether or not answered these questions.
- You are not allowed to use MATLAB functions to solve predefined methods seen in class (or Similar). Using a method that is not owned by the students will be cause for DA.

There is no single way to solve it, however, the results achieved must be accurate according to the chosen method, consider the importance of numerical methods is actually used or even modify them to get very close to the actual result. Note that this does not imply that the programming can be done without relying on conventional methods using excessive ifs and fors for example. If a new method is proposed for either of the solutions it must presented with sufficient evidence that the method can be used to solve other problems as well as their mathematical basis.

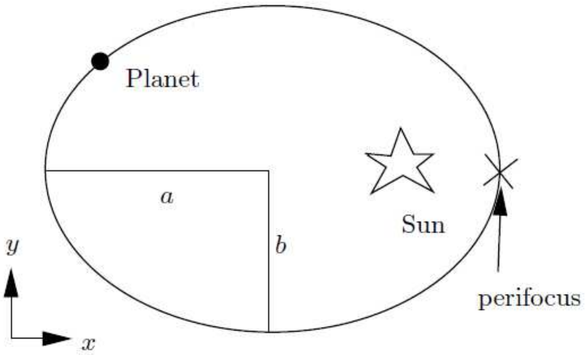
The difficulty of solving the problem is (like most of the time) in the approach to their solution and the data flow and not on the complexity of numerical methods or programming.

Final Recommendations

- Be sure to write your name and student number.
- Delivers after the due date are not accepted.
- As a reference, you can use the code available in: http://www.cs.cornell.edu/Courses/cs321/2003fa/Matlab%20Coding%20Style.pdf

Problem 1

One of the classical laws of planetary motion due to Kepler says that a planet revolves around the sun in an elliptic orbit as shown in the figure below:



Suppose one needs to find the position *(x, y)* of the planet at time t. This can be determined by the following ellipse formula:

$$x = a\cos(E - e)$$
$$y = a\sqrt{1 - e^2}\sin E$$

Where
e = The eccentricity of the Ellipse
E = Eccentric anomaly

The eccentricity of the ellipse is defined as the relation, $e = \frac{c}{a}$, where c is the distance from the center to the focus of the ellipse and *a* is the distance from the center to a vertex.

To determine the position *(x, y)*, one must know how to compute E, which can be computed from Kepler's equation of motion:

$$M = E - e\sin(E), \qquad 0 < e < 1,$$

where M is the mean anomaly ($0 < M < 2\pi$). This equation, thus, relates the eccentric anomaly, E to the mean anomaly, M. Thus to find E we can solve the nonlinear equation:

$$f(E) = M - E + e\sin(E) = 0$$

The iterative process ends when the relative change in E is less than the tolerance, ε:

$$\left|\frac{E_{k+1} - E_k}{E_{k+1}}\right| \leq \varepsilon$$

a) Make a general code in MATLAB to solve the nonlinear equation by the Newton-Raphson method, thus the value of E is obtained as we introduce any eccentricity (*e*) and any mean anomaly (*M*).

b) Solve the equation for all the values of *M* and calculate the correspondent *(x, y)* positions to plot the complete ellipse.

Use the next values:

a = 5 x 10^{11}, b = 4 x 10^{11} For the first Ellipse and a = 6 x 10^{11}, b = 3 x 10^{11}

Use tolerance as $\varepsilon = 0.000001$ and limit of iterations equal to 20.

Note: remember to calculate for all values of M (from 0-2π).

Problem 2

Find the roots of the function $f(x) = x^3 - \frac{31}{10}x^2 + \frac{1}{10}x + \frac{21}{5}$

a) Starting at x =0.0161, find a root in the left of the point and another in the right.

b) Starting at x =2.051, find a root in the left and another in the right of the point.

c) Starting at x =0.5, find a root in the left and another in the right of that point.

Propose and implement a method to know that there is any root by the left or right side of some point in a function. Evaluate the method in the last function with x=-2 and x=3.

Problem 3

Find all the roots for the polynomial$f(x) = x^4 - 10x^3 + 27x^2 - 2x - 40$, over the $(-3,4)$ interval and by using a single run. (Note: the interval is open).

*Optional: make a code to find the root of $ln(x)$ and $tan(x)$

Instructions:

- ¿If the roots is searched by the left of the point and the method finds it by the right, what is the implication?
- ¿Which is the problem when some zero or close-zero point is evaluated at the derivative?

- Use 50 iterations as the limit.
- Use a tolerance of $10e^{-5}$