Assorted numerical methods (Evaluated Task Set 3)

Soft deadline: 2020-05-26 00:00:01. After this date there will be 10% penalty for being late.

Hard deadline: 2020-05-30 00:00:01. After this date no submission will be accepted for evaluation.

Submission requirements: Same as for the Evaluated Task Set 1. You should also generate reports for each task and add them to your submission.

Due to the quarantine you will have to submit your solutions through e-learning platform. Archive all files in "Surname-ETS3" to a single zip archive (named "Surname-ETS3.zip").

Evaluation criteria: Is the submission on time? Does submission follow the requirements? Is the code clear and does it follow good coding practices? Are the input prompts proper? Is the output properly formatted? How well the task is performed? How well the author can explain his code/idea/approach? Is the approach reasonable? Note that in this task set you are already encouraged to be as "efficient" in "Matlab way" as you can.

Note: You will need to download and execute some files from the e–learning platform.

Tasks

1. Systems of linear equations. We are building a new (and better) version of Dinas Zauras for the Physicists Day. Your particular task is to buy parts and transport them to the faculty. You have been given a budget of 17500.71 Eur to buy new parts and 11 helpers to carry those. Your options for new parts include flamethrowers (at 3000 Eur/piece, can be carried by a single person) and car engines (at 2500 Eur/piece, can be carried by four people). As the money was provided by the dean's office, you might want to spend everything (or as much as you can) and make Dinas as "dinosauric" as you can (though we still need at least one engine). Your helpers will be extremely unhappy if at least one person has nothing to carry. Also your time is tight, so you can visit the shop only once. Solve this problem in two different ways using **inv** and **rref** functions. Check the answers you get match. Check if the answers are actually correct. Note that this problem has a few catches (both analytical and numerical), in your comments provide an explanation for the adjustments you made (for example, you are unlikely to be able to spend those 71 cents).

2. Root finding. Using fixed point iteration method find as many real roots of polynomial $f(x)$ as you can. Compare your results with the answer provided by the **roots** function. Provide an explanation for the differences (if any) in your comments. Generate a plot which would show 5 different convergence trajectories. You will have to generate $f(x)$ yourself by using "generateFx" function which is available on the e–learning platform (from the same page you have download this PDF file). You just have to pass your name and surname as two input arguments (translate non–latin letters into their latin equivalents (e.g., "č" → "c")):

```
>> generateFx('Namey','Surnamey')
ans =
        4 1 -8 0 -4 6
```

   So the student named Namey Surnamey would have to find roots of:

$$f(x) = 4x^5 - x^4 - 8x^3 - 4x + 6.$$

   To get values of polynomial use **polyval** function. Do NOT hardcode (enter by hand) polynomial.

3. Interpolation and extrapolation. For this task you will need to pick one General Physics laboratory work you did. This laboratory work should have a significant number of associated data and involve fitting of that data. Your task is to fit the data and provide a nicely looking figure (labels, title, legend, grid should be present and be informative). Be sure to print out fit parameter estimates and their respective error (uncertainty). In your comments explain what you have measured and how. Hint: use **fit** function.

4. Numerical differentiation. Explore numerical errors in context of simple forward difference numerical differentiation method and Richardson's numerical differentiation method. Calculate numerical derivative of the same polynomial as in the Task 2 at one of the roots you found. If you failed to find any root, then consider point $x = e$. Compare your numerical result against analytical (use **polyder** to get it). How does the absolute difference depend on the step size h? Do NOT hardcode (enter by hand) any polynomial.

5. Numerical integration. From the same page you have download this PDF get "task5.zip" file. This archive contains a Matlab program, which aims to approximate the following integral numerically:

$$I = \int_0^\pi \frac{\ln(1+x)}{x} \mathrm{d}x.$$

   There are multiple errors of different kinds. Find, fix and explain them. Provide documentation and comments as necessary. Note that analytically this integral evaluates as $I \approx 2.00411$ (after debugging you should get similar result).