UNIVERSITY of NICOSIA

MATH-335: Optimization Techniques
FALL 2023

Computational Assignment:
Applying Gradient Descent

INSTRUCTIONS :

- You must ***show all your work***. The examiner will attach great importance to the quality and completeness of your answers.

- Make sure you strictly ***adhere to the deadline*** on the course Moodle page.

For this Assignment the Submission Instructions will be slightly different. In addition to the single .pdf with your handwritten calculations and results, you will also have to submit:

1. All the source files and or scripts needed to generate your results. So you can submit Python (.py), MATLAB (.m) or Gnuplot (.plt) scripts, Mathematica (.nb) or Jupyter (. ipynb) notebooks. Be especially careful with Jupyter notebooks. I should be able to execute your submitted files and reproduce your results.

2. Screenshots in .jpg, .png or .pdf format of your terminal after the code has run and produced the results, or the text file (.txt) to which you are outputting the results.

3. A .jpg, .png, .pdf or .eps file with the plot as described in (d) of the item description.

- Before uploading, you **must** put all your files in a **single folder** named Firstname_Surname and then **compress** it to .gz, .gzip, .rar, .tar or .zip format to avoid having any issues with Moodle. Please only upload the compressed folder.

- ***NO late submissions will be accepted.***.

DESCRIPTION :

Consider the bivariate objective function $f(x, y) = x^2 + y^2 + xy - 5x - 7y + 20$. You are to calculate the exact relative (and absolute) minimum by hand as well as approximate it using gradient descent. For more information, read the relevant lecture notes and textbook sections.

Specifically, you are to :

(a) Plot $f(x, y)$ on $-2 \le x \le 4$, $-2 \le y \le 4$. *(Code & image file with plot)* *(10 pts)*

(b) Use **Calculus** to find the **exact** point $\boldsymbol{x_{min}} = (x_{min}, y_{min})$ at which f attains its relative minimum, as well as its minimum value f_{min} . Why is this also the absolute minimum? *(Scanned handwritten)* *(20 pts)*

(c) Formulate gradient descent for the objective function f. The learning rate should be $\alpha = 0.4$ and the initial point $\boldsymbol{x}^{[0]} = (x^{[0]}, y^{[0]}) = (2, 2)$. *(Scanned handwritten)* *(5 pts)*

(d) Write and execute **your own code** in order to implement the algorithm you formulated in (c). The convergence criterion is $|f(\boldsymbol{x}^{[k+1]}) - f(\boldsymbol{x}^{[k]})| < 10^{-5}$ where $\boldsymbol{x}^{[k]} = (x^{[k]}, y^{[k]})$ is the k-th gradient descent iterate and $f_k = f(\boldsymbol{x}^{[k]})$ is the value of the objective function at that point. *(Code & screenshot of output)*

Your code must correctly implement the algorithm *(20 pts)* be well-commented *(10 pts)* and should output:

i. The **approximation** $\boldsymbol{x^*} = (x^*, y^*)$ of the location of the minimum. *(20 pts)*

ii. The corresponding value $f^* = f(\boldsymbol{x^*})$ of the objective function. *(5 pts)*

iii. The iterate number n at which convergence occurs. *(5 pts)*

(e) Use a computational utility and a ready routine/command to confirm your results. *(Code & screenshot)* *(5 pts)*

(f) Plot the values of the objective function f_k as a function of the iteration number k. Note that $k = 0, 1, 2, \ldots, n$, where n is the iteration number at which convergence is reached. *(Code & image file with plot)* *(10 pts)*

Note: There are 10 bonus points available.