Coursework 2: Optimization

6CCE3EAL Engineering Algorithms

David Moxey and Rym M'Hallah, 29th June 2023

- This coursework will count for **30%** of your total grade in the course.
- Remember that all work you submit should be **your own work**. Although you can e.g. discuss your approach to the problem at hand with others, this coursework is not meant to be a team exercise. The College treats plagiarism very seriously.
- You should submit your report by **16:00, Thursday 27th July 2023** using the submission link on the KEATS site for 6CCE3EAL.
- If you have more general or administrative problems please use the online forum or consult during drop-in-sessions or support sessions. If an email is required, please include the course number (6CCE3EAL) in the subject line.

Overview

The success of optimization techniques depends on the complexity and nature of the objective function and feasible region. This coursework focuses on a classic therapy problem and on two notoriously difficult objective functions, and investigates the most "appropriate" way of identifying their optima.

> **!** This coursework has a number of components that need to be completed. You should ensure that you allow enough time to complete them before the deadline!

Question 1

For this question, you should complete the Optimization Onramp tutorial, keeping both the certificate and the MATLAB copy of the therapy project.

Question 2

1. Write a MATLAB function that implements a line search algorithm for a function $f(x)$ using the secant method. The arguments to this function are the name of the m-file for the gradient ∇f, the current point, and the search direction. For example, the function may be called `linesearch_secant` and be used by the function call

   ```
   alpha = linesearch_secant('grad', x, d, epsilon, max_iter);
   ```

   where:

   - `grad` is the m-file or equivalent MATLAB function containing the gradient;
   - `x` is the starting line search point;

- `d` is the search direction;
- `epsilon` and `max_iter` are stopping criteria (see below);
- `alpha` is the value returned by the function (which we use in the following parts of the question as the step size for iterative algorithms).

Use the stopping criterion

$$|d^\top \nabla f(x + \alpha d)| < \epsilon |d^\top \nabla f(x)|,$$

where $\alpha > 0$ is a pre-specified number, ∇f is the gradient, x is the starting line search point, and d is the search direction. Your code should use an initial guess for $\alpha_0 = 0$ and $\alpha_1 = 0.001$ for the secant method, and perform no more than `max_iter` iterations. You can use the MATLAB function `feval` to evalute `grad` at a given location, e.g.

```
grad_at_x = feval(grad, x);
```

2. Write a MATLAB program for implementing the method of steepest descent called `steep_desc`. For this routine, you should use a fixed step length $\gamma > 0$, passed as a parameter to the routine, so that the search direction $d = -\gamma \nabla f$ within the steepest descent algorithm. For the stopping criterion, use the condition $|\nabla f(x^k)| < \epsilon$. Fix the maximum number of iterations to a parameter `max_iter`, and return the obtained solution and the number of iterations N.

   Test your code to find the minimizer of

   $$f(x_1, x_2, x_3) = (x_1 - 4)^4 + (x_2 - 3)^2 + 4(x_3 + 5)^4.$$

   Use the initial point $x = [4, 2, -1]^\top$.

3. Using your solution from the previous question as a starting point, implement a second routine called `steep_desc_secant`, which uses the `linesearch_secant` method from question 1 to determine a variable step size γ_n at each step n of the steepest descent method. Use a maximum of 100 steps and a tolerance of 10^{-6} for the line search.

   Test your program using the function f from the previous question with an initial condition of $[-4, 5, 1]^\top$, determine the number of iterations required to satisfy the stopping criterion, and evaluate the objective function at the final point to see how close it is to 0.

   Compare the output with the output from question 2.

4. Apply your program for the steepest descent algorithm to Rosenbrock's function:

   $$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2,$$

   where $x = (x_1, x_2)^\top$. Use an initial condition of $x^0 = [-2, 2]^\top$. Terminate the algorithm when $|\nabla f(x^k)| < 10^{-4}$.

Tasks

- Download the MATLAB archive which contains a template for the functions that you should complete.
- After completing your questions, submit a short report that outlines your routines, results, and lessons learnt. This should ideally be in the form of a MATLAB `.mtx` live script, as we have been using in the labs.

Debugging tips

This project involves many different bits of code.

- Deliberately I have split this into many functions. That is because this is **easier to test**!
- Ensure you use MATLAB to help debug your programs: you can set breakpoints by clicking on the line numbers, which will allow you to stop the program and observe values of variables.
- Also ensure you make use of MATLAB's routines.

Marking scheme

Your submission will be assessed on the following criteria:

- Uploading the onramp optimization certificate and solving the therapy project. **[20%]**
- Fully working implementation of question 2. **[40%]**
- Report that highlights main findings. **[25%]**
- Discretionary marks for good programming technique (in terms of efficiency, presentation of code, and structure) and appropriate use of commenting. **[15%]**

You should attempt all of the above routines; incomplete solutions will be given partial credit. Note that these routines will all be marked independently, and you can still receive full credit for complete solutions even if the preceding parts are not fully implemented.

! 'Fully working' in the above is left *deliberately* vague. You should try to consider the different inputs that your functions may receive and act accordingly. For your report, *appropriate use cases* are also vague: you should do some research to explore other possible solutions.

Submission

Bundle your report, MATLAB code, and onramp certificate into a zip file. You should submit your work via the online submission point on KEATS by the assignment deadline: **16:00, Thursday 27th July 2023**.