MIE 1621 Computational Project

Due Dec 8, 2021 by 5PM. E-mail a softcopy of your report and code(and script) to David at ryan.islip@mail.utoronto.ca

Part 1

Write a program in MATLAB or python for minimizing a multivariate function $f(x)$ using gradient-based method with backtracking. You must code your gradient method from scratch and not use any existing function for gradient methods. You need to write a brief report that summarizes your results as required below. Also, in your report you need to have a print out of your code (use good programming practice such as commenting your code.) Finally, send a soft copy of your code to the TA along with a script so that the TA can easily execute your code to see the results in your report.

(a) Use backtracking as described in class to compute step-lengths (so you need to set the parameters $s, \gamma,$ and β).

(b) Use as a stopping condition $\|\nabla f(x)\| / (1 + |f(x)|) \leq \epsilon$ with $\epsilon = 10^{-5}$ or stop if the number of iterations hits 1000.

(c) Print the initial point and for each iteration print the search direction, the step length, and the new iterate $x^{(k+1)}$. If the number of itrerations is more than 15 then printout the details of the just the first 10 iterations as well as the details of the last 5 iterations before the stopping condition is met. Indicate if the iteration maximum is reached.

(d) Test your algorithms on the following test problems

$f_1(x) = x_1^2 + x_2^2 + x_3^2$ with $x^{(0)} = (1, 1, 1)^T$

$f_2(x) = x_1^2 + 2x_2^2 - 2x_1x_2 - 2x_2$ with $x^{(0)} = (0, 0)^T$

$f_3(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$ with $x^{(0)} = (-1.2, 1)^T$

$f_4(x) = (x_1 + x_2)^4 + x_2^2$ with $x^{(0)} = (2, -2)^T$

$f_5(x) = (x_1 - 1)^2 + (x_2 - 1)^2 + c(x_1^2 + x_2^2 - 0.25)^2$ with $x^{(0)} = (1, -1)^T$

For $f_5(x)$, test the following three different settings of the parameter $c = 1, c = 10$, and $c = 100$. Comment on how larger c affects the performance of the algorithm.

(e) Are your computational results consistent with the theory of the gradient-based methods?

Part 2

In this section code in MATLAB or python various versions of Newton's method for function minimization. You must not use any existing functions that perform Newton's method and must write the code from scratch. Print the initial point and for each iteration print the search direction, the step length, and the new iterate $x^{(k+1)}$. If the number of itrerations is more than 15 then printout the details of the just the first 10 iterations as well as the details of the last 5 iterations before the stopping condition is met. Organize this information neatly and compactly using tables.

(a) Code the pure version of Newton's method (use the version from class slides) and use it to solve the problem

$$\text{minimize } 100x_1^4 + 0.01x_2^4$$

with initial point $x^{(0)} = (1,1)^T$. Use as the stopping condition $\left\|\nabla f(x^{(k+1)}\right\| \leq \varepsilon$ where $\varepsilon = 10^{-6}$. How many iterations until convergence? Were the Hessians always positive definite for each iteration?

(b) Repeat (a) but use gradient descent with backtracking with $s = 1, \gamma = 0.5,$ and $\beta = 0.5$ and initial point $x^{(0)} = (1,1)^T$. How many iterations until convergence?

(c) Use your code for the pure version of Newton's method from (a) above and use it to solve the problem

$$\text{minimize } \sqrt{x_1^2 + 1} + \sqrt{x_2^2 + 1}$$

with initial point $x^{(0)} = (1,1)^T$. Use as the stopping condition $\left\|\nabla f(x^{(k+1)}\right\| \leq \varepsilon$ where $\varepsilon = 10^{-8}$. How many iterations until convergence?

(d) Repeat (c) but use gradient descent with backtracking with $s = 1, \gamma = 0.5,$ and $\beta = 0.5$ and initial point $x^{(0)} = (1,1)^T$. How many iterations until convergence?

(e) Repeat (c) and (d) but with $x^{(0)} = (10,10)^T$.

(f) Now code Newton's method with backtracking with $s = 1, \gamma = 0.5,$ and $\beta = 0.5$ and $x^{(0)} = (10,10)^T$ and solve the problem in (c). How many iterations until convergence?

Note: You need to write a brief report that summarizes your results as required above for Parts 1 and 2. Also, in your report you need to have a print out of your code (use good programming practice such as commenting your code.) Finally, send a soft copy of your code to the TA along with a script so that the TA can easily execute your code to see the results in your report.