ECE 506: Homework #4: Line search, Trust Reg, and Conj Grad

To get help with the homework, please join me for the Saturday morning discussion sessions starting at 9am at https://unm.zoom.us/j/99977790315.

Reference:
Nocedal, Jorge, and Stephen J. Wright. Numerical optimization. New York, NY: Springer New York, 2006.

Matlab code:
Download the code from `https://github.com/pattichis/opt/blob/main/Matlab-code-for-opt.zip`.

Coding examples:
For all of your homework solutions, you must provide (1) documented source code, (2) plots, and (3) discussion. In the discussion, I sketch examples. For your solutions, I would like to see coding examples. This is always true, unless I specifically ask for a "sketch".

Problem 1. [Stepsizes and convergence rates for line search methods]
1(a) Consider the following general equation form:

$$f(x) = ax_1^2 + bx_2^2 + cx_1 + dx_2 + e.$$

For this case, compute the optimal point x^* and provide conditions that guarantee that this is an optimal point.

1(b) Compute the optimal stepsize for steepest descent line search.

1(c) For simplicity, consider the 1D case for $b = c = d = e = 0$. We know that $x^* = 0$. Compute simplified expressions for the stepsize and the magnitude of the gradient. Plot the stepsize as a 2D function of a, x_1 for $x_2 = 0$. Answer the following:

i. What kind of symmetries do you observe in your plot? For example, consider replacing a by $-a$ and/or x_1 by $-x_1$.

ii. Is the stepsize relatively large for large x_1? Explain.

iii. Is the step $||\alpha_x p_k||$ relatively large for large x_1? Explain.

iv. What happens to the stepsize as we approach the optimal point? Explain.

v. What happens to the step $||\alpha_x p_k||$ as we approach the optimal point? Explain.

1(d) Let us consider a simple case to see how the steepest descent algorithm works. Assume that:

$$f(x) = x_1^2 + 10x_2^2.$$

Suppose that $x_0 = x^* + [1, 1]^T$. Compute two steps for the steepest descent algorithm.

1(e) Verify your results in 1(c) with the provided code. You need to provide the code that you used to run the provided code.

1(f) For our quadratic case, provide the following:

i. Compute an expression for $||.||_Q$.

ii. Compute an expression in terms of the distance from x^*.

iii. Provide the best convergence case in terms of a, b.

iv. Provide the worse condition case in terms of a, b, assuming that $a > b$.

1(g) Repeat 1(d) for Newton's method. Do you need two steps? Explain using the Newton's algorithm convergence theorem.

Problem 2. Consider the following methods for which you are given Matlab codes for line-search:

1. Steepest Descent,
2. BFGS, and
3. Newton's method.

We want to consider how each method performs on finding the minima of (see `https://en.wikipedia.org/wiki/Test_functions_for_optimization` for definitions):

1. Sphere function
2. Beale function
3. Goldstein-Price function
4. Booth function

You will need to assess:

Robustness: Run the program with random initial guesses (that still satisfy any constraints), to assess the performance. Show results from at-least 3 random initial points.

Efficiency: Compute the required memory and function evaluations for each iteration. For memory requirements, express your results in terms of n, the dimensionality of the problem. Thus, a gradient requires that we store n floating-point values. You will need to modify the code to keep track. In terms of function evaluations, **report function, gradient, and hessian evaluations separately**. Thus, you may need to produce a total of four plots. **When computing requirements, you should not include any extra requirements associated with storing the path for visualization purposes. In other words, you need to focus on essential requirements only.**

Accuracy: For each run, **establish** the best rate of convergence. **Discuss** whether the theoretical rate of convergence is accomplished. For the rate of convergence, apply both methods provided in the hints. Do the two methods agree? Explain.

Problem 3. You are provided with code for solving problem 4.3 using Conjugate Gradient Steihaug. Repeat problem 2 for this case.

Hints:

1. Symbolic Gradients and Hessians (optional).

In Matlab, you can evaluate the gradient and the Hessian by simply using:

```
syms x y z
f = x*y + 2*z*x;
hessian(f,[x,y,z])
gradient(f)

ans =
[ 0, 1, 2]
[ 1, 0, 0]
[ 2, 0, 0]

ans =
y + 2*z
x
2*x
```

2. **Rates of convergence.**

To establish the rate of convergence, you will need to consider Q-linear, Q-superlinear, and Q-quadratic convergence. If we know the optimal point x^*, then we can establish convergence based on the plots of:

$$\frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|}, \quad \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^2} \quad \text{versus the iteration number } k.$$

Unfortunately, when developing our algorithms, we do not know x^*. Instead, we do know that at the optimal points, we also get that:

$$||\nabla f_k(x^*)|| \to 0 \quad \text{as} \quad k \to \infty.$$

Thus, without knowning x^*, we can look at:

$$\frac{\|\nabla f_{k+1}\|}{\|\nabla f_k\|}, \quad \frac{\|\nabla f_{k+1}\|}{\|\nabla f_k\|^2} \quad \text{versus the iteration number } k.$$

This is the method to follow to establish the rate of convergence.

To establish the rate of convergence, note that:

$$\text{Q-quadratic conv.} \implies \text{Q-superlinear conv.} \implies \text{Q-linear conv.}$$

Thus, you only need to establish the highest order of convergence. Furthermore, for problem 6, we consider showing that:

$$\liminf_{k\to\infty} \|\nabla F(w_k)\| = 0.$$