MACM 316 - Computing Assignment 3

- Read the *Guidelines for Assignments* first.
- Submit a one-page PDF report to Canvas and upload your Matlab scripts (as m-files). Do not use any other file formats.
- Keep in mind that Canvas discussions are open forums.
- You must acknowledge any collaborations/assistance from colleagues, TAs, instructors etc.

Numerical optimization

An important problem in numerical computing is finding the minimum x^* of a function $f(x)$. This is very much related to the root-finding problem. Indeed, if f is differentiable, then a minimum $x = x^*$ of $f(x)$ is a zero of the derivative $f'(x)$.

Unfortunately, an approach based on applying the bisection method to $f'(x)$ may not work, since the derivative values may not be available in practice. Fortunately, there is an algorithm similar to the bisection method can be used to find the minimum x^* using values of $f(x)$ only.

Recall that the bisection method produces pairs of numbers $[a_n, b_n]$. For finding minima, we will instead produce a sequence of triples $[a_n, b_n, c_n]$ that have the following bracketing property

$$f(a_n) > f(b_n) \quad \text{and} \quad f(b_n) < f(c_n).$$

Hence b_n can be used as an approximation to the minimum x^* at step n. To compute such triples, the algorithm proceeds in the following way:

1. Choose a new point x using the formula:

$$x = \begin{cases} b_n + \lambda(c_n - b_n) & \text{if } (c_n - b_n) > (b_n - a_n) \\ b_n + \lambda(a_n - b_n) & \text{if } (c_n - b_n) < (b_n - a_n) \end{cases}$$

2. Update the triple using the formula:

$$[a_{n+1}, b_{n+1}, c_{n+1}] = \begin{cases} [a_n, x, b_n] & \text{if } x < b_n \text{ and } f(x) < f(b_n) \\ [b_n, x, c_n] & \text{if } x > b_n \text{ and } f(x) < f(b_n) \\ [x, b_n, c_n] & \text{if } x < b_n \text{ and } f(x) > f(b_n) \\ [a_n, b_n, x] & \text{if } x > b_n \text{ and } f(x) > f(b_n) \end{cases}$$

You may wish to verify that the update formula guarantees the bracketing property holds at each step. Note that the choice of λ will affect the convergence rate. It can be proven that $\lambda = (3-\sqrt{5})/2$ is the optimal choice. Use this value throughout.

Implement the algorithm to find the minimum of the function

$$f(x) = -\cos(x^k)$$

for $k = 1, 2, ..., 5$. Use 100 iterations of the algorithm and the initial triple $[a_0, b_0, c_0] = [-1, 1/2, 1]$. Plot error versus iteration number for each k.

Discuss the accuracy of the algorithm. How would you characterize the error as a function of the iteration number? What effect, if any, does k have on the accuracy?

How does the robustness of the algorithm depend on k? Explain your observations. Hint: expand $f(x)$ in a Taylor series about x^*, noting that $f'(x^*) = 0$. (Robustness refers to an algorithm's ability to resist round-off error.)

Coding Tips

- None of the comparisons of the algorithm account for equality. (Think about why this is the case.)
- Be very careful in the update step of the algorithm. Make sure not to overwrite data you'll need at the next iteration.
- You should only need one graph for this report. Each data group should be easily identifiable and properly labelled according to its value of k.