MACM 316 – Computing Assignment 4

- Read the *Guidelines for Assignments* first.
- Write a one-page PDF report summarizing your finding. This report must also include all your figures.
- Submit the one-page PDF report using Crowdmark and add your Matlab code to it (as extra pages). You must use the link sent by Crowdmark to your SFU email address. Follow the instructions on the Crowdmark to upload your file. If the report is hand-written then use the CamScanner app with your cellphone to scan the report and every page of the code. You will lose marks for poor quality pictures.
- You must acknowledge any collaborations/assistance from fellow students, discussion forums, TAs, instructors, etc.

Numerical optimization

An important problem in numerical computing is finding a 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 point at which the derivative $f'(x) = 0$.

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

$$f(a_n) > f(b_n) \quad \text{and} \quad f(b_n) < f(c_n). \tag{1}$$

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:
$$x = \begin{cases} b_n + \gamma(c_n - b_n) & \text{if } (c_n - b_n) > (b_n - a_n) \\ b_n + \gamma(a_n - b_n) & \text{if } (c_n - b_n) < (b_n - a_n) \end{cases}$$
2. Update the triple using the formula:
$$[a_{n+1}, b_{n+1}, c_{n+1}] = \begin{cases} [a_n, x, b_n] & \text{if } x < b_n \text{ and } f(x) < f(b_n) \\ [b_n, x, c_n] & \text{if } x > b_n \text{ and } f(x) < f(b_n) \\ [x, b_n, c_n] & \text{if } x < b_n \text{ and } f(x) > f(b_n) \\ [a_n, b_n, x] & \text{if } x > b_n \text{ and } f(x) > f(b_n) \end{cases} \tag{2}$$

You may wish to verify that the update formula in (2) guarantees the bracketing property (1) holds at each step of the algorithm. Note that the choice of γ in (1) will affect the convergence rate. It is possible to prove that the optimal choice is $\gamma = \frac{3-\sqrt{5}}{2}$. You should use this value throughout.

Your task in this assignment is to study this algorithm. First, write a script to implement the algorithm. Once you have done this, run it for $N = 100$ iterations on the function

$$f(x) = -\frac{\cos(x)}{1 + x^2},$$

using the initial values $[a_0, b_0, c_0] = [-1, 1/2, 1]$. Plot the error versus iteration number and comment on the observed accuracy, efficiency and robustness of the algorithm.

Next, replace $f(x)$ by the function

$$f(x) = -\frac{\cos(x^k)}{1 + x^{2k}},$$

where $k \geq 1$ is an integer. Repeat the above experiment for several different values of k and comment on the accuracy, efficiency and robustness of the algorithm once more. How do these change as k varies?

Finally, seek to explain the observed robustness (or lack thereof) for the different values of k. To do this, consider the Taylor series expansion of f around x^*:

$$f(x) = f(x^*) + (x - x^*)f'(x^*) + \frac{(x - x^*)^2}{2} f''(x^*) + \frac{(x - x^*)^3}{3!} f'''(x^*) + \ldots.$$

How small close can x get to x^* before issues due to floating point arise?