Optimitzation
Modelling for Science and Engineering, 2022-2023

1 Minimization problem without constraints

1.1 Objective

Let's to consider the following optimization problem without constraints

$$min\{f(x) : x \in \mathbb{R}^n\}$$

The needed optimal condition, $f'(x) = 0$, can be the basis of an application of Newton's method. The minimization problem is equivalent to finding the zeros of a system of non-linear equations

$$F(x) = 0, \quad \text{where} \quad F : \mathbb{R}^n \to \mathbb{R}^n,$$

$F = f'$ and $F' = f''$ (the Hessian of f).

1.2 The Newton's method (univariate root finding)

(Let's start with the simplest case when $n = 1$).

We start with an initial point, x_0, and we compute the sequence

$$x_{k+1} = x_k - \frac{F(x_k)}{F'(x_k)}, \quad \text{with} \quad k \geq 0 \tag{1}$$

Let's suppose that $F = f'$ has continuous derivatives with $F'(\bar{x}) = f'' \neq 0$ (we can see that $\bar{x}$ is a simple zero of F).

If x_0 is closely enough of $\bar{x}$, the series (1) quadratically converges to $\bar{x}$, or in other words

$$\lim_{k\to\infty} \frac{|x_k - \bar{x}|}{|x_{k-1} - \bar{x}|^2} = C, \quad \text{where} \quad C = \frac{|F''(\bar{x})|}{2|F'(\bar{x})|},$$

Therefore, when k is large enough

$$|x_k - \bar{x}| \approx C\,|x_{k-1} - \bar{x}|^2.$$

We will use the difference between a point and the previous one ($|x_{k+1} - x_k|$) as an estimation of the error ($|x_k - \bar{x}|$) because we don't know it a priori. Then, given x_0, we will obtain the sequence x_k and we will stop the iteration when

$$|x_{k+1} - x_k| < tol_1 \qquad \text{or} \qquad |F(x_k)| = |f'(x_k)| < tol_2 \tag{2}$$

where tol_1 and tol_2 are the fixed tolerances.

You need to program Newton's method in a `function` named `Newton`, where we will use as inputs:

x0	=	initial point
MAXit	=	maximum number of iterations
tol1, tol2	=	tolerances like (2) that will allow to finish the iterations

and as outputs

sol	=	calculated solution
h	=	difference between the two most recent calculated points before ending
Fsol	=	value of F in the calculated solution
NTiter	=	total number of iterations

(Note that you need to define a vector `v` that contains the iterative solutions of the series computed by Newton's method defined in (1)).
The function F and its derivative F' will be evaluated in a `function` named `CALCfun`.

Finally, you can note that being $\bar{x} = min\{f(x) : x \in \mathbb{R}^n\}$, we have $F(\bar{x}) = f'(\bar{x}) = 0$, and we can find $\bar{x}$ with the Newton's method.

One issue with Newton's method is that in order to prevent *overflow*, we must choose an initial point that is closer to the solution. By giving an interval $[a, b]$ that contains the zero of F rather than the initial point x_0, you can implement Newton's technique in a way that is safer and avoids this issue. Do a function named NewtonM that does it. Then, when a potential iteration solution is outside of the interval, the method must return an error. When we don't know $F'(x)$ (for example, it could be very complicated and tedious to derive by hand), then we can instead approximate it using some numerical method. We can for example use the backward approximation of a derivative, given by Eq. (4) below

$$F'(x_k) \approx \frac{F(x_k) - F(x_{k-1})}{x_k - x_{k-1}}, \quad \text{with} \quad k \geq 0 \tag{3}$$

By combining Eq. (1) and Eq. (4) we obtain the secant method where

$$x_{k+1} = \frac{x_{k-1}F(x_k) - x_k F(x_{k-1})}{F(x_k) - F(x_{k-1})}, \quad \text{with} \quad k \geq 0 \tag{4}$$

In this work you need:

1. To program a basic Newton method for univariate root finding and show an example. You need to plot a function with at least a zero, run the algorithm, and provide the number of iterations that you needed to achieve the solution, the plot of $|x_{k+1} - x_k|$ for all k, and the solution. You need to provide it for an x0 close to the solution and an x0 far away and argue what happens in both cases in terms of convergence.
2. To program a basic Newton method to find the minimum of a 3D function. Again, test it with an example, i.e., provide a 3D function that has a minimum. You can plot it using:
   - Exemple>> `ezmeshc('-1/exp(x^2+0.2*y^2)',[-2,2])`

   And verify the minimum. Similarly to before, test your method and argue what happens if your [x0, y0] is far or close to the minimum, respectively. Provide in both cases the number of iterations needed to achieve the solution, the plot of the error that you chose for the stop criteria, and the solution.

You need to provide the code and a report arguing the advantages and drawbacks of the Newton method. You need also to argue how is the convergence of the Newton method.