Dragos Pop (2618006)

Assignment 2 – Secant method

Concise method description

The function implemented in Matlab determines the zero of a given function (as long as this exists) using the Secant method. In other words, it iteratively draws a straight line through the previous two points on the graph and takes the zero of that line, starting from the initial guesses. Moreover, the method returns the vector of approximate values obtained in the iterations (including the two initial guesses) if the user demands two output variables.

Function syntax

Given the required name of the Matlab file, the function can be called by *Assignment_2_Pop()*, specifying the right arguments. These are:

- *func* = the function for which the user would like to determine the root
- *x0* and *x1* = the initial guesses
- *tol (optional)* = the desired accuracy of the result (it stops the program if the change in x is smaller, thus, the smaller the value, the more precise the result). In case *tol* is not specified by the user in the input, it takes default value 0.0001
- *nmax (optional)* = the maximum number of iterations the program tries to find a root until it stops. In case *nmax* is not specified by the user in the input, it takes default value 1000

Accordingly, calling *Assignment_2_Pop(@(x) 2*x^2-5*x+2, 1, 1.1, 0.0001, 1000)* is the same as calling *Assignment_2_ Pop(@(x) 2*x^2-5*x+2, 1, 1.1),* since tol and nmax are optional, and returns 0.5 as the root of the function and the following approximate values obtained in the iterations [1.0000 1.1000 -0.2500 0.7727 0.6034 0.4749 0.5018 0.5000].

Restrictions

The current implementation of the program does not allow the user to input *nmax,* but leave *tol* aside, and whenever this happens, it performs erroneously. However, the opposite works.

Another restriction of the program is that it is not always able to find a zero due to the Secant method limitations. For instance, when the line formed by the initial guesses does not go through the horizontal axis, the method is not able to find a root and the program returns a NaN. Also, if the given function does not have a root at all, the program will continue to search for one until the iterations budget is finished.

Program design

In the beginning, the parameters *nmax* and *tol* are set to optional, followed by the evaluation of the function at the two initial guesses. Next, the program iteratively evaluates the function at the second point, computes the change in x (the second term in the secant method), which leads to the following x, transforms the second point in the first point of the next iteration and adds it to the vector of approximate values. The loop stops once the change in x is smaller than *tol*, the desired precision of the result, or when the maximum number of iterations is reached. If the second condition is the one responsible, the program outputs a message stating it could not converge to the desired tolerance due to *nmax*. Finally, the root of the function is returned as the second point in the last iteration.

Figure interpretation

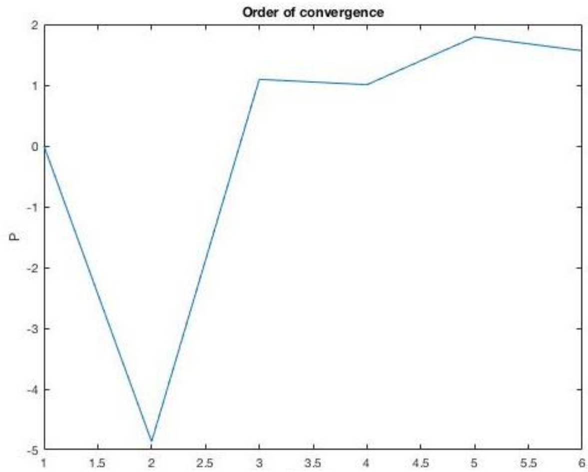

Figure 1 – Order of convergence of Assignment_2_Pop(@sin, 1, 1.1, 0.000001, 10)

The illustration on the right represents the order of convergence of *Assignment_2_Pop(@sin, 1, 1.1, 0.000001, 10)*. This was computed in the following way, calling *convergence(@sin, 1, 1.1, 0.000001, 10*): for every iteration, a value p was computed as $p \approx \frac{\ln(\frac{x_{n+1}-\alpha}{x_n-\alpha})}{\ln(\frac{x_n-\alpha}{x_{n-1}-\alpha})}$. (as expressed in Berg J. B. & Planqué B., *Lecture Notes on Numerical Mathematics*), where alpha is the root of the function *sin* and x_n is the value of x at iteration n. Then, the values of p were plotted against the iteration when they were calculated. Accordingly, one can see from Figure 1 that in this example the order of convergence is approaching 1.61 ('Golden Ratio'), which is the actual theoretical convergence of the Secant method. On the other hand, the order of convergence of Newton's method is 2, which means that on average, Newton's method is converging faster than the Secant method.

Comparison with Newton's method

According to the previous section, one disadvantage of the Secant method is that it is converging slower than Newton's method because its order of convergence is smaller. Also, it requires two initial guesses which can be perceived as an extra effort of the user, hence, a drawback of the method.

When it comes to its advantages, the Secant method does not need the derivative of the function, which may be difficult to compute in some cases.