

Numerical Methods in Physics and Astrophysics

Problem Set 1; Problem 3

For problem 3 we are given a system of two equations as follows:

$$\begin{aligned}f(x, y) &= xy - 0.1, \\g(x, y) &= x^2 + 3y^2 - 2.\end{aligned}$$

1. In order to solve this system of equations numerically we will employ the generalized Newton's method for a system of two equations:

$$\begin{aligned}x_{n+1} &= x_n - \left(\frac{f \cdot g_y - g \cdot f_y}{f_x \cdot g_y - g_x \cdot f_y} \right)_n \\y_{n+1} &= y_n - \left(\frac{g \cdot f_x - f \cdot g_x}{f_x \cdot g_y - g_x \cdot f_y} \right)_n\end{aligned}$$

Calculating the needed partial derivatives yields:

$$\begin{aligned}f_x &= y \\f_y &= x \\g_x &= 2x \\g_y &= 6y\end{aligned}$$

These substitutions were then defined as separate equations in code and the general Newton Method was implemented.

Looking at a graph of the equations, we know that there are four roots.

Four different starting positions were thus chosen:

$$\begin{aligned}x_{0,1} &= (-1, 0) \\x_{0,2} &= (0, 1) \\x_{0,3} &= (1, 0) \\x_{0,4} &= (0, -1)\end{aligned}$$

This in turn yielded four roots with less than six iterations needed to find each root with an error of less than 10^{-6} .

$$r_1 = (-1.40885974, -0.07097939)$$

$$r_2 = (0.12293990, 0.81340555)$$

$$r_3 = (1.40885974, 0.07097939)$$

$$r_4 = (-0.12293990, -0.81340555)$$

Given the careful input and debugging, the program very quickly found the roots. Some conditions that may impeded the program's ability to find the root include picking poor starting locations, not picking enough starting locations, and picking a location that may cause one of the functions to diverge.

2. Our initial equations have the form

$$\begin{aligned} f(x, y) &= xy - 0.1 = 0, \\ g(x, y) &= x^2 + 3y^2 - 2 = 0. \end{aligned}$$

In order to implement an iterative method of finding the roots, we will first rearrange the equations to find y in terms of x :

$$f \Rightarrow y = \frac{0.1}{x}$$

Substituting this into $g(x, y)$ yields

$$\begin{aligned} x^2 + 3\left(\frac{0.1}{x}\right)^2 - 2 &= 0 \\ \Rightarrow x^2 + \frac{0.03}{x^2} &= 2 \end{aligned}$$

which can then finally be reformed to

$$\begin{aligned} x &= \sqrt{2 - \frac{0.03}{x^2}}, \\ \Rightarrow x_{n+1} &= \sqrt{2 - \frac{0.03}{x_n^2}} \end{aligned}$$

yielding a formula that can be used iteratively. Similarly, one can derive an iterative formula for y .

$$\begin{aligned}
f &\Rightarrow x = \frac{0.1}{y} \\
g &\Rightarrow \left(\frac{0.1}{y}\right)^2 + 3y^2 - 2 = 0 \\
&\Rightarrow 3y^2 = 2 - \frac{0.01}{y^2} \\
&\Rightarrow y = \sqrt{\frac{2 - \frac{0.01}{y^2}}{3}} \\
&\Rightarrow y_{n+1} = \sqrt{\frac{2 - \frac{0.01}{y_n^2}}{3}}
\end{aligned}$$

After implementing this iterative approach in software and choosing the following initial values

$$\begin{aligned}
x_{0,1} &= (2, 1) \\
x_{0,2} &= (-2, -1) \\
x_{0,3} &= (1, 2) \\
x_{0,4} &= (-1, -2),
\end{aligned}$$

one sees that all four starting positions converge on the same values of

$$r = (1.40885975, 0.81340555)$$

after four iterations. The relative performance of this method is about the same as for the calculation of the roots using the generalized Newton Method above (which took four to five iterations for the same error size).

Due to the implementation of the iterative method, all of the calculated answers will be positive. If one consider both the positive and negative roots, then one could more correctly obtain all of the roots for the problem, though the final calculated values will only depend on the magnitudes of the initial values since they are squared in the iterative steps.