Minimisation

Gavin Cheung ★ 09328173

February 9, 2011

Abstract

Using C, the roots of a function could be found with the bisection and Newton-Raphson method. The two methods were then compared and it was found that the Newton-Raphson method was much more efficient. This method was then used to find the minimum of a potential function describing the interaction between a sodium and chlorine atom. This minimum corresponds to the bond length and was found to be 0.236054 nm.

⋆ Introduction

A lot of problems in physics involves minimising or maximising functions. Given some differentiable function $f(x)$, to find the max/min, it is required to find the root of $f'(x)$. While this sounds simple for some function, $f(x) = x^2 + 2x + 1$, say, trying to find the minimum of $f(x) = -\frac{1}{r} + e^{-r}$ is impractical without a computer. In fact, this function describes bonding between atoms which makes finding the stable point a very important topic.

Firstly, the root of a quadratic function will be found using the bisection method followed by the Newton-Raphson method such that a comparison can be made between the two methods. In particular, the function is $f(x) = x^2 + 2x - 12$. Analytically, the roots are $x = -1 \pm \sqrt{13}$.

The Newton-Raphson method is then used to find the minimum of the function $V(x) = -\frac{1.44}{x} + 1090e^{-0.033x}$. This is done by finding the root of $V'(x)$ and verifying that $V''(x) > 0$. If $V''(x) < 0$, then the root is a maximum. This function is the potential between a sodium and chlorine atom and the minimum corresponds to the bond length where the state is stable.

⋆ Experimental Method

Bisection Method

The first algorithm used to find the root is called the bisection method which is an application of the intermediate value theorem. Given x_1, x_2; if $f(x_1) < 0$ and $f(x_2) > 0$, then there exists some $z \in (x_1, x_2)$ such that $f(z) = 0$. Taking the average value $x_3 = \frac{x_1 + x_2}{2}$ and checking whether it is negative or positive, x_3 then replaces x_1 or x_2 respectively. This narrows down the lower and upper bound of the root. The process is continued until $f(x_3)$ is sufficiently close to 0. This is implemented in the code by ending the loop when $f(x_3) < c$ where c is the tolerance.

The number of steps taken to reach this value is also recorded. Using Gnuplot, a graph of the function could be plotted to give an idea of suitable values for x_1 and x_2 to take (see Appendix). For convenience, only the positive root will be considered but the other root can easily be found as well by choosing a suitable interval.

Newton-Raphson Method

The second algorithm used is the Newton-Raphson method. Taking the Taylor Expansion of $f(x)$ at z,

$$f(x) = f(z) + (x - z)f'(x) + \dots$$

Since $f(z) = 0$,

$$f(x) \approx (x - z)f'(x)$$

$$z \approx x - \frac{f(x)}{f'(x)}$$

provided the point x is close to z such that we can neglect the higher order terms. Writing this as an iteration,

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Unlike the first method, this method only requires one initial value, x_1. With each iteration, a value closer to the root is given. Once again, Gnuplot is used to pick a suitable value for x_1 and the program is run until $f(x_n)$ is sufficiently close to 0. Due to the second term, it is necessary to put a simple if-else statement that will terminate the program if $f'(x_i) = 0$ to prevent division by zero.

⋆ Results and Analysis

1) $x^2 + 2x - 12$

For the bisection method, x_1 is chosen to be 0 and x_2 is chosen to be 4. For the Newton-Raphson method, x_1 is chosen to be 0. The following table shows the results for different values of accuracy.

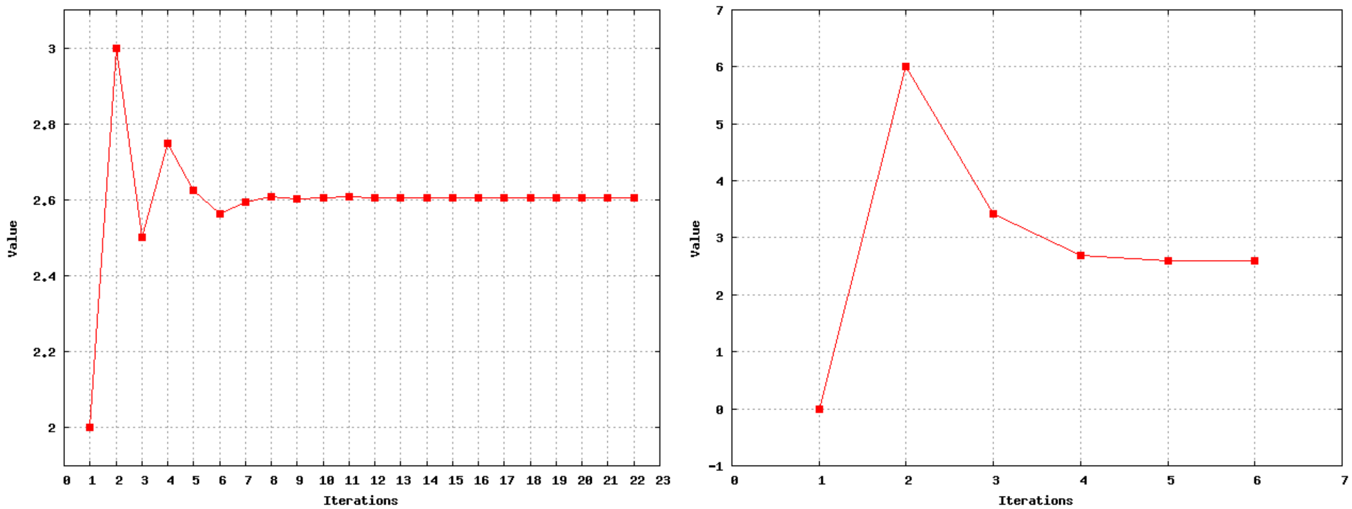
Tolerance	Steps	Root
0.1	8	2.6
0.01	11	2.61
0.001	11	2.606
0.0001	18	2.6056
0.00001	22	2.60555

Bisection

Tolerance	Steps	Root
0.1	5	2.6
0.01	5	2.61
0.001	6	2.606
0.0001	6	2.6056
0.00001	6	2.60555

Newton-Raphson

While it is obvious that with improving accuracy, the number of steps increases, the Newton-Raphson method is clearly much more efficient than the bisection method. The explanation for this comes from the geometry of how the Newton-Raphson method works. The slope converges quickly towards the root whereas the bisection method is completely random. One can say that the bisection method blindly looks for a root whereas the Newton-Raphson method sniffs out the root. The two following graphs show how the different methods approach the root with each iteration for the same tolerance.



(a) Bisection (b) Newton-Raphson

2) $V(x) = -\frac{1.44}{x} + 1090e^{-0.033x}$

Using Gnuplot and a bit of trial and error, the initial value was set as 0.2. The tolerance is set as 0.000001 and in 6 steps, the minimum was found to be 0.236054. For other initial values, $x_1 = 0.1$ gives the answer in 10 steps and $x_1 = 0.3$ fails to give a minimum. This shows one of the disadvantages of the Newton-Raphson method since it is dependent on the choice of the initial value. What is happening in this case is that the value is approaching the maximum of the function at $x \to \infty$ instead of the minimum.

$\star$ Conclusion

To conclude, the Newton-Raphson method is clearly more efficient than the bisection method. However, both come with advantages and disadvantages. The bisection method is simpler and only requires a suitable interval and it will find the root but the disadvantage is that it is slow. The Newton-Raphson method is a lot faster than the bisection method but it requires $f'(x)$ to exist and be nonzero at the points and the success depends on a correct choice of the initial value which was seen during this experiment.

The bond length of sodium chloride was then found to be 0.236054 nm. This would've been difficult to find by hand showing how powerful computational physics can be.

⋆ Appendix

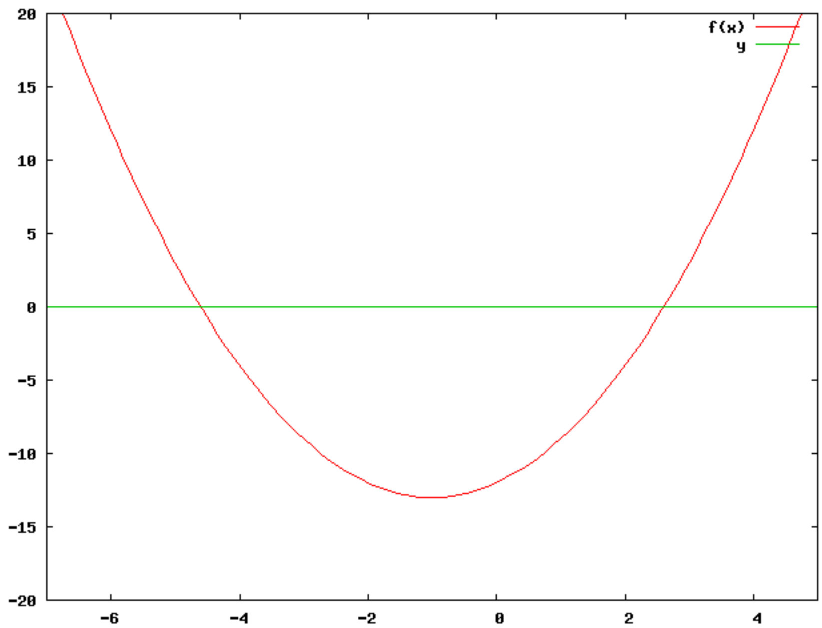


Figure 1: Graph of $f(x) = x^2 + 2x - 12$

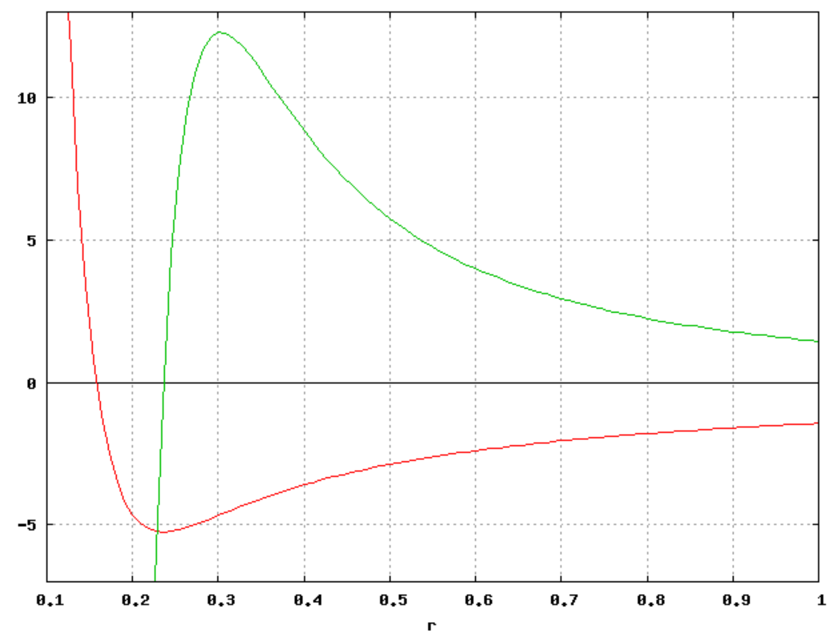


Figure 2: Graph of $V(x) = -\frac{1.44}{x} + 1090e^{-0.033x}$ in red with $V'(x)$ in green