Numerical Solution To Equation

Oves Badami

May 2023

1 Semiconductor Concepts

Figuring out the conduction band edge with respect to the Fermi level Application for Monte Carlo Algorithms

2 Numerical Solutions

2.1 Introduction

what is iteration ? Iteration is the process of repeating the same calculations with "better" guesses so as to converge to the results. Note that repeating the calculation may or not always lead to correct solution. When the results converges to a correct value, we say that the solution has converged else the solution has diverged. How do we decide the convergence has reached, the answer is simple, We can calculate the approximate relative error as

$$Error = |\frac{x_n - x_{n-1}}{x_n}| \tag{1}$$

when the error becomes lesser than the tolerance criterion then we say that convergence has reached. Note that we are using the absolute value because we are interested in how far we are away from the actual root **But just because error is below the tolerance doesn't necessarily mean that the results are correct. We must also do a sanity check to ensure that the results make sense. For example when we try to calculate the occupation probabilities it is possible that roots would converge but would result in negative values of the occupation probabilities. Another case is when we calculate carrier concentration for electrons or holes. The solver might say that the update is smaller than the tolerance value but may have negative carrier concentrations. Moral of the story is dont be quick in declaring victory**

2.2 Bounded methods

These methods work on the principle of getting two initial guesses such that actual root lies within these two solutions.

2.2.1 Bisection Method

Bisection method is perhaps the simplest of all the methods. This method is based on intermediate value theorem which states that if f, a continuous function, has opposite signs at some x = a and x = b then f must be zero at some intermediate values. As the name suggests, we first get a range within which there is a solution and then we keep on bisecting (dividing) the range into 1/2 until we hit the root (this is called as iteration).

Geometrical interpretation of bisection method/ Intermediate Value theorem.

Algorithm

1. Fix values of a and b such that f(a)<0 and f(b)>0
2. Calculate c such that $c = 0.5(a + b)$
3. Reset the range
   (a) if $f(c) < 0$ a = c
   (b) else if $f(c) > 0$ b = c
   (c) else f(c) = 0 converged
4. Check for the convergence ?
   (a) if yes: End the simulation
   (b) if no: Go to step 2

The best point is that this method always converges i.e. once we have the initial range then the root of the equation is guaranteed. But the convergence is slow. To determine the convergence we can The true error at the n^{th} iteration is defined as

$$\varepsilon_T = |x_T - x_n| \tag{2}$$

The approximate error is defined as

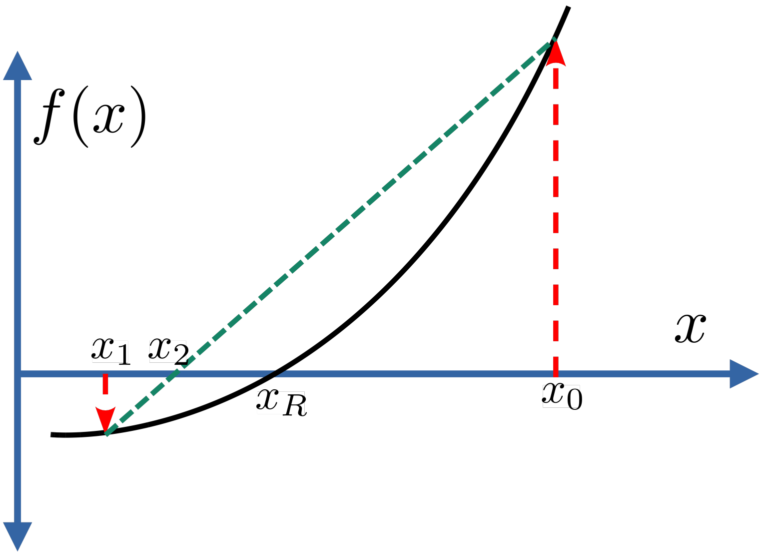
$$\varepsilon_A = |x_{n+1} - x_n| \tag{3}$$

However, we cannot always get the true error because it implies that we know the true root. Hence we have to use the approximate error form. **And interestingly it can be justified that for the bisection method the true error is always lower than the approximate error.**

Can you come up with another error criterion that is more robust and reliable?

2.2.2 Regula Falsi Method

It is important to realize an important fact that in the bisection method we haven't considered the slope in the root calculation. Now it is natural to realize that if the root lies between two values (x_0, x_1) (we know this because $f(x_0)f(x_1) < 0$) the we may be able to get a better approximation to the root by employing the slope. To see this consider the figure as shown. Considering the similar triangles we can write the following equation



$$\frac{f(x_0)}{x_2 - x_0} = \frac{f(x_1)}{x_2 - x_1} \tag{4}$$

$$x_2 = x_1 - f(x_1)\frac{x_1 - x_2}{f(x_1) - f(x_2)} \tag{5}$$

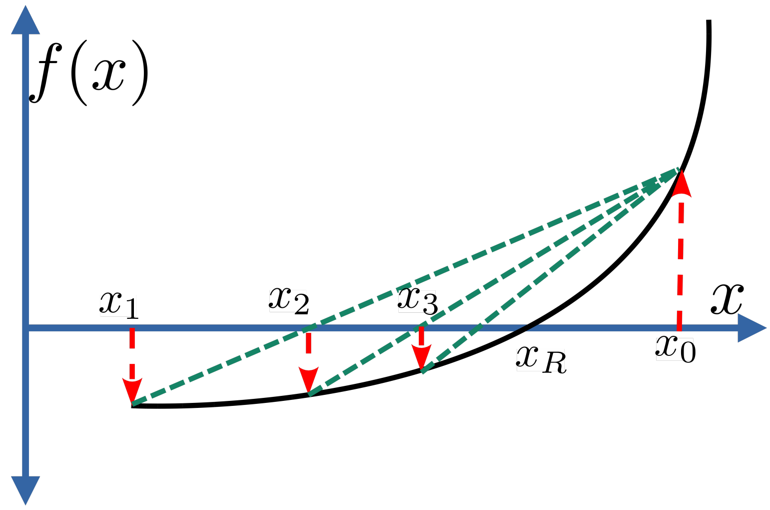
Thus we can continue in this direction with each iteration successively improving the approximate root values. This method since it employees additional information of the slope generally gives results faster than the bisection method.

Algorithm

1. Fix the values of a and b such that f(a)¡0 and f(b)¿0
2. Calculate the new root using the similarity of triangles
3. Reset the range
   (a) if $f(c) < 0$ a = c
   (b) else if $f(c) > 0$ b = c
   (c) else f(c) = 0 converged
4. Check for the convergence ?
   (a) if yes: End the simulation
   (b) if no: Go to step 2

If you are feeling that use of slope may also be a sort of problem under certain circumstances then yes you are right. Look that the figure shown. In this case it is easy to see that bisection method would certainly converge faster than the Regula Falsi method. This is shown by a fact that one of the bracketing points remains fixed and this leads to the poor convergence. **Can you come up with a better algorithm**.

Can you see what are we trying to do under the fancy name of regula falsi – we are linearly interpolating the function (i.e. we are assuming that the function $f(x)$ is linear.)



2.3 Open Methods

2.4 Fixed Point method

In this method, we write the equation ($f(x) = 0$) that has to be solved as $x = g(x)$. Here we guess a value of x (x_0) and then calculate $g(x_0)$ and the calculate the value x (x_1) As you can realise the subscript is the iteration index. So for the first iteration we have

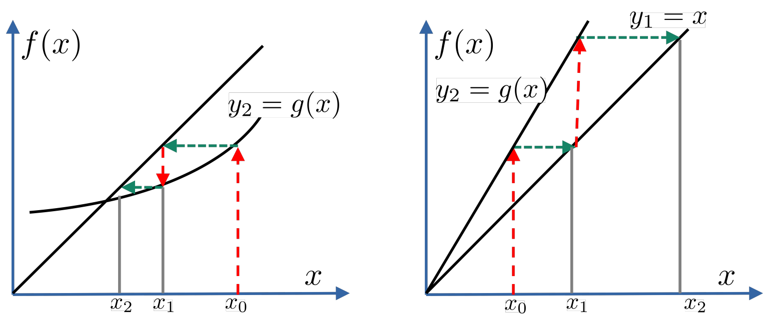
$$x_1 = g(x_0) \tag{6}$$

and in general we can write the solution at the n^{th} iteration as

$$x_n = g(x_{n-1}) \tag{7}$$

We continue this iteration until the convergence is reached ($|x_n - x_{n-1} < \delta|$)

Discussion on Convergence of the fixed point method Another way to look at the fixed point method is to recall that if we can find the root by plotting $y_1 = x$ and $y_2 = g(x)$ on the same graph and their intersection is the root of the equation $x = g(x)$. Two examples are shown in the above figure. Here the iteration schemes



are also marked. It can be seen that in one case the root converges while in another case the calculate root diverges from the actual value. Thus this shows that convergence is not always guaranteed unlike the bracketing methods. So what exactly causes the convergence. It is relatively easy to see that the convergence essentially has to do something with the slope. At this point it important to specify that the slope of the $y_1 = x$ is 1 ($m_{y_1} = 1$). To prove this point (assume that x_R is the real root of the equation)

$$x_{i+1} = g(x_i) \tag{8}$$

$$x_R = g(x_R) \tag{9}$$

$$x_{i+1} - x_R = g(x_i) - g(x_R) \tag{10}$$

Derivative mean value theorem states that if a function g(x) and g'(x) are continuous over the interval $a \geq x \leq b$ then there exists atleast one value ζ in the interval such that

$$g'(\zeta) = \frac{g(b) - g(a)}{b - a} \tag{11}$$

Using the above equation we can write the

$$x_{i+1} - x_R = g'(\zeta)(x_i - x_R) \tag{12}$$

$$E_{T,i+1} = g'(\zeta)E_{T,i} \tag{13}$$

where $E_{T,j}$ is the true error at j^{th} iteration clearly for the convergence $E_{T,i+1} < E_{T,i}$ which implies that the slope in the interval (x_i, x_R) should be lesser than 1.

2.4.1 Newton Raphson Method

Till now we have not really used an important

If we assume that x_R is the real root of the equation and x_A is the approximate root which sufficiently close to x_R we can expand the function f(x) using the Taylor's series in the vicinity of x_A

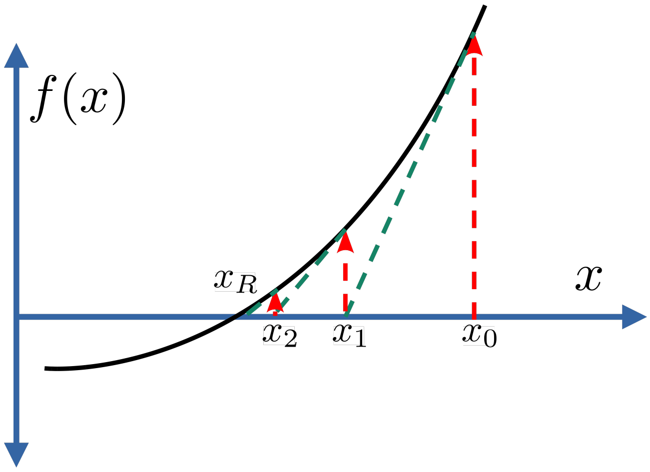
$$f(x_R) = f(x_A) + \frac{f'(x_A)}{\Delta_x} + \cdots \tag{14}$$

Since $f(x_R) = 0$, we can rewrite the above equation as

$$\Delta_X = \frac{f(x_A)}{f'(x_A)} = x_R - x_A \tag{15}$$

$$x_R = x_A - \frac{f(x_A)}{f'(x_A)} \tag{16}$$

where $\Delta_X = x_R - X_A$ Since we are dealing with numerical solution I can say that x_R is better guess than x_A. In such a case we can write the above equation as



$$x_n = x_{n-1} - \frac{f(x_n - 1)}{f'(x_{n-1})} = x_{n-1} - (f'(x_{n-1}))^{-1} f(x_{n-1}) \tag{17}$$

Note that in general the Newton-Raphson equation gives the change/update in the solution rather than the actual solution itself. So we have add the update to the previous solution

Algorithm

1. Guess initial root ($x_{n-1} = x_0$)
2. Evaluate f(x) and f'(x) at the approximate root
3. Calculate the update $\frac{f(x_{n-1})}{f(x_{n-1})}$
4. Calculate the new root $x_n = x_{n-1} - \frac{f(x_{n-1})}{f(x_{n-1})}$
5. Check for the convergence ($\Delta_x < \delta$)
   (a) if yes: Terminate the run
   (b) if no: Goto step 2

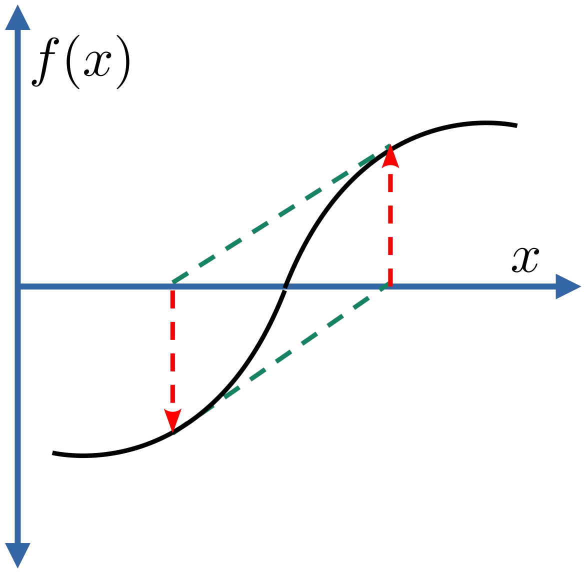
Pitfalls: It must be noted that the convergence of the Newton-Raphson method is strongly dependent on the initial guess. Due to initial guess it may happen that the solution would oscillate and thereby . An illustration of this is shown below.

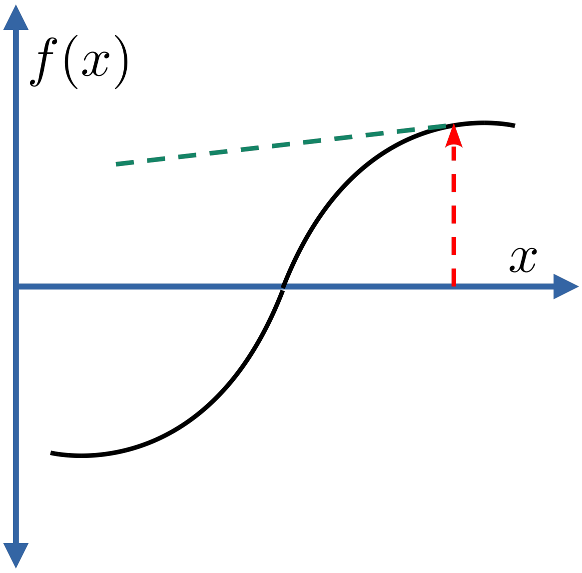
$$x_n = x_{n-1} + \omega \Delta_x \tag{18}$$

$$= \omega x_n + (1 - \omega) x_{n-1} \tag{19}$$

To sort this issue a fix would be damp the update. This is called as the relaxation scheme ($0 < \omega \leq 1$). To accelerate the convergence you can also have $1 < \omega \leq 2$. Selection of ω is mainly done with experience or trial and error. **Experience at IBM and IIT Bombay – damping the tunneling generation rate. Currently damping the Poisson solver** Another important point that you should remember is this method is sensitive to the slope for f(x) at the current root x_n. If the slope is very small then it will lead to divergence of the root.

Application for the optimization problem and extension to Gradient descent method





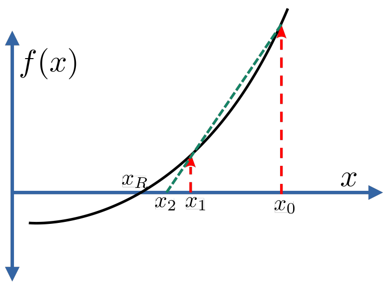
2.4.2 Secant Method

In many cases it is not so simple to calculate the derivative of the function (f(x)), (for example if we want to evaluate the derivative of the eigen energy with respect to the effective mass in the Schrodinger equation – case study of the effective mass paper). This may be due to the fact the function provided to us in the form of a data. What can we do in such a case ... We can employ the Newton Raphson method but instead of the $f'(x)$ we will approximate it with the numerical derivative

$$x_n = x_{n-1} - f(x_{n-1}) \frac{x_{n-2} - x_{n-1}}{f(x_{n-2}) - f(x_{n-1})} \tag{20}$$

Geometrically its meaning is as shown

Note the geometrical similarity between the Secant and Regula Falsi method. Even though the Secant method



also requires two initial guess like the Regula Falsi method but in this case it is not required the root lie between the (x_{i-2}, x_{i-1}). This may lead to divergence in some cases, but in general it will converge faster than the Regula Falsi method. Another difference is the way in which the roots are updated. In the Regula Falsi method the roots are updated - in regula falsi the roots are updated such that the actual root lies in the region between the approximate root while in the secant method the roots are updated sequentially i.e. x_{i-1} becomes x_{i-2} x_i is mapped onto x_{i-1} and so on and so forth. Thus we may lose the bracketing property and this may lead to divergence.

2.5 Convergence

The order of convergence in ascending order is as follows Bisection Method, Regula Falsi for the bracketing method, and Fixed point method, Secant method and Newton Raphson method. As we have see that as the convergence rate increases there also a chance of divergence, hence any one method may or may not work Role of initial guess