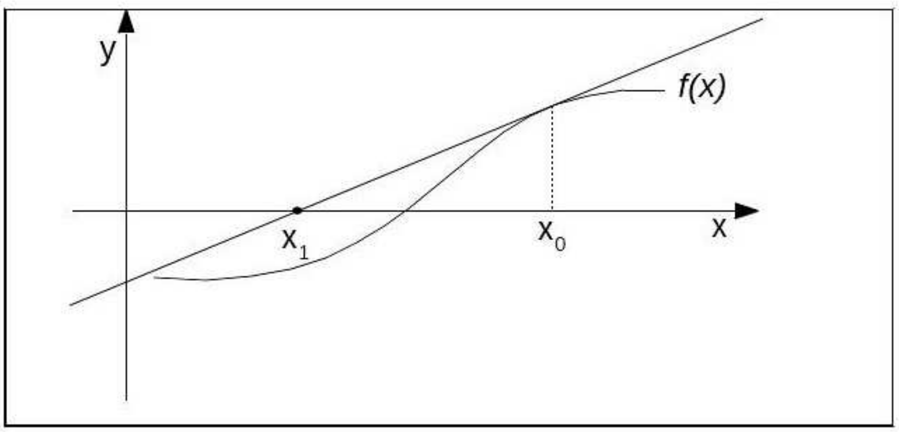
CS323 LECTURE NOTES - LECTURE 4

1 Newton's Method

Newton's method converges to the solution faster than the bisection method, so it is one of the most widely used methods in practice. There are two different derivations of the formula used by Newton's Method.

1.1 Graphical derivation

One way to obtain Newton's formula is to approximate $f(x)$ using a line that is tangent to $f(x)$ at the point $(x_0, f(x_0))$, and using the x intercept of this line to determine a value x_1 as an approximation to the root of $f(x)$.



The slope of the line tangent to $f(x)$ that passes through the point $x = x_0$, is $f'(x_0)$, so the equation of this line is:

$$y - f(x_0) = f'(x_0)(x - x_0)$$

The x intercept of this line occurs when $y = 0$, $x = x_1$, therefore:

$$0 - f(x_0) = f'(x_0)(x_1 - x_0)$$

Solving for x_1:

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

We can then use this x_1 as the new x_0 and repeat the process in order to find a better approximation of a root of $f(x)$. Before we proceed to talk about the algorithm itself we will explain how to obtain Newton's formula using Taylor series.

1.2 Using a Taylor Series

Even though the graphical derivation of Newton's formula is easy to understand, it does not provide a way to compute the rate of convergence of the algorithm. We will now use a Taylor series to obtain Newton's formula and later use it to compute the convergence rate of Newton's Algorithm using Taylor's remainder Theorem.

We know that a function that satisfies several continuity requirements can be approximated by the Taylor Series:

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + f''(x_0)\frac{(x - x_0)^2}{2!} + f^{(3)}(x_0)\frac{(x - x_0)^3}{3!} + \dots$$

If we only keep the first two terms of the series we get

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2}(x - x_0)^2 f''(\xi)$$

Recall from what we talked about before, that the remainder (last term) corresponds to the error if we only use the first two terms of the series. Since we are interested in an approximate solution, we have that:

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0)$$

Since the equation that we want to solve is $f(x) = 0$, we can find the value x_1 such that $f(x_1) = 0$. From this observation we get:

$$f(x_0) + f'(x_0)(x_1 - x_0) = 0$$

Solving for x_1 we get

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

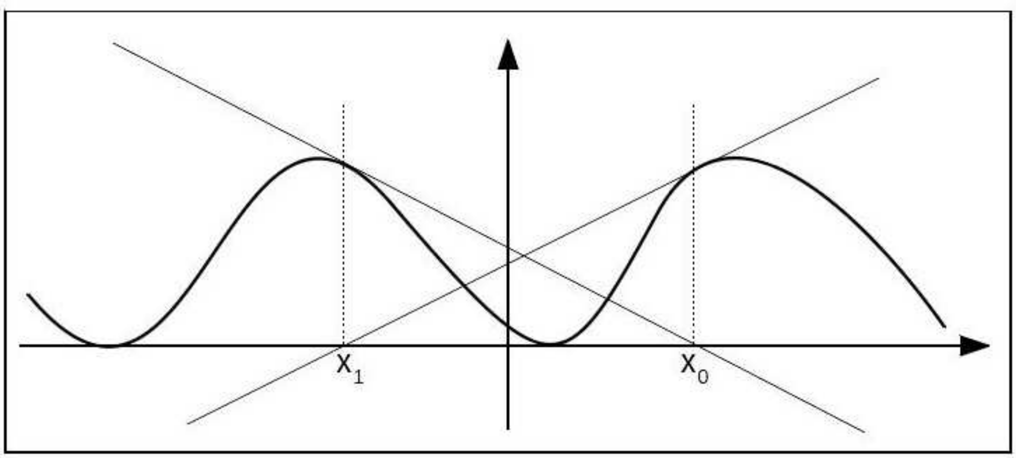
As mentioned before, this formula to compute x_1 can be used to design an algorithm assuming that we initially have an approximation x_0 to the solution of the equation $f(x) = 0$. Using the formula given above, we can find a second approximation x_1 to the solution. We can now use this approximation as our new x_0 and repeat the process again and again until the difference between two successive approximations is less than the given error tolerance ϵ.

1.3 Non-convergence of Newton's Method

A big difference with respect to the Bisection Method is that Newton's Method might actually not converge to a solution. There are two cases that we have to be aware of:

1. We can see that the derivative of $f(x)$ is in the denominator of Newton's formula, so if $f'(x_0) = 0$ for some iteration of the method, the algorithm must stop with an error condition.

2. Another problem is when the values of x_0 y x_1 are constantly bouncing off one another without ever converging. For example if $f(x) = \cos x$, if we start with $x_0 = 2.975086$, and use Newton's formula to compute x_1, we get $x_1 = -2.975086$, and if we now repeat the process using the new $x_0 = -2.975086$, we get again the same value that we started with, $x_1 = 2.9750863217$. This process could be repeated forever, so it will be necessary to provide, as part of the input, an integer N which will limit the number of iterations and avoid the program ending up in an infinite loop as can be seen in the figure.



1.4 Newton's Algorithm

Newton's Algorithm
INPUT: function $f : \mathbb{R} \to \mathbb{R}$, function $f' : \mathbb{R} \to \mathbb{R}$
$\qquad x_0, \epsilon \in \mathbb{R}$, $N \in \mathbb{N}$
$i \leftarrow 0$
repeat
$\qquad x_1 \leftarrow f(x_0) - \dfrac{f(x_0)}{f'(x_0)}$
$\qquad i \leftarrow i + 1$
$\qquad e \leftarrow |x_1 - x_0|$
$\qquad x_0 \leftarrow x_1$
until $e \leq \epsilon$ or $i > N$
if $i \leq N$
$\qquad$ return x_1
else
$\qquad$ halt with error (no solution found)

1.5 Example

Let us now solve the same equation that we used as an example in the case of the bisection method so that we can compare how the methods work and its convergence rate.

The inputs to the algorithm are:

$$f(x) = x^2 - 2; \quad x_0 = 1; \quad \epsilon = 10^{-10}$$

In our case Newton's formula:

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

$$x_1 = x_0 - \frac{x_0^2 - 2}{2x_0}$$

Here is a table illustrating the tracing of the computation:

i	x_0	x_1	$\lvert x_0 - x_1 \rvert$
1	1	1.5	0.5
2	1.5	1.4166666667	0.0833333333
3	1.4166666667	1.4142156863	0.0024509804
4	1.4142156863	1.4142135624	2.12×10^{-6}
5	1.4142135624	1.4142135624	1.59×10^{-12}

Notice that after 5 iterations we get an error of less than 10^{-10}, which is much better than the 10^{-2} that we got after 7 iterations of the Bisection Method.

1.6 Convergence

Recall that we used the Taylor series approximation to $f(x)$ in order to obtain Newton's formula:

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2}f''(\xi)(x - x_0)^2$$

We know that the error term is given by the remainder

$$R_n(x) = \frac{1}{(n+1)!}f^{(n+1)}(\xi)(x - x_0)^{n+1}$$

Since in the case of Newton's formula we are using a linear approximation then $n = 1$ and we have that

$$R_1(x) = \frac{1}{2}f^{(2)}(\xi)(x - x_0)^2$$

where ξ is between x_0 and x.

In order to compute the error term, let us assume that r is a root of $f(x) = 0$, which means that $f(r) = 0$. Let us now substitute this r in the Taylor series approximation of $f(x)$ given above, i.e.

$$f(r) = f(x_0) + f'(x_0)(r - x_0) + \frac{1}{2}f''(\xi)(r - x_0)^2$$

dividing by $f'(x_0)$ we obtain

$$\begin{aligned} 0 &= \frac{f(x_0)}{f'(x_0)} + (r - x_0) + \frac{1}{2}\frac{f''(\xi)}{f'(x_0)}(r - x_0)^2 \\ r - \left(x_0 - \frac{f(x_0)}{f'(x_0)}\right) &= -\frac{1}{2}\frac{f''(\xi)}{f'(x_0)}(r - x_0)^2 \end{aligned}$$

Suppose that x_0 and x_1 are two successive approximations to a using Newton's Algorithm ,i.e. $x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$, and taking the absolute value on both sides, we obtain

$$|r - x_1| = \frac{1}{2}\frac{f''(\xi)}{f'(x_0)}(r - x_0)^2$$

Notice that $e_0 = |r - x_0|$, and $e_1 = |r - x_1|$ are the absolute errors of approximating r with x_1 and x_0 respectively, hence

$$e_1 = \frac{1}{2} \left| \frac{f''(\xi)}{f'(x_0)} \right| e_0^2$$

where $\xi \in [r, x_0]$

If x_0 is close to the actual root r, we have that $\frac{f''(\xi)}{f'(x_0)}$ is almost constant for several iterations of the algorithm, and based on the previous formula

$$e_1 \approx c e_0^2$$

Where e_1 happens to be the error of next approximation after e_2, so we can write

$$e_{i+1} \approx c e_i^2$$

where i is the iteration number in Newton's Algorithm, and c is a positive constant. This formula shows that the error is reduced quadratically after each iteration. Compare this with the linear reduction of the error in the case of the Bisection Method.