

Newton-Raphson Iteration in C

Randolph Langley

April 11, 2017

The classical Newton-Raphson method is a simple and efficient means of computing roots for many “well-behaved” equations.

The method consists of simply iterating

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

until $|x_{i+1} - x_i| < \epsilon$ where ϵ is some suitably small number.

Thus for the computation of an arbitrary square root $x^2 = c$, the equation would be $f(x) = x^2 - c$, and the iteration would be over:

$$x_{i+1} = x_i - \frac{x_i^2 - c}{2x_i}$$

A simple C program to compute this is found in listing 1 on page 2; you can see that this iteration quickly converges to 14.730919862 when $N = 217$ by looking at the figure 1 on page 3.

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define CUTOFF 0.0000000001

// Newton-Raphson iteration:
//
//  $x_{i+1} = x_i - (x_i^2 - c) / (2x_i)$ 

double nr(double c)
{
    double x_prev = 1.0;
    double x_current =
        x_prev - (x_prev * x_prev - c) / (2 * x_prev);
    double diff = fabs(x_prev - x_current);
    int count = 0;

    while(diff > CUTOFF)
    {
        x_current =
            x_prev - (x_prev * x_prev - c) / (2 * x_prev);
        diff = fabs(x_prev - x_current);
        x_prev = x_current;
        count++;
    }

    return(x_current);
}

int main(int argc, char *argv[])
{
    if(argc > 1)
        nr(atof(argv[1]));
}

```

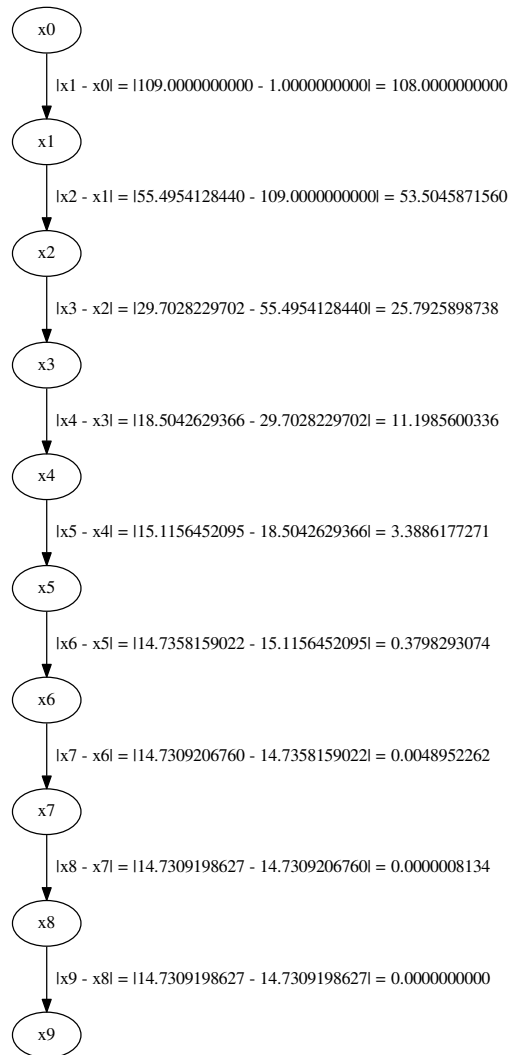


Figure 1: Convergence when computing $\sqrt{217}$ with Newton-Raphson method