Newton-Raphson’s Method

Dr. Rahul Das Gupta

Finding the exact solution is not always possible...

Example: $\boxed{3x + 8^{x} + 5x^{2} \ln(\sin^{-1} x) = 0}$

Approximate Solution of the equation $f(x)=0$

Iterative Approach (Successive Approximation)

$$x = \boxed{x^{(n)}} + \boxed{h}$$

Actual Solution

Correction Factor

Solution obtained at n th iteration

Taylor Series Expantion

$$f(x^{(n)} + h) = f(x^{(n)}) + hf'(x^{(n)}) + \frac{h^{2}}{2!} f''(x^{(n)}) + \dots$$

$$\simeq f(x^{(n)}) + hf'(x^{(n)})$$

Neglecting terms involving higher power of h

Given Equation is: $\boxed{f(x)=0}$

Here, $x = x^{(n)} + h$

Hence, $\boxed{f(x) = f(x^{(n)} + h) = 0}$

Again, $\boxed{f(x^{(n)} + h) \simeq f(x^{(n)}) + hf'(x^{(n)})}$

Hence, $f(x^{(n)}) + hf'(x^{(n)}) = 0 \quad \therefore \quad \boxed{h = -\frac{f(x^{(n)})}{f'(x^{(n)})}}$

Correction Factor: $\boxed{h = -\frac{f(x^{(n)})}{f'(x^{(n)})}}$

Again, $x = x^{(n)} + h = \boxed{x^{(n)} - \frac{f(x^{(n)})}{f'(x^{(n)})}}$

** The above solution may not be a good one too and may give better solution in the next iteration.

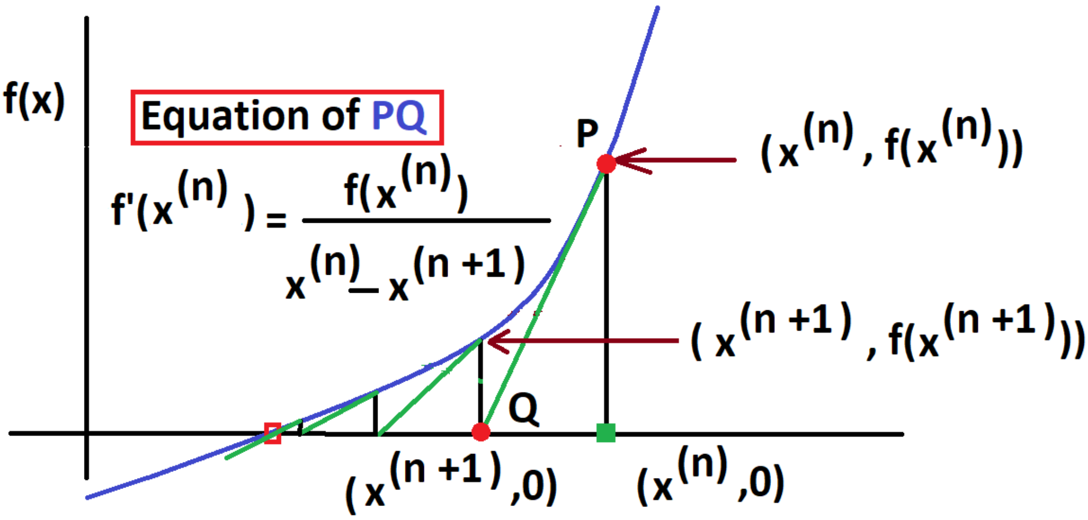
** Here, $f'(x^{(n)})$ is not equals to zero.

Solution in the next iteration:

$$\boxed{x^{(n+1)} = x^{(n)} - \frac{f(x^{(n)})}{f'(x^{(n)})}}$$

where, $f'(x^{(n)})$ is not equals to zero.

The above process is repeated until a sufficiently small value of $f(x^{(n+1)})$ is reached.



Geometrically, $(x^{(n+1)}, 0)$ is the intersection of the x-axis and the tangent of the curve of $f(x)=0$ at $(x^{(n)}, f(x^{(n)}))$.

Limitation:

Newton-Raphson's method will fail in cases where the derivative is zero. When the derivative is close to zero, the tangent line is nearly horizontal and hence may overshoot the desired root (numerical difficulties).

C-Program for Newton-Raphson's Method:

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

float function(float );
float function_derivative (float );
void newton_raphson_method(float, float );
```

```
void main( )
{
    float  allowed_error = 0.0, seed_value;
    printf("\nEnter a Seed Value:\t");
    scanf("%f", &seed_value);
    printf("\nEnter Allowed Error:\t");
    scanf("%f", &allowed_error);
    newton_raphson_method(value, allowed_error);
}

void newton_raphson_method(float seed_value,
                                        float allowed_error)
{
    float value = seed_value;
    float h = -function(value) / function_derivative (value);
    for ( ;fabs(function(value)) >= allowed_error; )
    {
        if( fabs(function_derivative (value))>0.000000001)
                h = -function(value) / function_derivative (value);
        else
       {
              printf(“\n Newton-Raphson’s method fails.”);
              exit(0);
        }

        value = value + h;
    }
    printf("\n Root Value: %f", value);
    printf("\n Functional Value: f(%f) = %f", value, function(value));
}

float function(float value)
{
    return (value * value * value - 2);
}
```

```
float function_derivative (float value)
{
    return (3 * value * value);
}
```