Newton-Raphson's Method:

Newton's Method is a technique for approximating a root to an equation of the form f(x)=0. It is a method for finding successively better approximations to the roots (or zeroes) of a real-valued function. The Newton-Raphson method in one variable is implemented as follows:

The method starts with a function f defined over the real number x, the function's derivative f, and an initial guess x0 for a root of the function satisfies the assumption made in the derivation of the formula and the initial guess is close, then a better approximation x1 is

$$X_1=x_0 - f(x_0)/f'(x_0)$$

Where,

- X_0 is the initial value of x,
- $f(x_0)$ is the value of the equation at initial value, and
- $f'(x_0)$ is the value of the first order derivative of the equation or function at the initial value x_0.

Note: $f'(x_0)$ should not be zero else the fraction part of the formula will change to infinity which means f(x) should not be a constant function.

It is one of the fastest convergences to the root. It converges on the root quadratic. It involves iteratively refining an initial guess to converge it toward the desired root.

However, it is not efficient to calculate the roots of the polynomials or equations with higher degrees but in the case of small-degree equations, this method yields quick result.

Newton Raphson Method Formula

The formula is written as follows:

$$X_{n+1} = x_n - f(x_n)/f'(x_n)$$

Where,

- x_n is the estimated $(n)^{th}$ root of the function,
- $f(x_n)$ is the value of the equation at $(n)^{th}$ estimated root, and
- $f'(x_n)$ is the value of the first order derivative of the equation or function at $x_{n.}$

Convergence of Newton Raphson Method

The Newton-Raphson method tends to converge if the following condition holds true:

$$|f(x).f''(x)|< |f'(x)|^2$$

It means that the method converges when the modulus of the product of the value of the function at x and the second derivative of a function at x is lesser than the square of the modulo of the firs derivative of the function at x. The Newton-Raphson Method has a convergence of order 2 which means it has a quadratic convergence.

Application of NRM

- To find the square root of any number
- To find the inverse
- To find the inverse square root
- Root of any given equation.

The main use of NRM is that it can approximate solutions to an equation with incredible accuracy. It is also a method to approximate numerical solutions (i.e x-intercepts, zeros, or roots) to equations that are too hard for us to solve by hand.

Example: Verify that the equation x^3 – 5x – 40 = 0 has a root x =3 and x=4. Use the Newton-Raphson Method to find the root.

$$X_{n+1} = x_n – f(x_n)/f'(x_n)$$

$f(x) = x^3 – 5x – 40$

putting x = 3 gives

$f(x) = f(3) = 3^3 – 5(3) – 40$

$f(3) = -28$

and putting x = 4 gives

$f(x) = f(4) = 4^3 - 5(4) – 40$

$f(4) = +4$

Sine f(x) has changed sign between x = 3 and x = 4, the graph of the function must cross the x-axis in this interval, so there is a root between 3 and 4. The Newton-Raphson Iterative formula is

$$X_{n+1} = x_n – f(x_n)/f'(x_n$$

$f'(x) = 3x^2 – 5$

Using SageMath, the solution is

```
x = var('x')
f(x) = (x**3) - 5*x - 40
df = diff(f, x)
NewtonIt(x) = x - f(x) / df(x)

x_initial = 2  # Initial guess
tolerance = 1e-6  # Tolerance for convergence

iteration = 4
while True:
    x_new = NewtonIt(x_initial)
    print(f"Iteration {iteration}: x{iteration} = {x_new.n(digits=6)}")

    if abs(x_new - x_initial) < tolerance:
        print(f"Hence, {x_new.n(digits=6)} is the root of {f(x)}")
        break

    x_initial = x_new
    iteration += 1
```

and the result of the coding is

```
Iteration 1: x1 = 3.90698
Iteration 2: x2 = 3.90445
Iteration 3: x3 = 3.90445
Iteration 4: x4 = 3.90445
Hence, 3.90445 is the root of x^3 - 5*x - 40
```

2. Using Newton-Raphson Iterative formula, find the root of the equation $x^3 - 3$

Using SageMath, the solution is

```
x = var('x')
f(x) = (x**3) - 3
df = diff(f, x)
NewtonIt(x) = x - f(x) / df(x)

x_initial = 2  # Initial guess
tolerance = 1e-6  # Tolerance for convergence

iteration = 1
while True:
    x_new = NewtonIt(x_initial)
    print(f"Iteration {iteration}: x{iteration} = {x_new.n(digits=6)}")

    if abs(x_new - x_initial) < tolerance:
        print(f"Hence, {x_new.n(digits=6)} is the root of {f(x)}")
        break

    x_initial = x_new
    iteration += 1
```

the answers are

```
Iteration 1: x1 = 1.58333
Iteration 2: x2 = 1.45445
Iteration 3: x3 = 1.44235
Iteration 4: x4 = 1.44225
Iteration 5: x5 = 1.44225
Hence, 1.44225 is the root of x^3 - 3
```

3. Use Newton's Method to find the only real root of the equation $x^3 - x - 1 = 0$.

We have $f(x) = x^3 - x - 1$ and the $f'(x) = 3x^2 - 1$. Since $f(1) = -1$ and $f(2) = 5$, the function has a root in the interval [1,2] since function changes sign between [1,2]. Let us make the intial guess $x_0 = 1.5$.

Newton's formula here is

$$X_{n+1} = x_n - (x^3_n - x_n - 1) / (3x^2_n - 1)$$

Using SageMath, the solution is

```
x = var('x')
f(x) = (x**3) - x - 1
df = diff(f, x)
NewtonIt(x) = x - f(x) / df(x)

x_initial = 1.5  # Initial guess
tolerance = 1e-6  # Tolerance for convergence

iteration = 4
while True:
    x_new = NewtonIt(x_initial)
    print(f"Iteration {iteration}: x{iteration} = {x_new.n(digits=6)}")

    if abs(x_new - x_initial) < tolerance:
        print(f"Hence, {x_new.n(digits=6)} is the root of {f(x)}")
        break

    x_initial = x_new
    iteration += 1
```

the answers are

```
Iteration 4: x4 = 1.34783
Iteration 5: x5 = 1.32520
Iteration 6: x6 = 1.32472
Iteration 7: x7 = 1.32472
Hence, 1.32472 is the root of x^3 - x - 1
```