Assignment

Numerical Analysis (AMP-621)

Name: Arun Kumar

Reg. No.: 1830058

Branch: BE GCS

Aim: WAP to solve this equation x^3-3x-5=0 using Newton Raphson Method.

Theory: The Newton-Raphson method is one of the most widely used methods for root finding. It can be easily generalized to the problem of finding solutions of a system of non-linear equations, which is referred to as Newton's technique.

The most basic version starts with a single-variable function f defined for a real variable **x**, the function's derivative **f ′**, and an initial guess **x0** for a root of **f**. If the function satisfies sufficient assumptions and the initial guess is close, then

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

is a better approximation of the root than **x0**. Geometrically, **(x1, 0)** is the intersection of the x-axis and the tangent of the graph of **f** at **(x0, f (x0))**: that is, the improved guess is the unique root of the linear approximation at the initial point. The process is repeated as

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

until a sufficiently precise value is reached.

Program:

```
using namespace std;
#include <iostream>
#include<conio.h>
#include<iomanip>
#include<math.h>

float f(float x) {
    return (x*x*x-(3*x)-5); // Given Equation X^3-3X-5=0
}

float df(float x) {
    return (3*(x*x)-3);
}

int main() {
 int itr,maxitr;
 float h,x0,x1,aerr;
 cout<<"Enter X0, Allowed Error, " <<"Maximum Iteration"<<endl;
 cin>>x0>>aerr>>maxitr;
 for(itr=1;itr<=maxitr;itr++) {
   h=f(x0)/df(x0);
   x1=x0-h;
   cout<<"Iteration no."<<setw(3)<<itr<<"x="<<setw(9)<<setprecision(6)<<x1<<endl;
   if(fabs(h)<aerr) {
     cout<<"After"<<setw(3)<<itr <<" iteration , Root="<<setw(8)<<setprecision(6)<<x1;
     return 0;
   }
   x0=x1;
 }
 cout<<"Iteration not sufficient,"<<"solution does not converge"<<endl;
 return 1;
}
```

Output:

C:\TURBOC3\BIN\NM\NR.exe

```
Enter X0, Allowed Error, Maximum Iteration
2.2 0.00001 20
Iteration no.   1x=   2.28264
Iteration no.   2x=   2.27903
Iteration no.   3x=   2.27902
After  3 iteration , Root= 2.27902
Process returned 0 (0x0)   execution time : 17.705 s
Press any key to continue.
```

Root = 2.27902

Advantages:

1. One of the fastest methods which converges to root quickly.
2. Converges on the root quadratically i.e. rate of convergence is 2.
3. As we go near to root, number of significant digits approximately doubles with each step.
4. It makes this method useful to get precise results for a root which was previously obtained from some other convergence method.

Disadvantages:

1. We must find the derivative to use this method.
2. If the tangent is parallel or nearly parallel to the x-axis, then the method does not converge.
3. Dependent on initial guess
    - May be too far from local root
    - May encounter a zero derivative
    - May loop indefinitely

Flowchart: