TITLE: NEWTON-RAPHSON METHOD FOR FINDING ROOTS OF EQUATIONS

Definition:

The Newton-Raphson method is an iterative numerical technique used to find approximate roots of a real-valued function. It uses the function and its derivative to rapidly converge to a solution, starting from an initial guess.

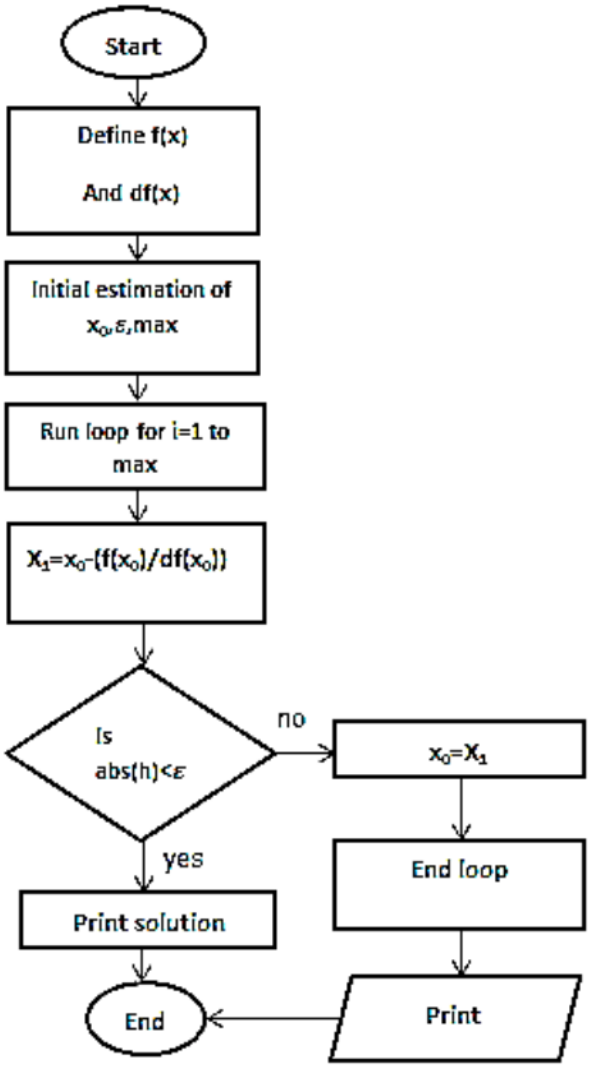
Algorithm:

1. Define function $f(x)$, $f'(x)$ & error ϵ.
2. Input the initial root value x_0.
3. Calculate the value of root:

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

4. If $f(x_1) = 0$
   then print ("root lies at $x = x_1$");
   Go to step 6
   End
5. If $|x_1 - x_0| > \epsilon$ then
   Set
   $x_0 = x_1$;
   $f(x_0) = f(x_1)$;
   $f'(x_0) = f'(x_1)$;
   Go to step 3
   else
   print ("the root lies at $x = x_1$");
   End
6. Stop

Flowchart:



```
%Title:-To calculate root of given eqn using NR method
%Developed by:- Arpan Adhikari
%Date:- 26th June
%---Three critical statements
close all;
clear variables
clc;
%---function declaration section---
syms x;
func = input('Enter function f(x)=');
f = inline(func);
df = diff(f(x));
g = inline(df);
disp(f);
disp(g);
E = 0.0005;
%---User i/p section---
x0 = input('Enter inital value for x=');
f0 = f(x0);
g0 = g(x0);
%--- Calculation Section ----
x1 = x0 - (f0/g0);
f1 = f(x1);
disp('_____');
disp('x0 f(x0) df(x0) x1 f(x1)');
disp('_____');
out = [x0, f0, g0, x1, f1];
disp(out);
while(abs(x1-x0)>E)
    x0= x1;
    f0 = f1;
    g0 = g(x0);
    x1 = x0 - (f0/g0);
    f1 = f(x1);
    out = [x0, f0, g0, x1, f1];
disp(out);
end

%Output final result
disp('.....')
;
result = strcat('The root lies  at x = ', num2str(x1), ' with f(x) = ',
num2str(f1));
display(result);
```

Output:

```
enter the function f(x)= x-cos(x)

     Inline function:
     f(x) = x-cos(x)


     Inline function:
     gr(x) = sin(x)+1.0

enter the initial value for root x0= 2
-------------------------------------------------
   x0       f(x0)     df(x0)       x1       f(x1)
-------------------------------------------------
    2.0000    2.4161    1.9093    0.7345   -0.0076

    0.7345   -0.0076    1.6702    0.7391    0.0000

    0.7391    0.0000    1.6736    0.7391    0.0000


Root found at x = 0.739085 after 3 iterations
```

Conclusion:

The Newton-Raphson method is a powerful and efficient tool for finding the roots of equations.

TITLE: LAGRANGE INTERPOLATION METHOD

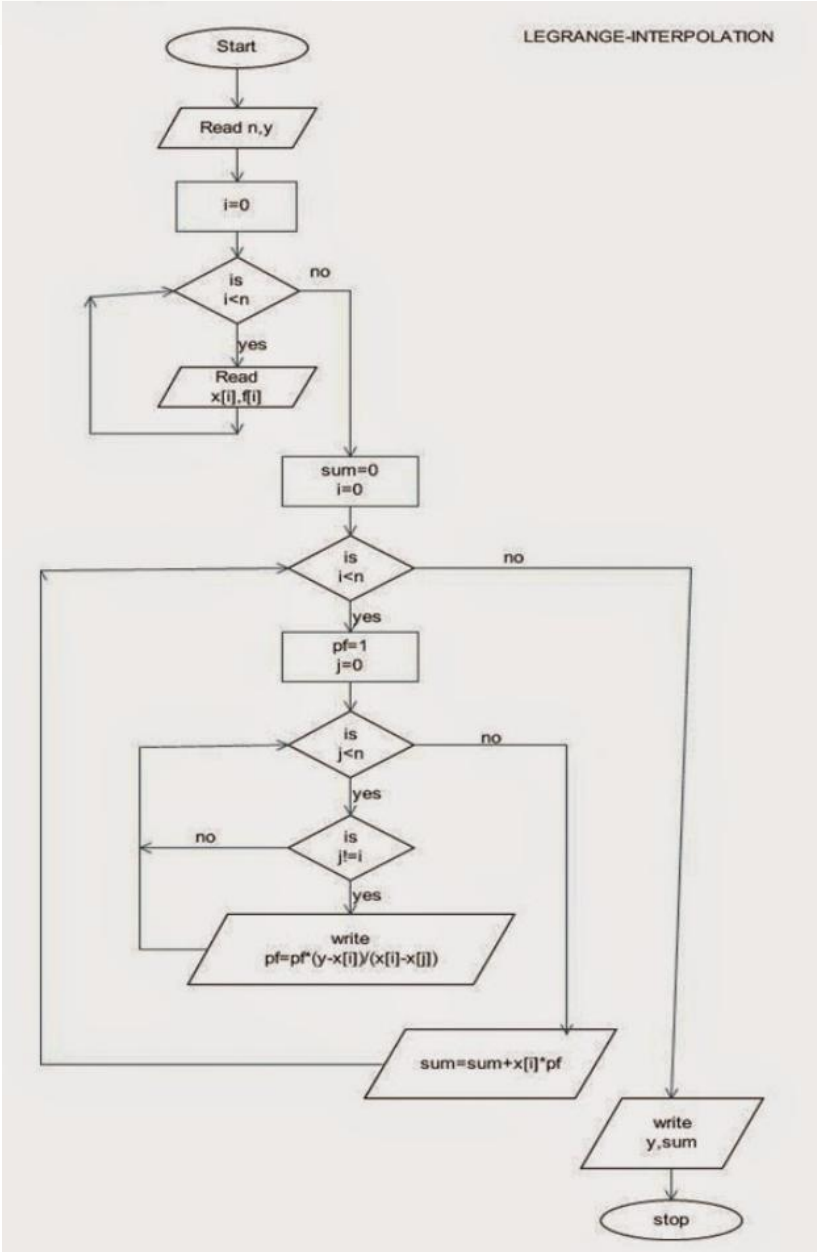
Definition:

Lagrange interpolation is a numerical technique used to estimate the value of a function at a desired point, given a set of known discrete data points. It constructs a polynomial that passes exactly through all the provided data points, allowing for the interpolation of intermediate values.

Algorithm:

1. **Start**
2. **Input** the initial values for x and y arrays (data points).
3. **Check** that the lengths of x and y are equal.
   - If not, prompt the user to re-enter the arrays until their dimensions match.
4. **Input** the value X at which the interpolated value Y is to be found.
5. **Initialize** $sum = 0$.
6. **For** $i = 1$ to n (number of data points):
   a. Set $prod = 1$
   b. **For** $j = 1$ to n:
   - If $i \neq j$:
     $prod = prod \times \frac{(X - x_j)}{(x_i - x_j)}$
     c. $sum = sum + prod \times y_i$
7. **Output** the interpolated value $Y = sum$ at point X.
8. **Stop**

Flow Chart:

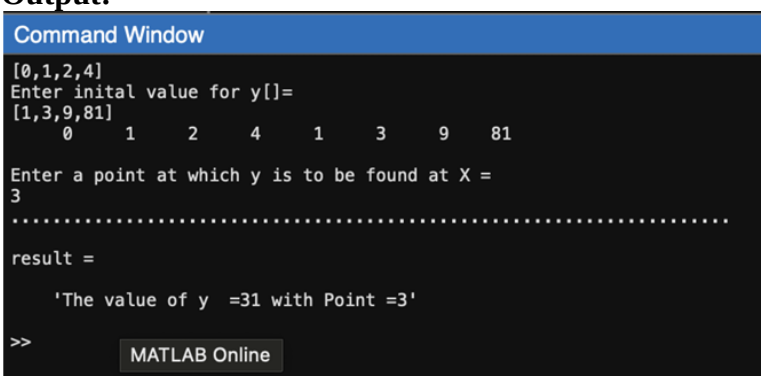


```
%Title:-To determine functional value of a discrete function at any arbitary
point using Langrange interpolation
%Developed by:- Arpan Adhikari
%Date:- 26th June
%---Three critical statements
close all;
clear variables
clc;
%---User i/p section---
x = input('Enter inital value for x[]=');
y = input('Enter inital value for y[]=');
while(length(x)~=length(y))
   clc;
   disp('Dimension of x and y must match');
   x = input('Enter inital value for x[]=');
    y = input('Enter inital value for y[]=');
end
out=[x;y];
disp(out);

X = input('Enter point at which y is to be found X = ');

%-----Calculation--------
sum = 0;
n = length(x);
for i= 1:n
    prod = 1;
    for j = 1:n
        if(i ~= j)
            prod = prod*(X-x(j))/(x(i)-x(j));
        end
    end
    sum = sum + prod * y(i);
end
%Output final result
disp('.....................................................................')
;
result = strcat('The value of y is = ', num2str(sum), ' at point x = ',
num2str(X));
disp(result);
```

Output:



Conclusion:

Lagrange interpolation provides a straightforward way to estimate the value of a function at any point within the range of given data. The method is exact for the provided data points and is widely used in engineering and science for curve fitting and data analysis.