DHAKA COLLEGE

DEPT. OF PYSICS

Honours 3rd Year

Test - 2020

Name: Elias Bhuiyan

Reg (DU): 1812700 2064

Class Roll : 220171802 7045

Subject: Computational Physics

Sub code: PH 307

Session : 2017 - 2018

Mobile : 01767767287

Ans to the Q N-1

Newton-Raphson method:

$$f(x) = x^2 - 3x + 2$$

$$\Rightarrow f'(x) = 2x - 3$$

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

$$= \frac{2x_n^2 - 3x_n - x_n^2 + 3x_n - 2}{(2x_n - 3)}$$

$$\therefore x_{n+1} = \frac{(x_n^2 - 2)}{(2x_n - 3)}$$

$x_0 = 0$

$x_1 = 0.667$

$x_2 = 0.9334$

$x_3 = 0.996$

$x_4 = 0.999$

$x_5 = 0.999$

Hence the required root is 0.999 Ans

Solving with C++:

```
#include <bits/stdc++.h>
#define EPSILON 0.001
// The function is x^2 - 3x + 2
double func (double x)
{
    return x*x - 3*x + 2
}
double derivFunc (double x)
// Derivative of the above function which is 2*x - 3
double derivFunc (double x)
{
    return 2*x - 3
}
// Function to find the root
void newtonRaphson (double x)
{
    double h = func(x) / derivFunc(x);
    while (abs(h) >= EPSILON)
    { h = func(x) / derivFunc(x);
```

```
// x(i+1) = x(i) - f(x)/f'(x)
    x = x - h;
}
cout << "The value of the root is: " << x;
}

// Drive program to test above

int main()
{
double x0 = 0; // Initial values assumed
  newton Raphson (x0);
  return 0;
}
```

output :

The value of root is : 0.999

Ans to the Q.N. 2

Trapezoidal rule method:

let, $y = \frac{1}{1+x^2}$

Here, $a=0$, $b=1$, we shall divide the interval into six equal parts.

Now,

$x_0 = 0 \longrightarrow y_0 = 1$

$x_1 = 1/6 \longrightarrow y_1 = 0.972$

$x_2 = 2/6 \longrightarrow y_2 = 0.900$

$x_3 = 3/6 \longrightarrow y_3 = 0.800$

$x_4 = 4/6 \longrightarrow y_4 = 0.692$

$x_5 = 5/6 \longrightarrow y_5 = 0.590$

$x_6 = 1 \longrightarrow y_6 = 0.500$

Now By Trapezoidal rule, we have

$$\int_0^1 \frac{1}{1+x^2}\,dx = \frac{h}{2}\left[(y_0+y_6) + 2(y_1+y_2+y_3+y_4+y_5)\right]$$

$$= \frac{1}{12} \times (9.410) = \boxed{0.784}$$

$\therefore \pi/4 = 0.784 \Rightarrow \pi = 3.1369$ ans

Solving with C++:

```
#include <studio.h>
//approximate value in computed using trapezoidal
// rule
    float y (float x)
{// Declaring the function f(x) = 1/(1+x*x)
    return 1/(1+x*x);
}
// Function to evalute the value of integral
 float trapezoidal (float a, float b, float n)
{// Grid spacing
 float h = (b-a)/n;
// Computing sum of first and last terms
// in above formula
 float s = y(a)+y(b);
// Adding middle terms in above formula
 for (int i=1; i<n; i++)
        s+ = 2* y(a+i*h);
        return (h/2)*s;
}
```

```
int main()
{float x0=0;
 float xn = 1;
 int n = 6;
 printf("Value of integral is %6.4f\n",
        trapezoidal(x0, xn, n));
 return 0;
}
```

Output: value of integral is **0.784**

Ans

Ans to the Q.N. 3

Simpsons (1/3) method:

Let, $y = f(x) = 2x$

Here, $a=0$, $b=1$, we shall divide the interval into six equal parts.

Hence, $h = \frac{b-a}{n} = \frac{1}{6}$

Now,

$x_0 = 0 \longrightarrow y_0 = 0$

$x_1 = x_0 + h = \frac{1}{6} \longrightarrow y_1 = \frac{1}{3}$

$x_2 = \frac{1}{3} \longrightarrow y_2 = 2/3$

$x_3 = \frac{1}{2} \longrightarrow y_3 = 1$

$x_4 = 2/3 \longrightarrow y_4 = 4/3$

$x_5 = 5/6 \longrightarrow y_5 = 5/3$

$x_6 = b = 1 \longrightarrow y_6 = 2$

From,

Simpson's $(1/3)$ rule is,

$$\int_0^1 2x\,dx = \frac{h}{3}\left[(y_0+y_6) + 4(y_1+y_3+y_5) + 2(y_2+y_4)\right]$$

$$= \frac{1}{18}(2+12+4)$$

$$\therefore \int_0^1 2x\,dx = 1 \quad \underline{\text{Ans}}$$

Solving with C++ :

```
# include <iostream>

# include <math.h>
// Function to calculate f(x)
 float func (float x)

{  return log (x);

}
// Function for approximate integral
 float simpsons - (float (*f)(float), float a, int n)
```

```
{// calculating the value of h
  float h= (ul-ll)/n;
// Array fore storing value of x and f(x)
  float x[10], fx[10];
// calculating values of x and f(x)
  for (int i=0; i<=n; i++)
  {
  x[i] = ll + i*h;
  fx[i] = func (x[i]);
  }
// Calculating result
  float res =0;
  for (int i=0; i<=n; i++)
  {
  if (i==0 || i==n)
  res += fx[i];
  else if (i%2 !=0];
  else
    res += 2*fx[i];
  }
  res = res * (h/3);
  return res;
}
```

```
// Drivere program
int main()
{ float lowere-limit=0;
float upere-limit=1;
int n=6;
cout<<simpsons-(lowere_limit, upperc-limit, n);
return 0;
}
```

Output: the root is [1] Ans

Ans to the Q.N.4

Simpson's (3/8) method:

let, $y = f(x) = x^2+1$

Here, $a = 1$

$b = 5$

let $n = 4$

$\therefore h = \frac{5-1}{4} = 1$

Now,

$x_0 = a = 1 \longrightarrow y_0 = 2$

$x_1 = 2 \longrightarrow y_1 = 5$

$x_2 = 3 \longrightarrow y_2 = 10$

$x_3 = 4 \longrightarrow y_3 = 17$

$x_4 = b = 5 \longrightarrow y_4 = 26$

By Simpson's (3/8) rule;

$$\int_1^5 (1+x^2)\,dx = \frac{3}{8}\left[(y_0+y_4)+3(y_1+y_2)+2(y_3)\right]$$

$$= \frac{3}{8}(28+3(15)+34)$$

$$= \frac{3}{8}\times 107$$

$$\therefore \int_1^5 (1+x^2)\,dx = 40.125 \quad \text{(ans)}$$

Solving with C++:

```
#include <iostream>
float func (float x)
{return (1+x*x);
}
// Function to perform calculations
float calculate (float lower_limit, float
                 upper_limit, int interval_limit)
{
float value;
float interval_size = (upper_limit - lower_limit)
                      / interval_limit;
float sum = func (lower_limit) + func (upper_limit);
// calculates values till integral limit
for (int i=1; i < interval_limit; i++)
{ if (i%3 == 0)
  sum = sum+2 * func (lower_limit + i*
                      interval_size);
```

```
    else
      sum = sum + 3* func(lower_limit + i*
                          interval_size);
    }
    return (3*interval_size/8)* sum;
}
    int main()
{
    int interval_limit = 5;
    float lower_limit = 1;
    float upper_limit = 5;
    float integral_res = calculate (lower_limit,
      upper_limit, interval_limit);
      cout << integral_res;
      return 0;
}
```

Output: The root is: 40.125

Ans

Ans to the Q.N. 5

Let $f(x) = x^3 + 2x - 2$

$f(0) = -2 < 0$

$f(1) = 1 > 0$

$x_0 = \frac{0+1}{2} = 0.5$

$f(0.5) = (0.5)^3 + 2(0.5)^2 - 2 = -0.875 < 0$

Here, $f(1) > 0$ & $f(0.5) < 0$

$\therefore x_1 = \frac{1+0.5}{2} = 0.75$

$\therefore f(0.75) = -0.078 < 0$

Here, $f(1) > 0$ & $f(0.75) < 0$

$\therefore x_2 = \frac{1+0.75}{2} = 0.875$

$\therefore f(0.875) = 0.419 > 0$

Here, $f(0.419) > 0$ and $f(0.75) < 0$

$\therefore x_3 = 0.564$

$\therefore f(0.584) = -0.631 < 0$

Here, $f(0.875) > 0$ and $f(0.584) < 0$

$\therefore x_4 = 0.729$

$\therefore f(0.729) = -0.151 < 0$

Here, $f(0.729) < 0$ & $f(0.875) > 0$

$\therefore x_5 = 0.802$

$\therefore f(0.802) = 0.119 > 0$

Here, $f(0.802) > 0$ and $f(0.729) < 0$

$\therefore x_6 = 0.765$

$\therefore f(0.765) = -0.020 < 0$

$\therefore x_7 = 0.783$

$\therefore f(0.783) = 0.048 > 0$

Here, $f(0.783) > 0$ and $f(0.7655) < 0$

$\therefore x_8 = 0.774$

$\therefore f(0.774) = 0.012 > 0$

Here $f(0.7655) < 0$ and $f(0.774) > 0$

$\therefore x_9 = 0.769$

$\therefore f(0.769) = -0.004 \simeq \boxed{0}$

$\therefore$ The root is $\boxed{0.769}$

Ans

Solving with C++ :

```
#include <bits/stdc++.h>
#define EPSILON 0.01
// The Function is x^3+2x-2
double func (double x)
{
    return x*x*x + 2*x - 2;
}
// Prints root of func(x) with error
of EPSILON
void bisection (double a, double b)
```

```
{ if (func(a) * func(b) >= 0)
  {cout << "You have not assumed right a & b\n";
    return 0;
  }
  double c=a;
  while ((b-a) >= EPSILON)
  {
  // Find the middle point
    c = (a+b)/2;
  // check if middle point is root
    if (func(c) == 0.0)
      break;
  // Decide the side to repeat the steps
    else if (func(c) * func(a) < 0)
          b=c;
    else
          a=c;
  }
    count << "the value of root is:" << c;
}
```

```
// Drivere program to test above function
    int main()
{
    // Initial values assumed
    double a=-2, b=1
    biseetion (a, b);
    return 0;
}
```

Output:

The value of root is [0.769] in Ans