**Romberg Algorithm Dimitrije Prosevski MATH 414**

**Source Code:**

clear

clc

format longG

fprintf("For N = 2 order:\n");

testN2 = RombergAlgorithm(2);

disp(real(testN2));

fprintf("For N = 3 order:\n");

testN3 = RombergAlgorithm(3);

disp(real(testN3));

fprintf("For N = 4 order:\n");

testN4 = RombergAlgorithm(4);

disp(real(testN4));

function R = RombergAlgorithm(N)

%initialize zero matrix

N = N + 1;

R = zeros(N,N);

interval = 2;

%for first order

for order = 1:N

R(order, 1) = trapezoid(order\*interval);

end

%for other orders

% Ij,k+1 = [(4^k)(Ij+1,k) - (Ij,k)]/4^k - 1

for column = 2:N

k = column - 1;

for row = 1:(N - column + 1)

R(row, column) = ((4^k)\*R(row+1, column-1)...

- R(row, column - 1))/(4^k-1);

end

end

%1)the triangles of data for N=2,3,4.

% Integrate 1/(1+x\*x) from 0 to 1.

function int = trapezoid(n)

syms x;

inputFunction = 1/(1+x\*x);

f=inline(inputFunction);

a = 0; b = 1;

int = 0;

h = (b-a)/n;

for x = a:h:(b-h)

int = int + 1/2 \*(f(x)+f(x+h))\*h;

end %end for loop

end %end function trapezoid

end %end of function RombergAlgorithm

**Output:**

