Nama : Azka Hariz Rabu, 11 November 2021
NIM :

ASSIGNMENT
NUMERICAL METHOD

Hitung $I = \int_0^2 -x^3 - x^2 - 5x - 3\, dx$, dengan menggunakan:

1. Analitis
2. Metode trapezoidal 1 segmen
3. Metode trapezoidal multi segmen
4. Metode Simpsons 1/3 rule
5. Metode Simpsons 3/8 rule
6. Metode Romberg
7. Metode Gauss Quadrature

Bandingkan ketelitian tiap-tiap metode dengan hasil perhitungan analitisnya!

Jawab:

1. Metode analitis

$$I = \int_0^2 -x^3 - x^2 - 5x - 3\, dx$$

$$I = -\frac{1}{4}x^4 - \frac{1}{3}x^3 - \frac{5}{2}x^2 - 3x\Big|_0^2$$

$$I = -\frac{1}{4}(2)^4 - \frac{1}{3}(2)^3 - \frac{5}{2}(2)^2 - 3(2)$$

$$I = -\frac{1}{4}(16) - \frac{1}{3}(8) - \frac{5}{2}(4) - 3(2)$$

$$I = -4 - \frac{8}{3} - 10 - 6$$

$$I = -20 - \frac{8}{3}$$

$$I = \frac{-60 - 8}{3}$$

$$I = -\frac{68}{3} = -22.66666666666666 \text{ ...}$$

2. Metode trapezoidal 1 segmen

Hasil integral = -28.000000000000000

True percent relative error = 23.529411764705877

3. Metode trapezoidal multi segmen

Jumlah Segmen	Hasil	True Percent Relative Error
2	-24.000000000000000	5.882352941176465

3	-23.259259259259256	2.614379084967300
4	-23.000000000000000	1.470588235294112
5	-22.880000000000003	0.941176470588241
6	-22.814814814814813	0.653594771241817
7	-22.775510204081630	0.480192076830717
8	-22.750000000000000	0.367647058823524
9	-22.732510288065839	0.290486564996344
10	-22.720000000000002	0.235294117647064

4. Metode Simpsons 1/3 rule

Jumlah Segmen	**Hasil**	**True Percent Relative Error ($\times 10^{-14}$)**
2	-22.6666666666667	1.56737368182375
4	-22.6666666666667	1.56737368182375
6	-22.6666666666667	3.13474736364750
8	-22.6666666666667	1.56737368182375
10	-22.6666666666667	0

5. Metode Simpsons 3/8 rule

Jumlah Segmen	**Hasil**	**True Percent Relative Error ($\times 10^{-14}$)**
3	-22.6666666666667	1.56737368182375
6	-22.6666666666667	1.56737368182375
9	-22.6666666666667	1.56737368182375

6. Metode Romberg

	$O(h^2)$	$O(h^4)$	$O(h^6)$	$O(h^8)$
$n = 1$	-28	-22.6666666666667	-22.6666666666667	-22.6666666666667
$n = 2$	-24	-22.6666666666667	-22.6666666666667	0
$n = 4$	-23	-22.6666666666667	0	0
$n = 8$	-22.7500000000000	0	0	0

Reminder	**True Percent Relative Error**
$O(h^2)$	0.367647058823524
$O(h^4)$	0
$O(h^6)$	0
$O(h^8)$	0

7. Metode Gauss Quadrature

Points	**Hasil**	**True Percent Relative Error ($\times 10^{-9}$)**
2	-22.6666666649150	7.72802998065291
3	-22.6666666693967	12.0441382357646
4	-22.6666666771373	46.1940289941300
5	-22.6666666549141	51.8494894078338

LAMPIRAN
SOURCE CODE

main.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Title        : Assignment Numerical Computation Lecture 10
% Author       : Azka Hariz
% Date         : November 7, 2021
% Code version : 1.1
% Availability : https://github.com/azkahariz/integrationMethod
%
% Please add the following citations if you use this code:
% Hariz, A (2021)  Assignment Numerical Computation Lect 10 (Version 1.1)
% [Source code]. https://github.com/azkahariz/integrationMethod
%
% In this source code, there are several integral programs:
% 1. Trapezoidal method 1 segment
% 2. Multi-segment trapezoidal method
% 3. The Simpsons 1/3 rule method
% 4. The Simpsons 3/8 rule method
% 5. Romberg's method
% 6. Gauss Quadrature Method
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all; close all; clc

% input function here
x = -3:0.01:3;
f = @fun;

% Bound
a = 0;      % lower bound
b = 2;
% b = 0.8;  % upper bound
I_true = -68/3; % Analytic

%% Trapezoidal method 1 segment
I_trap = trapezoidal(1,a,b,f);

%% Multi-segment trapezoidal method (n = 2 to 10)
n = 2:10;
I_trap = [I_trap; trapezoidal(n,a,b,f)];
clear n;

%% The Simpsons 1/3 rule method
n = 2:2:10;
I_sp13 = simpson13(n,a,b,f);
clear n;

%% The Simpsons 3/8 rule method
n = 3:3:9;
I_sp38 = simpson38(n,a,b,f);
clear n;

%% Romberg's method
n = 8;
[I_romb,v] = romberg(n,a,b,f);

%% Gauss Quadrature Method
for i=2:5
    I_gsqd(i-1,1) = gaussQuad(i,a,b,f);
end

%% True Percent Relative Error
```

```
% Error trapezoidal
e_trap = abs((I_true - I_trap)/I_true)*100;
% Error Simpson 1/3
e_sp13 = abs((I_true - I_sp13)/I_true)*100;
% Error Simpson 3/8
e_sp38 = abs((I_true - I_sp38)/I_true)*100;
% Error Romberg
e_romb = abs((I_true - v)/I_true)*100;
% Error Gauss-Quadratur
e_gsqd = abs((I_true - I_gsqd)/I_true)*100;
```

trapezoidal.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Title        : The Trapezoidal Rule (Integration)
% Author       : Azka Hariz
% Date         : November 9, 2021
% Code version : 1.1
% Availability : https://github.com/azkahariz/integrationMethod
%
% Please add the following citations if you use this code:
% Hariz, A (2021)  The Trapezoidal Rule (Integration) (Version 1.1)
% [Source code]. https://github.com/azkahariz/integrationMethod
%
% How to use:
% n is number of segment, a is lower bound of integral, b is upper bound of
% integral, and f is a function  of f(x). The output is integration result I
% and every segment width h.
% Example : f(x) = 0.2 + 25*x - 200*x^2 + 675*x^3 - 900*x^4 + 400*x^5;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [I,h] = trapezoidal(n,a,b,f)
    sum_f = zeros(max(size(n)),1);
    h(1:max(size(n)),1) = (b - a)./n;
    for j = 1:max(size(n))
        for i = 1:n(j) - 1
            sum_f(j) = sum_f(j) + f(a+i*h(j));
        end
    end
    I = h/2.*(f(a) + 2.*sum_f + f(b));
end
```

simpson13.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Title        : The Simpsons 1/3 rule method (Integration)
% Author       : Azka Hariz
% Date         : November 9, 2021
% Code version : 1.1
% Availability : https://github.com/azkahariz/integrationMethod
%
% Please add the following citations if you use this code:
% Hariz, A (2021)  The Simpsons 1/3 rule method (Integration) (Version 1.1)
% [Source code]. https://github.com/azkahariz/integrationMethod
%
% How to use:
% n is number of segment and must be even, a is lower bound of integral, b
% is upper bound of integral, and f is a function  of f(x). The output is
% integration result I, and every segment width h.
% Example : f(x) = 0.2 + 25*x - 200*x^2 + 675*x^3 - 900*x^4 + 400*x^5;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [I,h] = simpson13(n,a,b,f)
    sum_f = zeros(max(size(n)),1);
    h(1:max(size(n)),1) = (b - a)./n;
    for j = 1:max(size(n))
        for i = 0:2:(n(j)-2)
            sum_f(j) = sum_f(j) + f(a+i*h(j)) + 4*f(a+(i+1)*h(j)) +
f(a+(i+2)*h(j));
```

```
        end
    end
    I = h/3.*sum_f;
end
```

simpson38.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Title        : The Simpsons 3/8 rule method (Integration)
% Author       : Azka Hariz
% Date         : November 9, 2021
% Code version : 1.1
% Availability : https://github.com/azkahariz/integrationMethod
%
% Please add the following citations if you use this code:
% Hariz, A (2021)  The Simpsons 3/8 rule method (Integration) (Version 1.1)
% [Source code]. https://github.com/azkahariz/integrationMethod
%
% How to use:
% n is number of segment and must be multiple of 3, a is lower bound of
% integral, b is upper bound of integral, and f is a function  of f(x).
% The output is integration result I, and every segment width h.
% Example : f(x) = 0.2 + 25*x - 200*x^2 + 675*x^3 - 900*x^4 + 400*x^5;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [I,h] = simpson38(n,a,b,f)
    sum_f = zeros(max(size(n)),1);
    h(1:max(size(n)),1) = (b - a)./n;
    for j = 1:max(size(n))
        for i = 0:3:(n(j)-3)
            sum_f(j) = sum_f(j) + f(a+i*h(j)) + 3*f(a+(i+1)*h(j)) + ...
                3*f(a+(i+2)*h(j)) + f(a+(i+3)*h(j));
        end
    end
    I = (3/8)*h.*sum_f;
end
```

romberg.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Title        : Romberg's method (Integration)
% Author       : Azka Hariz
% Date         : November 11, 2021
% Code version : 1.1
% Availability : https://github.com/azkahariz/integrationMethod
%
% Please add the following citations if you use this code:
% Hariz, A (2021)  Romberg's method (Integration) (Version 1.1)
% [Source code]. https://github.com/azkahariz/integrationMethod
%
% How to use:
% n is the number of segments and must be a n = 2^i.
% a is the lower bound of the integral, and b is the upper bound of the
% integral. f is a function of f(x) to be found integral. The output is the
% result of integration I.
% Example : f(x) = 0.2 + 25*x - 200*x^2 + 675*x^3 - 900*x^4 + 400*x^5;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [I,v] = romberg(m,a,b,f)
for i = 0:log2(m)
    if i == 0
        n = 2^i;
    else
        n = [n; 2^i];
    end
end
```

```
I(1:log2(m)+1,1) = trapezoidal(n,a,b,f);
for k = 2:max(size(n))
    for j = 1:max(size(n)) - k + 1
        I(j,k) = (4^(k-1)*I(j+1,k-1) - I(j,k-1))/(4^(k-1)-1);
    end
end
for i=1:max(size(n))
    v(i,1) = I(max(size(n))+1-i,i);
end
end
```

gaussQuad.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Title        : Gauss Quadrature Method (Integration)
% Author       : Azka Hariz
% Date         : November 11, 2021
% Code version : 1.1
% Availability : https://github.com/azkahariz/integrationMethod
%
% Please add the following citations if you use this code:
% Hariz, A (2021)  Gauss Quadrature Method (Integration) (Version 1.1)
% [Source code]. https://github.com/azkahariz/integrationMethod
%
% How to use:
% n is the desired number of points. The number of points available in the
% program n = 2, 3, 4, and 5. a is the lower bound, and b is the upper
% bound. f is a function of f(x) whose integral result is sought. I is the
% output of the result of the integration.
% Example : f(x) = 0.2 + 25*x - 200*x^2 + 675*x^3 - 900*x^4 + 400*x^5;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function I = gaussQuad(n,a,b,f)
if n == 2
    x = [-0.577350269; 0.577350269];
    w = [ 1.000000000; 1.000000000];
    x = (a+b)/2 + (b-a)*x/2;
    I = (b-a)*sum(w.*f(x))/2;
elseif n == 3
    x = [-0.861136312; -0.339981044;
          0.339981044;  0.861136312];
    w = [ 0.347854845;  0.652145155;
          0.652145155;  0.347854845];
    x = (a+b)/2 + (b-a)*x/2;
    I = (b-a)*sum(w.*f(x))/2;
elseif n == 4
    x = [-0.774596669; 0.000000000; 0.774596669];
    w = [ 0.555555556; 0.888888889; 0.555555556];
    x = (a+b)/2 + (b-a)*x/2;
    I = (b-a)*sum(w.*f(x))/2;


elseif n == 5
    x = [-0.906179846; -0.538469310; 0.000000000;
        0.538469310;  0.906179846];
    w = [0.236926885;   0.478628670; 0.568888889;
        0.478628670;   0.236926885];
    x = (a+b)/2 + (b-a)*x/2;
    I = (b-a)*sum(w.*f(x))/2;
else
    error('Masukan n antara 2 sampai 5');
end
end
```