

Object Oriented Modeling and Programming in Engineering

Homework 1

The logo of Bauhaus-Universität Weimar, featuring the university's name in white sans-serif font on a solid red rectangular background.

Bauhaus-
Universität
Weimar

TEACHER:

Mathias Artus

SUBMITTED BY:

Cesar Fernando Gamba Tiusaba
Registration number 127138

1. From an oscillating energy system, you've got the equation for the power:

$$P(t) = a_1 * t + \sin(2 * \pi * f_1 * t)$$

Whereat:

$$a_1 = \frac{127138}{40000} W$$

$$f_1 = 2 \text{ Hz}$$

For this system you want to calculate the energy consumption E(t) for t=20 seconds.

$$E(t) = \int_0^t p(t) * dt$$

$$P(t) = \frac{127138}{40000} * t + \sin(2 * \pi * 2 * t)$$

$$E(t) = \int_0^{20} \left(\frac{127138}{40000} * t + \sin(2 * \pi * 2 * t) \right) * dt$$

$$E(t) = \int_0^{20} \left(\frac{63569}{20000} * t + \sin(4 * \pi * t) \right) * dt$$

$$E(t) = \left(\frac{63569 * t^2}{40000} - \frac{\cos(4 * t * \pi)}{4 * \pi} \right) \Big|_0^{20}$$

$$E(t) = \left(\frac{63569 * 20^2}{40000} - \frac{\cos(4 * 20 * \pi)}{4 * \pi} \right) - \left(\frac{63569 * 0^2}{40000} - \frac{\cos(4 * 0 * \pi)}{4 * \pi} \right)$$

$$E(t) = \left(\frac{63569}{100} - \frac{1}{4 * \pi} \right) - \left(-\frac{1}{4 * \pi} \right)$$

$$E(t) = \left(\frac{63569}{100} \right) = 635.69$$

2. UML Diagram

The UML diagram presented in the Figure 1 is used to show and explain the class **MainClass_Homework1** and the methods used to calculate and graph the area of the function using 3 different numerical methods and the analytical function, the class. The C# is called MainClass_Homework1.

The UML diagram presented in the Figure 2 is used to show and explain the class **TestFunction**, this class is used to test the class **MainClass_Homework1**

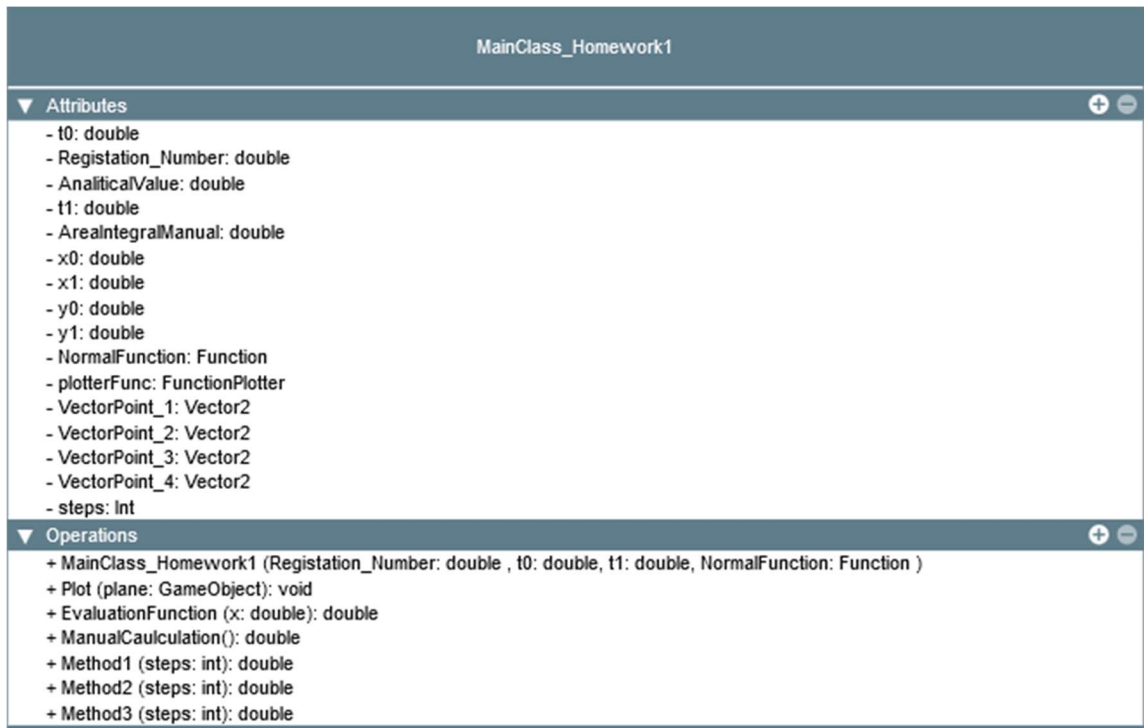


Figure 1 UML MainClass_Homework1

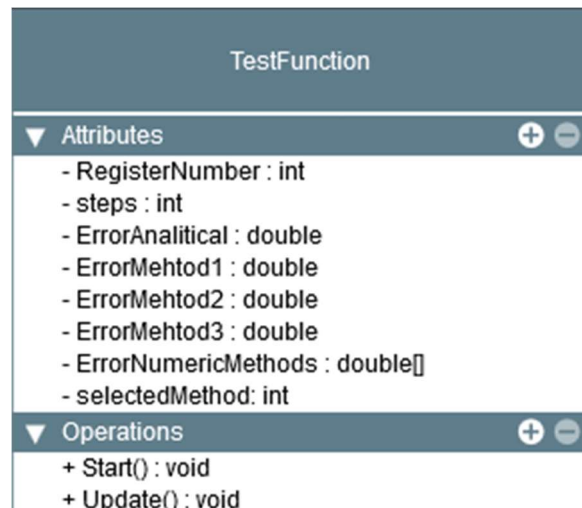


Figure 2 UML TestFunction

3. Function plots
 - Function plot.
 -

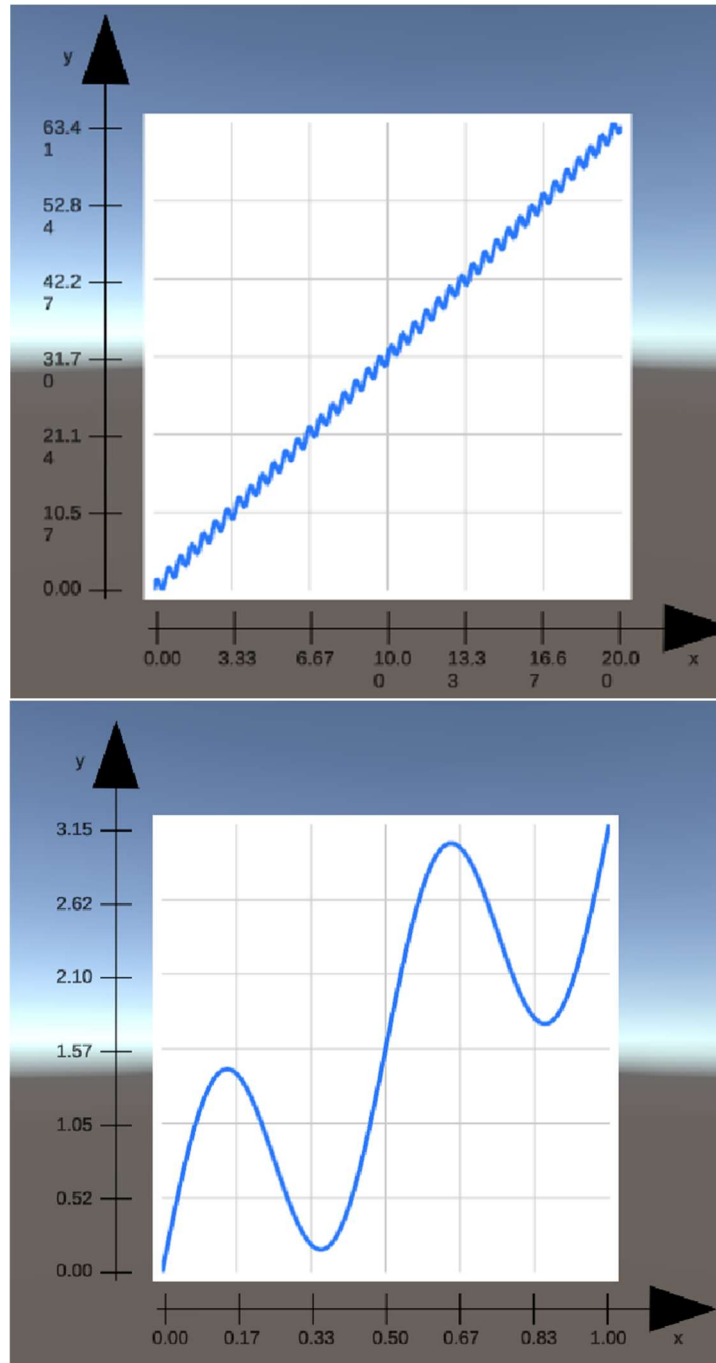


Figure 3 Function Plot

- Method 1.

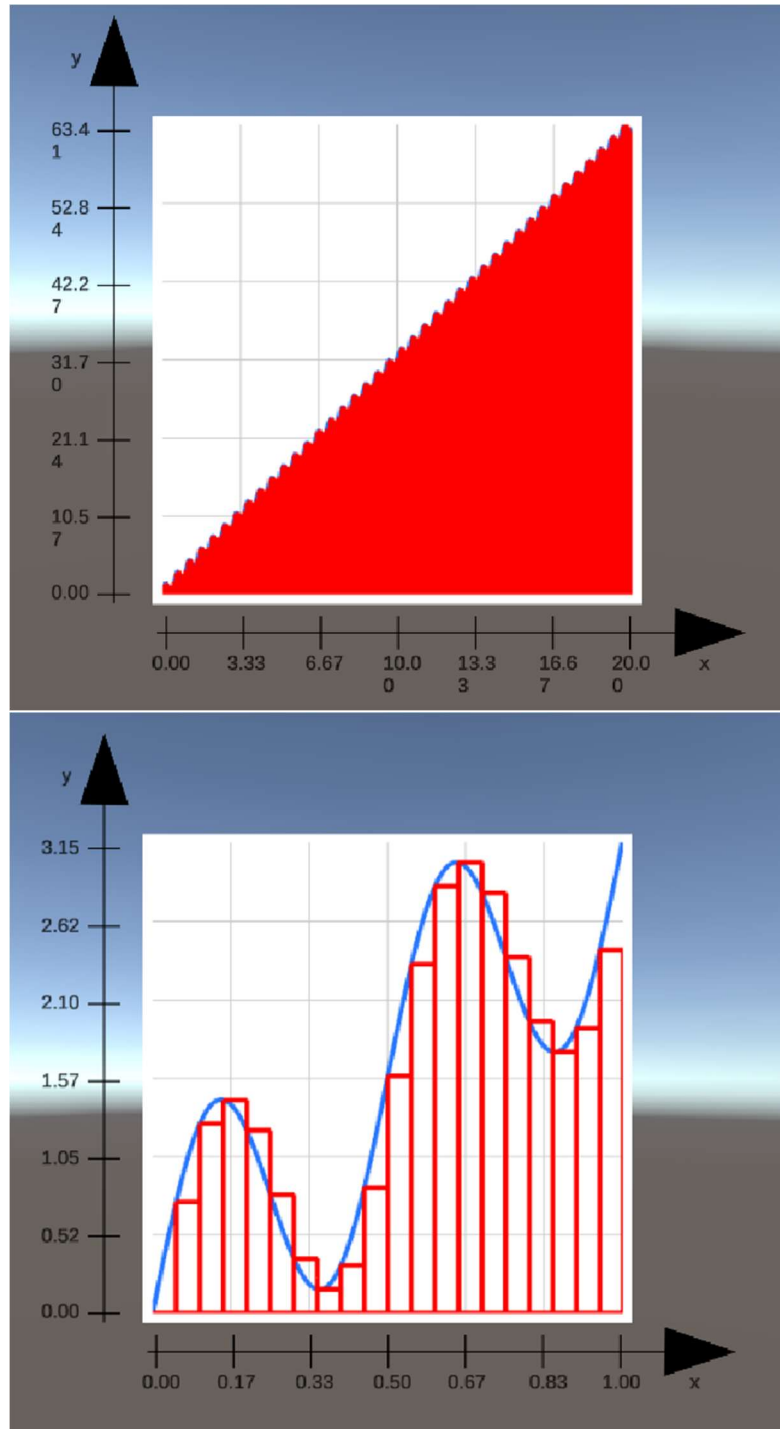


Figure 4 Graph method 1

- Method 2.

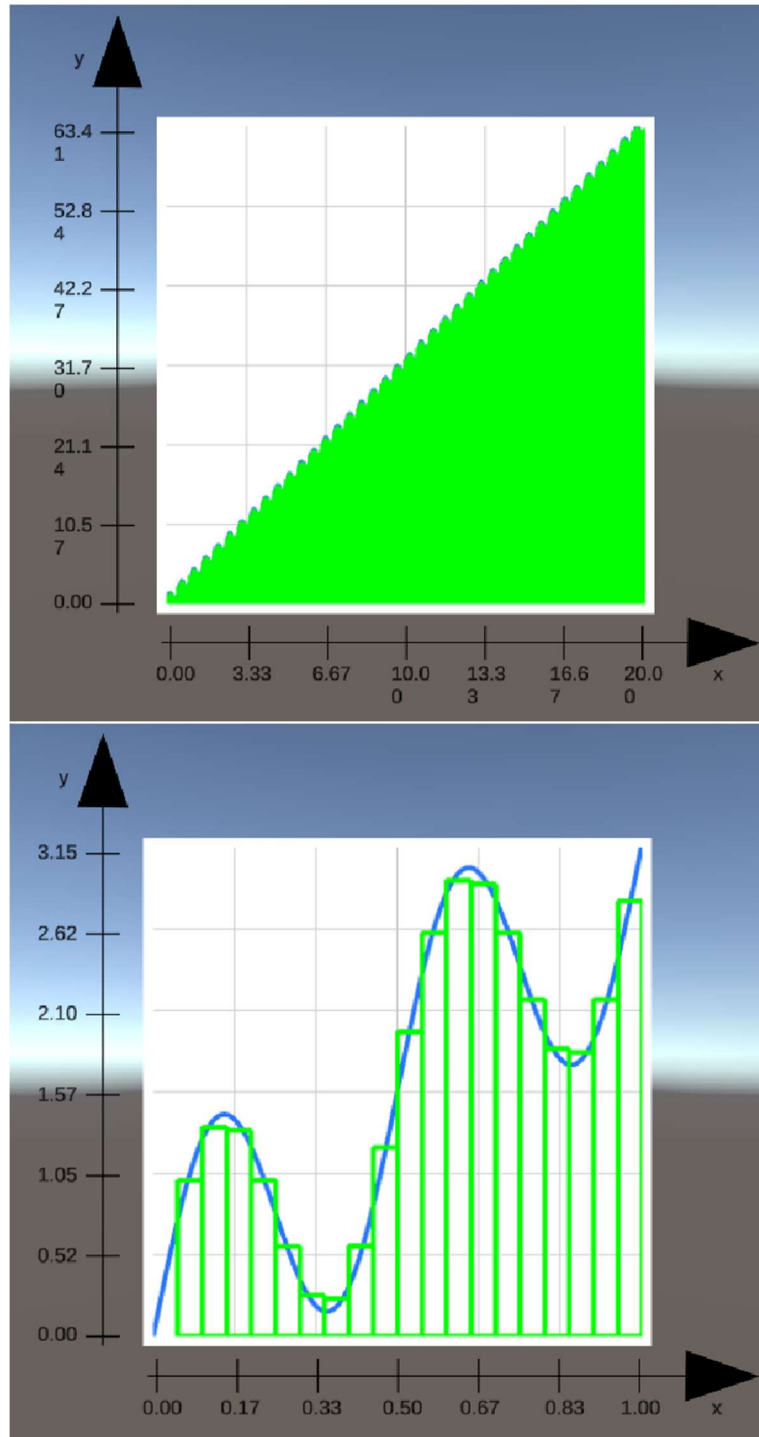


Figure 5 Graph method 2

- Method 3.

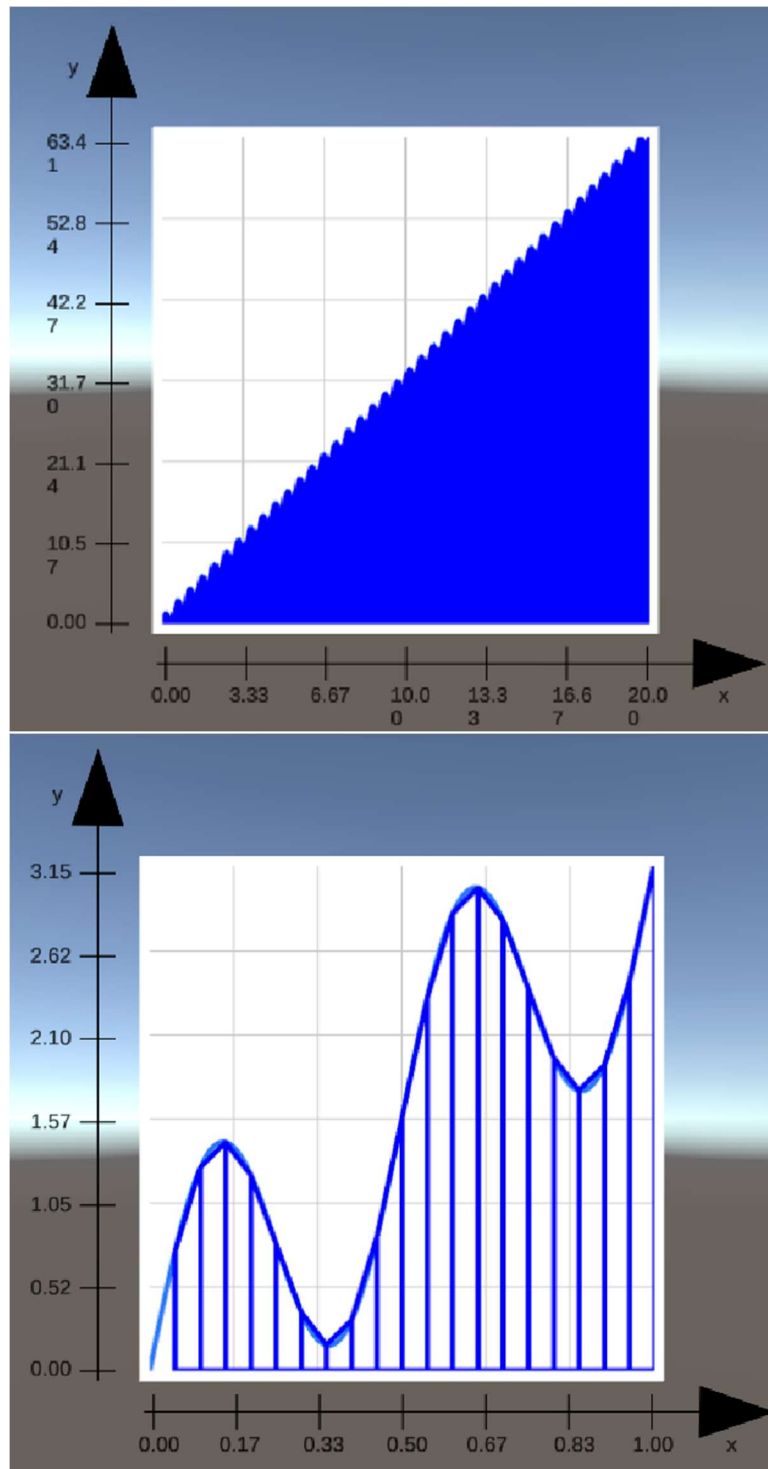


Figure 6 Graph method 3

4. Nassi – Schneiderman Diagrams

Method 1.

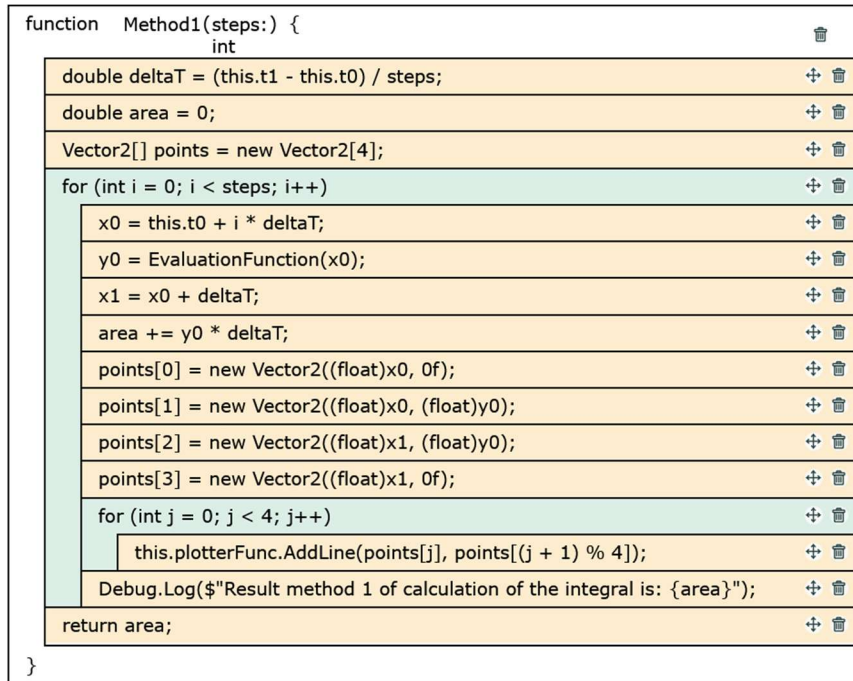


Figure 7 Nassi – Schneiderman Diagrams - Method 1

Method 2.

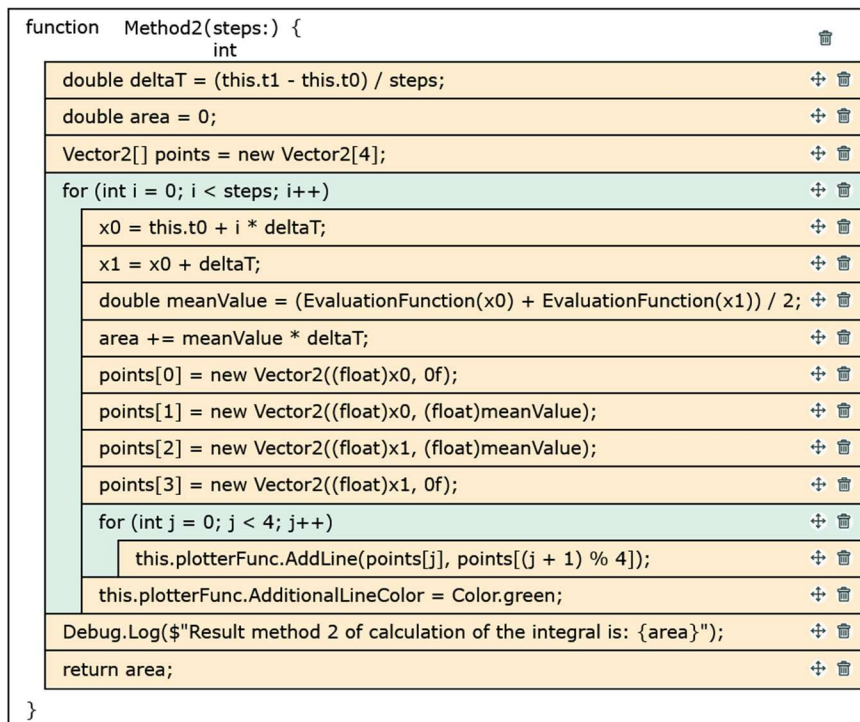


Figure 8 Nassi – Schneiderman Diagrams - Method 2

Method 3.

function Method3(steps:) {	🗑
int	
double deltaT = (this.t1 - this.t0) / steps;	✚ 🗑
double area = 0;	✚ 🗑
Vector2[] points = new Vector2[4];	✚ 🗑
for (int i = 0; i < steps; i++)	✚ 🗑
x0 = this.t0 + i * deltaT;	✚ 🗑
x1 = x0 + deltaT;	✚ 🗑
y0 = EvaluationFunction(x0);	✚ 🗑
y1 = EvaluationFunction(x1);	✚ 🗑
area += 0.5 * (y1 + y0) * deltaT;	✚ 🗑
points[0] = new Vector2((float)x0, 0f);	✚ 🗑
points[1] = new Vector2((float)x0, (float)y0);	✚ 🗑
points[2] = new Vector2((float)x1, (float)y1);	✚ 🗑
points[3] = new Vector2((float)x1, 0f);	✚ 🗑
for (int j = 0; j < 4; j++)	✚ 🗑
this.plotterFunc.AddLine(points[j], points[(j + 1) % 4]);	✚ 🗑
this.plotterFunc.AdditionalLineColor = Color.blue;	✚ 🗑
Debug.Log(\$"Result method 3 of calculation of the integral is: {area}");	✚ 🗑
return area;	✚ 🗑
}	

Figure 9 Nassi – Schneiderman Diagrams - Method 3

5. Results of the numerical and analytical methods.

the manual calculation of this integral is: 635.69

Result method 1 of calculation of the integral is: 632.51155

Result method 2 of calculation of the integral is: 635.626554924185

Result method 3 of calculation of the integral is: 635.626554924185

6. Result absolute error and relative error.

Absolute Error for Method 1 : 3.17844999999977

Relative Error for Method 1 : 0.00502512562814034

Absolute Error for Method 2 : 0.0634450758145704

Relative Error for Method 2 : 9.98150176751785E-05

Absolute Error for Method 3 : 0.0634450758145704

Relative Error for Method 3 : 9.98150176751785E-05