Integration using Trapezoid and Romberg Method

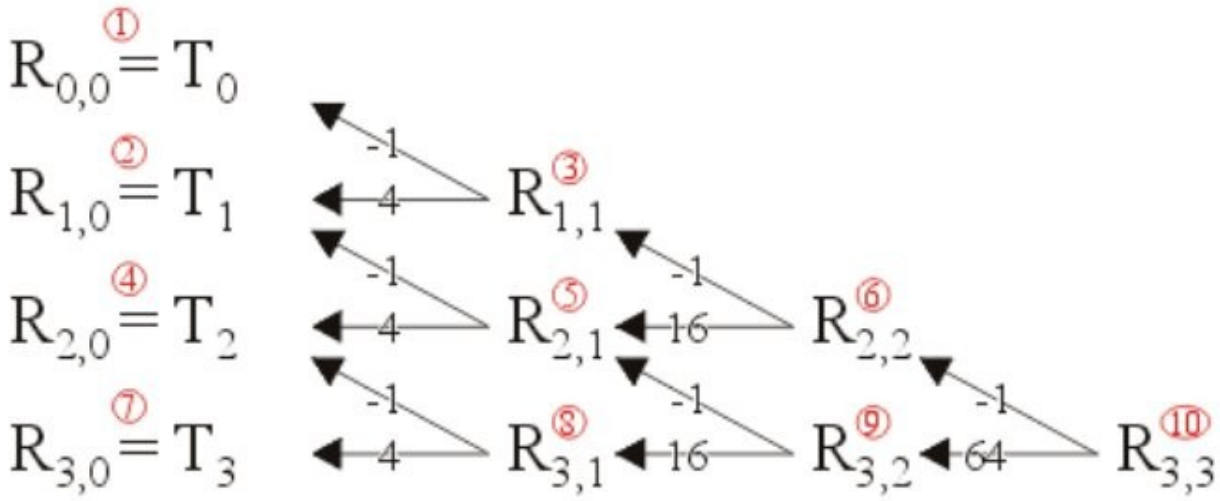

Figure 1. Calculating the Romberg approximations.

The above stated indices are consistent with the code in which,a matrix has been used , thus , indices begin from 0.

Furthermore, the submitted program is one order higher than what is depicted in the picture.

R(p,0) are equated to T(p) since Trapezoid method is being used to calculate them.

The program finds these values column wise,from left to right

$$R_k^i = \frac{4^i R_k^{i-1} - R_{k-1}^{i-1}}{4^i - 1}$$

This is the formula used to calculate the corrected intergrals of the higher order Romberg Method

The values of R[k][j] are :

R[0:4][0]	R[1:4][1]	R[2:4][2]	R[3:4][3]	R[4][4]
1.518745e-08				
1.518757e-08	1.518761e-08			
1.518746e-08	1.518742e-08	1.518740e-08		
1.518739e-08	1.518737e-08	1.518736e-08	1.518736e-08	
1.518755e-08	1.518761e-08	1.518762e-08	1.518763e-08	1.518763e-08

Taking steps of 16 units each (INPUT = 1)//Input is the number of substeps taken in a 16 unit Original Step

ROMBERG METHOD

% time	cumulative seconds	self seconds	calls	self Ts/call	total Ts/call	name
0.00	0.00	0.00	5	0.00	0.00	Trapezoid_Integrate
0.00	0.00	0.00	1	0.00	0.00	Romberg_Integrate

TRAPEZOID METHOD

% time	cumulative seconds	self seconds	calls	self Ts/call	total Ts/call	name
100.69	0.01	0.01				main
0.00	0.01	0.00	1	0.00	0.00	Trapezoid_Integrate

Taking steps of 8 units each (INPUT = 2) //Input is the number of substeps taken in a 16 unit Original Step

ROMBERG METHOD

% time	cumulative seconds	self seconds	calls	self ms/call	total ms/call	name
100.45	0.01	0.01				main
0.00	0.01	0.00	4	0.00	0.00	Trapezoid_Integrate
0.00	0.01	0.00	1	0.00	0.00	Romberg_Integrate

TRAPEZOID METHOD

% time	cumulative seconds	self seconds	calls	self Ts/call	total Ts/call	name
100.71	0.01	0.01				main
0.00	0.01	0.00	1	0.00	0.00	Trapezoid_Integrate

Taking steps of 4 units each (INPUT = 4) //Input is the number of substeps taken in a 16 unit Original Step

ROMBERG METHOD

% time	cumulative seconds	self seconds	calls	self Ts/call	total Ts/call	name
0.00	0.00	0.00	3	0.00	0.00	Trapezoid_Integrate
0.00	0.00	0.00	1	0.00	0.00	Romberg_Integrate

TRAPEZOID METHOD

% time	cumulative seconds	self seconds	calls	self Ts/call	total Ts/call	name
100.69	0.02	0.02				main
0.00	0.02	0.00	1	0.00	0.00	Trapezoid_Integrate

Taking steps of 2 units each (INPUT = 8) //Input is the number of substeps taken in a 16 unit Original Step

ROMBERG METHOD

% time	cumulative seconds	self seconds	calls	self Ts/call	total Ts/call	name
50.22	0.01	0.01				main
0.00	0.01	0.00	1	0.00	10.04	Romberg_Integrate
50.22	0.02	0.01	2	5.02	5.02	Trapezoid_Integrate

TRAPEZOID METHOD

% time	cumulative seconds	self seconds	calls	self Ts/call	total Ts/call	name
100.69	0.01	0.01				main
0.00	0.01	0.00	1	0.00	0.00	Trapezoid_Integrate

Taking steps of 1 unit each (INPUT = 16) //Input is the number of substeps taken in a 16 unit Original Step

ROMBERG METHOD

% time	cumulative seconds	self seconds	calls	self Ts/call	total Ts/call	name
100.45	0.01	0.01				main
0.00	0.01	0.00	1	0.00	0.00	Romberg_Integrate
0.00	0.01	0.00	1	0.00	0.00	Trapezoid_Integrate

TRAPEZOID METHOD

% time	cumulative seconds	self seconds	calls	self Ts/call	total Ts/call	name
100.71	0.02	0.02				main
0.00	0.02	0.00	1	0.00	0.00	Trapezoid_Integrate

The time taken for the program to run is negligible for such a small dataset (32,768 points).

We can compare the time complexities of the methods in order to get the insight.
Taking the domain of integration from 'a' to 'b', and dividing it into 'n' parts,
The Time Complexity of Trapezoid Method is O(n).
In case of Romberg Method of Order 'p',The Trapezoid Method is called 'p' times with domain being divided in 'pk' parts 'k'th time
Thus the Time Complexity due to calling of the Trapezoid Method is O(p1+p2+....+pk)
The Time Complexity without the Trapezoid Method is O(ln(p)),which is significantly less than that for Trapezoid Method.

In Conclusion,
The Romberg Method is efficient if we do NOT use a high-level Method to calculate the initial integrals R[k][0].
Then the effect of correction by Romberg Method is substantial, with a lower time complexity.