FCM II Project 3

Chase Wiederstein

March 12th, 2024

1 Statement of the Problem

For this project, we are tasked with implementing three composite quadrature methods: trapezoidal rule, Simpson's rule, and midpoint rule. For each method, I will be using a uniform interval size of $H_m = \frac{b-a}{m}$. The integral under inspection is:

$$\int_0^3 e^x dx = e^3 - 1 \approx 19.08553692$$

For the midpoint rule, I will also be implementing a global uniform subinterval size refinement with a factor $\alpha = 1/k$ such that $h_{fine} = \alpha H_{coarse}$. Note that I will be using $\alpha = 1/3$ for the composite midpoint rule and $\alpha = 1/2$. This is done so that there is a complete reuse of the function evaluations on the coarse grid. I will then compare the performance and accuracy of each of the methods.

2 Description of the Algorithms and Implementation

Note that for each rule used, i will be using two stopping conditions of $tol = 10^{-2}$ and $tol = 10^{-4}$. I stop each algorithm once $|I - \tilde{I}| < tol$, where I is the true value of $e^3 - 1 \approx 19.08553692$, and $\tilde{I}$ is the integral approximated by the quadrature rule. I also use a variable to count the number of function evaluations as I implement each algorithm. For each rule, $H_m = \frac{b-a}{m}$ where $m$is increased each iteration.

2.1 Composite Trapezoidal Rule

For the composite midpoint rule, I will be using the following algorithm:

$$I_m^{ctr} = H_m/2(f_0 + f_m + 2\sum_{i=1}^{m-1} f_i) \quad \text{where} \quad x_i = a + iH_m$$

The global refinement for composite trapezoidal rule goes a step further and uses $\alpha = 1/2$ since this method uses two points on each interval. We have complete reuse of previous function evaluations in order to generate a fine grid of length $2m$. Thus, we include the following as a part of our algorithm:

$$I_{2m}^{ctr} = 1/2(I_m^{ctr} + H_m \sum_{m \text{ new points}} f_i)$$

2.2 Composite Midpoint Rule

For the composite midpoint rule, I use the following algorithm:

$$I_m^{cmp} = H_m \sum_{i=1}^{m-1} f_i \quad \text{where} \quad x_i = a + iH_m + H_m/2$$

For this method, I will be using the size refinement factor $\alpha = 1/3$ including the following:

$$I_{3m} = 1/3(I_m^{cmp} + H_m \sum_{i=0}^{m-1} (f(a + iH_m + H_m/6) + f(a + iH_m + 5H_m/6)))$$

2.3 Composite Simpson's Rule

For this method, there is no global refinement algorithm implemented and we will just use the following:

$$I_m^{csf} = H_m/6(f(a) + f(b) + 2\sum_{i=1}^{m-1} f(b_i) + 4\sum_{i=1}^{m} f(c_i)) \quad \text{where} \quad b_i = a + iH_m \quad c_i = a + (i + .5)H_m$$

3 Description of the Experimental Design and Results

3.1 Task I:

As mentioned, the stopping criteria that I employed for each method was $|I - \tilde{I}| < tol$ for the tolerance of 10^{-2} and 10^{-4}. I numerically calculated m using these tolerances by increasing m until convergence. Before numerically calculating, I manually calculated to solve for m using each methods corresponding theoretical error bounds with:

$$E_m^{ctr} = -(b-a)\frac{H_m^2}{12}f'', \quad E_m^{cmp} = (b-a)\frac{h_m^2}{12}f'', \quad E_m^{csf} = -(b-a)\frac{h_m^4}{2880}f^{(4)}$$

The results of the theoretical m values can be found in table 1 in section 5. Using these values, we can assume that Simpson's will outperform the others in terms of rate of convergence. Next, I put this hypothesis to the test by numerically solving for the m values which can be seen in tables 2 and 3 in section 5. Each of the numerically evaluated m values are lesser than the theoretical bounds and Simpson's is the lowest and is closest to its theoretical value due to its higher order. I have also included the error using the exact value of I and the amount of function evaluations.

3.2 Task II:

For this task, we were to analyze the global refinement methods for the composite midpoint and composite trapezoidal methods. For the composite midpoint rule with global refinement, I tested $m = 3^k$ values, and for the composite trapezoidal rule with global refinement, I used values of $m = 2^k$ for $k = 0, 1, 2, ...10$. The results are in tables 4 and 5 in section 5. One of the main goals of this task is to show that both the composite midpoint and composite trapezoidal methods are both of order 2. We do this by showing:

$$r_{cmr} \approx log_3(\frac{I_{3m}^{cmr} - I_m^{cmr}}{I - I_{3m}} + 1) \quad \text{and} \quad r_{ctr} \approx log_2(\frac{I_{2m}^{ctr} - I_m^{ctr}}{I - I_{2m}} + 1)$$

As m increases, each method does approach the values of $r \approx 2$. However, the composite midpoint rule takes much longer and is less efficient as compared to the composite trapezoidal rule due to the need to evaluate more points to reach its tolerance.

4 Conclusion

With the given data of each method, it is clear to see that Simpson's outperforms the other methods in terms of accuracy and efficiency. As with the global refinement methods, the trapezoidal method outperforms the midpoint method in terms of efficiency. Out of each of the methods, not utilizing the composite midpoint method would be a wise choice when deciding upon a quadrature method. For this project, we also use an analytic function that is infinitely differentiable. Other functions would behave much differently and perhaps require a deeper analysis of which quadrature method to use.

5 Tables

Table 1: Theoretical m values

Method	10^{-2}	10^{-4}
Composite Trapezoidal Rule	68	673
Composite Midpoint Rule	48	476
Composite Simpson's Rule	4	12

Table 2: Results for error of 10^{-2}

Method	m value	Function Evaluations	Error
Composite Trapezoidal Rule	38	39	0.00991182
Composite Midpoint Rule	27	27	0.00981413
Composite Simpson's Rule	3	7	0.00643474

Table 3: Results for error of 10^{-4}

Method	m value	Function Evaluations	Error
Composite Trapezoidal Rule	379	380	$9.96522e-05$
Composite Midpoint Rule	268	268	$9.96471e-05$
Composite Simpson's Rule	9	19	$8.15441e-05$

Table 4: Results for Global Refinement Composite Trapezoidal Rule

m	Error	Function Evaluations	r
1	12.5428	2	NA
2	3.45115	3	1.86171
4	0.886358	5	1.96112
8	0.223136	9	1.98997
16	0.0558819,	17	1.99747
32	0.0139766	33	1.99937
64	0.00349454	65	1.99984
128	0.000873659	129	1.99996
256	0.000218416	257	1.99999
512	$5.46041e-05$	513	2
1024	$1.3651e-05$	1024	2

Table 5: Results for Global Refinement Composite Midpoint Rule

m	Error	Function Evaluations	r
1	0.772633	2	1.80948
3	0.0880735	4	1.97671
9	0.00981413	10	1.99738
27	0.00109081	28	1.99971
81	0.000121205	82	1.99997
243	$1.34673e-05$	244	2
729	$1.49637e-06$	730	2
2187	$1.66263e-07$	2188	2
6561	$1.84737e-08$	6562	2
19683	$2.05278e-09$	19683	1.99993
59049	$2.28148e-10$	59050	1.99976

k	$m = 3^k$	r
0	1	1.109548370510806
1	3	1.809481808599895
2	9	1.809481808591895
3	27	1.976705077739497
4	81	1.976705077739497
5	243	1.997381717263836
6	729	1.997381717263836
7	$2,187$	1.999708703907611
8	$6,561$	1.999708703907611
9	$19,683$	1.999967629107464
10	$59,049$	2