Programming Assignment 7

Jenny Petrova

April 2025

1 Executive Summary

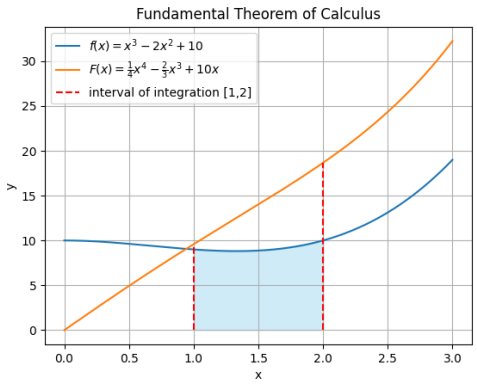
This report discusses composite quadrature methods for numerical integration. The Newton-Cotes methods, and their corresponding error error expressions, are derived for one-dimensional integrals over bounded (open and closed) intervals. The methods are then constructed in a Python program, and evaluated for several functions over a range of chosen intervals. Comparisons are made between the observed behavior and theoretical predictions of these techniques, and remarks are given regarding the overall accuracy of the approximations.

2 Statement of the Problem

Given a real-valued continuous function f on an interval $[a, b]$, the Fundamental Theorem of Calculus defines the integration of f on $[a, b]$ by

$$\int_a^b f(x)dx = F(b) - F(a), \tag{1}$$

where F is the anti-derivative of f on $[a, b]$. This theorem defines the method for finding the area beneath the function $f(x)$ between points $x = a$ and $x = b$, illustrated by the following example:



Now, suppose f is *not* a continuous function. We still wish to compute the integral over a defined interval, but more rigorous methods must be applied. Assume the value of the function f, defined on $[a, b]$, is known at $n + 1$ equally spaces nodes on $[a, b]$, such that $a = x_0 < x_1 < \cdots < x_n = b$. The **Newton-Cotes quadrature rules** can be applied to evaluate the integrand at the equally spaced nodes. The Newton-Cotes formulas are defined by

$$\int_a^b f(x)dx = \sum_{i=0}^{n} w_i f(x_i), \tag{2}$$

where

$$x_i = a + ih, \quad h = \frac{b-a}{n} \quad \text{(closed intervals)},$$
$$x_i = a + (i+1)h, \quad h = \frac{b-a}{n+2} \quad \text{(open intervals)}.$$

However, for wider intervals $[a, b]$ and more complex functions (with varying curvature and/or discontinuities), the Newton-cotes numerical integration methods can result in large approximation errors. To maintain a high degree of accuracy and efficiency in the approximation evaluation, we can partition the global interval $[a, b]$ into smaller sub-intervals, and apply the Newton-cotes formulas to each sub-interval. This is known as **Composite Newton-Cotes quadrature**, and is the subject of the derivations that follow.

3 Description of the Mathematics

Let $f(x) \in \mathcal{C}[a, b]$ and $f_n(x) \approx f(x)$, for $n+1$ mesh points. Numerical quadrature approximates the definite integral

$$I_n(f; a, b) \approx \int_a^b f(x)dx = \mathcal{I}(f; a, b). \tag{3}$$

Composite quadrature methods partition the interval $[a, b]$ into N sub-intervals $[a_i, b_i]$ such that

$$\mathcal{I}(f; a, b) \approx I_{comp}(f; a, b) = \sum_{i=1}^{N} I_{n_i}(f; a_i, b_i) = \sum_{i=1}^{N} \left(\int_{a_i}^{b_i} f_{n_i}(x)dx \right), \tag{4}$$

where f_{n_i} is the approximation of $f(x)$ on $[a_i, b_i]$, meaning $f_n(x)$ is a piecewise interpolatory approximation of $f(x)$. Therefore we consider $f_{n,i}(x) \approx p_{d,i}(x)$ to be an interpolatory polynomial of degree d on a local mesh with $d + 1$ points. The methods are applied to a uniform mesh, where $H = \frac{b-a}{N}$ defines the sub-intervals $[a_i, b_i] = [a + (i-1)H,\, a + iH]$ for $i = 1, 2, \ldots, N$. Since the local meshes may be open or closed, the endpoints a_i and b_i may or may not be used as interpolation points of $p_{d,i}(x)$. Since $\mathcal{I}(f)$ denotes the exact definite integral

on $[a, b]$ and $I_n(f)$ denotes the numerical approximation of $f(x)$ evaluated at $n+1$ points, we need to define an additional term to represent the error in the approximation. Let $h = b - a$. Taking the Taylor expansion of $\mathcal{I}(f)$ and $I_n(f)$ around a reference point $x_i \in [a, b]$ and comparing the terms, we can define the **Newton-Cotes Error** term to be of the form

$$E_n(f) = \mathcal{I}(f) - I_n(f) = C_n H^m f^{(d+i)} + O(H^{m+1}), \tag{5}$$

where m is the order of the method and d is the degree of exactness. Summing the error terms E_i over N sub-intervals $[a_i, b_i]$, $i = 1, 2, \ldots, N$, we arrive at the expression for the **Composite Newton-Cotes Error**:

$$E = \sum_{i=1}^{N} E_i = C_N (b - a) H^d f^{(d)} + O(H^{d+1}), \tag{6}$$

where d depends on the method and the value of n (where $n+1$ are the number of points used in each sub-interval):

$$E_{n,N}(f) = \begin{cases} C_{n,N} H^{n+2} f^{(n+2)} + O(H^{n+3}) & n \text{ even} \\ \tilde{C}_{n,N} H^{n+1} f^{(n+1)} + O(H^{n+2}) & n \text{ odd} \end{cases} \tag{7}$$

We discuss several composite quadrature methods (for both open and closed intervals) and their corresponding error expressions below. Detailed derivations of several error expressions are located in the appendix (6).

3.1 Closed Composite Numerical Quadrature

The following composite quadrature methods are evaluated on closed sub-intervals $[a_i, b_i]$ of the global interval $[a, b]$, where the endpoints of the sub-intervals are interpolation nodes used in the approximation.

3.1.1 Composite Left Rectangle Rule

This first method comes from the Riemann Sum approximation of integrals by a finite sum. The function $f(x)$ is approximated by partitioning the integral $[a, b]$ into N sub-intervals of equal length $H = \frac{b-a}{N}$. We take the values at the left endpoints, and calculate the area under $f(x)$ as a sum of rectangles with base (interval width) H and height $f(x_i)$, where $x_i = a + iH$. The **Composite Left Rectangle Rule** is defined by

$$I_{clr}(f) = \sum_{i=0}^{N-1} H f(x_i) = H \sum_{i=0}^{N-1} f_i, \tag{8}$$

and the **Composite Left Rectangle Error** is

$$E_{clr}(f) = \frac{1}{2}(b - a) H f'(\xi) + O(H^2). \tag{9}$$

3.1.2 Composite Newton-Cotes Trapezoidal Rule

This next method is also derived from Riemann Sum approximations, where we approximate the region under a graph $f(x)$ by calculating the area of each sub-interval as a trapezoid, such that

$$\int_{a_i}^{b_i} f(x)dx \approx \frac{b_i - a_i}{2}(f(a_i) + f(b_i)). \tag{10}$$

Let $H = \frac{b-a}{N}$, for N sub-intervals. Summing the resulting areas between each set of sub-interval endpoints defines the **Composite Trapezoidal Rule** to be

$$\begin{aligned} I_{ctr}(f) =& \frac{H}{2}[f(a) + f(a+H)] + \frac{H}{2}[f(a+H) + f(a+2H)] \\ &+ \cdots + \frac{H}{2}[f(a+(N-1)H) + f(a+NH)], \end{aligned}$$

where $a + NH = a + (b - a) = b$. Let $x_i = a + iH$. The above simplifies to

$$I_{ctr}(f) = \frac{H}{2}\left[f(a) + f(b) + 2\sum_{i=1}^{N-1} f(x_i)\right]. \tag{11}$$

The **Composite Trapezoidal Error** is

$$E_{ctr}(f) = -\frac{1}{12}(b-a)H^2 f''(\xi) + O(H^3). \tag{12}$$

3.1.3 Composite Newton-Cotes Simpson's First Rule

We construct a local quadratic interpolation on each local mesh, i.e. $d = 2$, so we require the use of $d + 1 = 3$ interpolation nodes on each sub-interval. We interpolate the endpoints and the midpoint of each interval. Let $H = \frac{b-a}{2}$ and define $H_{mid} = \frac{H}{2}$. Let the midpoint of each sub-interval $[a_i, b_i]$ be defined by $c_i = a_i + H_{mid}$. The **Composite Simpson's First Rule** is given by

$$I_{csfr}(f) = \sum_{i=0}^{N}\left(\int_{a_i}^{b_i} p_{n,i}(x)dx\right) = \sum_{i=0}^{N}\left(\frac{H_{mid}}{3}\left[f(a_i) + 4f(c_i) + f(b_i)\right]\right)$$

which simplifies to

$$I_{csfr}(f) = \frac{H}{6}\left[f(a) + f(b) + 2\sum_{i=0}^{N-1} f(b_i) + 4\sum_{i=1}^{N} f(c_i)\right], \tag{13}$$

with corresponding **Composite Simpson's First Rule Error** defined to be

$$E_{csfr}(f) = -\frac{1}{90}(b-a)H^4 f^{(4)}(\xi) + O(H^5). \tag{14}$$

3.2 Open Composite Numerical Quadrature

The following composite quadrature methods are evaluated on open partitions (a_i, b_i) of the global interval $[a, b]$.

3.2.1 Composite Newton-Cotes Midpoint Rule

We again assume $[a, b]$ consists of uniformly spaced nodes such that $H = \frac{b-a}{N}$ denotes the width of each sub-interval (a_i, b_i), $i = 1, 2, \ldots, N$. For this method, we use a constant interpolation $p_{0,i}(x)$ on each open local mesh consisting of one point $x_i \in (a_i, b_i)$, where x_i is taken to be the midpoint such that

$$x_i = a_i + \frac{H}{2} = a_i + H_{mid}. \tag{15}$$

The local mesh is open and the degree $d = 0$, therefore the constant piecewise polynomial $f_n(x) \approx f(x)$ is not continuous. The **Composite Midpoint Rule** is

$$I_{cmpr}(f) = \sum_{i=1}^{N} \left(\int_{a_i}^{b_i} p_{0,i}(x)dx \right) = \sum_{i=1}^{N} \left(\int_{a_i}^{b_i} f(x_i)dx \right).$$

This simplifies to

$$I_{cmpr}(f) = H \sum_{i=1}^{N} f(x_i). \tag{16}$$

Note that the construction of this formula resembles the Left Rectangle Rule we defined earlier (8). However, there exist important distinctions between these formulations. For the (open) Composite Midpoint rule, we evaluate the function at the midpoint of each sub-interval, where $x_i = a_i + H_{mid}$. For the (closed) Composite Left Rectangle rule, we evaluate the function at the left endpoint of each sub-interval, i.e. $x_i = a_i$. The difference in the chosen evaluation nodes is best demonstrated in the differing error expressions. The **Composite Midpoint Error** is defined to be

$$E_{cmpr}(f) = \frac{1}{24}(b-a)H^2 f''(\xi) + O(H^3). \tag{17}$$

which differs in degree and complexity from the Composite Left Rectangle error expression (9).

3.2.2 Composite Newton-Cotes Two-Point Rule

Following from the midpoint rule, we wish to construct a linear interpolation $p_{1,i}(x)$ on each open local mesh (a_i, b_i). We require two equidistant points for the interpolation, so we take

$$x_{i_1} = a_i + \frac{H}{3},$$
$$x_{i_2} = a_i + \frac{2H}{3},$$

where $H = \frac{b-a}{N}$. Therefore the **Composite Newton-Cotes Two-Point Rule**, constructed by

$$I_{c2pr}(f) = \sum_{i=1}^{N} \left(\int_{a_i}^{b_i} p_{1,i}(x)dx \right) = \sum_{i=1}^{N} \left(\int_{a_i}^{b_i} f(x_i)dx \right),$$

simplifies to

$$I_{c2pr}(f) = \frac{H}{2} \sum_{i=1}^{N} [f(x_{i_1}) + f(x_{i_2})]. \tag{18}$$

The corresponding **Composite Newton-Cotes Two-Point Error** is

$$E_{c2pr}(f) = \frac{1}{36}(b-a)H^2 f''(\xi) + O(H^3). \tag{19}$$

3.2.3 Composite Two-Point Gauss-Legendre Method

The n-point **Gauss-Legendre quadrature rule** for approximating the definite integral of a function taken over the interval $[-1, 1]$ is defined to be

$$I_{gl}(f) = \int_{-1}^{1} f(x)dx \approx \int_{-1}^{1} p_n(x)dx = \sum_{i=1}^{n} \gamma_i f(x_i), \tag{20}$$

where n is the number of points required, γ_j are the defined quadrature weights, x_j are the roots of the $(n+1)^{st}$ Legendre polynomial, and the evaluation has $2n+1$ degrees of exactness. For $n = 2$ points, the defined weights are $\gamma_j = 1$ and the roots of the quadratic Legendre polynomial are $x_j = \pm\frac{1}{\sqrt{3}}$, where $j = 1, 2$. Therefore, the formula for the Gauss-Legendre method is simply

$$\int_{-1}^{1} f(x)dx = f\left(\frac{1}{\sqrt{3}}\right) + f\left(\frac{-1}{\sqrt{3}}\right). \tag{21}$$

To evaluate the definite integral over an interval $[a, b]$, i.e. to evaluate $\int_a^b f(x)dx$, a change of measure must be applied. Let $x \in [a, b]$ and $t \in [-1, 1]$. The variable z can be mapped to x by the following formula:

$$x = \frac{b-a}{2}t + \frac{b+a}{2}. \tag{22}$$

Therefore

$$dx = \frac{b-a}{2}dt \quad \Rightarrow \quad \frac{dx}{dt} = \frac{b-a}{2}, \tag{23}$$

and we can take the integral to be

$$I_{gl}(f) = \int_a^b f(x)dx = \int_{-1}^{1} f\left(\frac{b-a}{2}t + \frac{b+a}{2}\right)\frac{dt}{dx}dx \tag{24}$$

$$= \frac{b-a}{2}\int_{-1}^{1} f\left(\frac{b-a}{2}t + \frac{b+a}{2}\right)dt \tag{25}$$

$$\approx \frac{b-a}{2}\sum_{j=0}^{n} \gamma_j f\left(\frac{b-a}{2}t_j + \frac{b+a}{2}\right) \tag{26}$$

for the general n-point Gauss-Legendre quadrature rule. Choosing $n = 2$, the above simplifies to

$$I_{gl}(f) = \frac{b-a}{2}\left[f\left(\frac{(b-a)}{2\sqrt{3}} + \frac{b+a}{2}\right) + f\left(\frac{-(b-a)}{2\sqrt{3}} + \frac{b+a}{2}\right)\right], \tag{27}$$

where we set the weights $\gamma_j = 1$ and roots $t_j = \pm\frac{1}{\sqrt{3}}$, $j = 1, 2$, to be mapped from $[-1, 1]$ to the chosen interval $[a, b]$ (21). The above expression is extended to derive the **Composite Two-Point Gauss-Legendre Method**. Suppose we again have a set of uniformly distributed nodes $\{x_i\}_{i=1}^{N}$ over an interval $[a, b]$. We partition the nodes into a set of N open sub-intervals (a_i, b_i) of equal spacing $H = \frac{b-a}{N}$, such that $(a_i, b_i) = (a+(i-1)H,\ a+iH)$ for $i = 1, 2, \ldots, N$. We map the roots $t_j = \pm\frac{1}{\sqrt{3}}$ to each interval (a_i, b_i), and denote $x_j = \frac{b_i-a_i}{2}t_j + \frac{b_i+a_i}{2}$ for $j = 1, 2$. Taking formula (27) and summing the value over each sub-interval gives

$$I_{cgl}(f) = \frac{H}{2}\sum_{i=1}^{N}\left[f(x_1) + f(x_2)\right]. \tag{28}$$

The corresponding error expression is

$$E_{cgl} = \frac{1}{270}(b-a)f^{(4)}(\eta), \tag{29}$$

which does not have an H term, meaning the error expression can be computed exactly for a known function $f(x)$, and is therefore independent of the sub-interval sizes H.

4 Description of the Methods, Algorithms and Implementation

The discussed composite quadrature methods are constructed in Python 3.9.6. on the PyCharm IDE. The relevant pseudocode is presented and the structure of these routines is discussed below.

Algorithm 1 Composite_Newton_Cotes(a, b, N, f, n_points, $closed = True$)

$H \leftarrow (b - a)/N$
$sum \leftarrow 0$
if *closed* **then**
    if $n\_points = 1$ **then** ▷ Left Rectangle Rule
        for $i \leftarrow 0$ **to** $N - 1$ **do**
            $sum \leftarrow sum + f(a + i \cdot H)$
        end for
        return $H \cdot sum$
    else if $n\_points = 2$ **then** ▷ Trapezoidal Rule
        for $i \leftarrow 1$ **to** $N - 1$ **do**
            $sum \leftarrow sum + f(a + i \cdot H)$
        end for
        $term \leftarrow f(a) + f(b) + 2 \cdot sum$
        return $(H/2) \cdot term$
    else if $n\_points = 3$ **then** ▷ Simpson's Rule
        $sum2 \leftarrow 0$
        for $i \leftarrow 1$ **to** $N - 1$ **do**
            if i is even **then**
                $sum \leftarrow sum + f(a + i \cdot H)$
            else
                $sum2 \leftarrow sum2 + f(a + i \cdot H)$
            end if
        end for
        $term \leftarrow f(a) + f(b) + 2 \cdot sum + 4 \cdot sum2$
        return $(H/3) \cdot term$
    end if
else ▷ Open Newton-Cotes Rules
    if $n\_points = 1$ **then** ▷ Midpoint Rule
        $k \leftarrow 1/2$
        for $i \leftarrow 0$ **to** $N - 1$ **do**
            $sum \leftarrow sum + f(a + (i + k) \cdot H)$
        end for
        return $H \cdot sum$
    else if $n\_points = 2$ **then** ▷ Two-Point Rule
        $k \leftarrow 1/3$
        for $i \leftarrow 0$ **to** $N - 1$ **do**
            $x_1 \leftarrow a + (i + k) \cdot H$
            $x_2 \leftarrow a + (i + 2k) \cdot H$
            $sum \leftarrow sum + f(x_1) + f(x_2)$
        end for
        return $(H/2) \cdot sum$
    end if
end if

The above algorithm runs each of the discussed Composite Newton-Cotes quadrature methods (the Left Rectangle, Trapezoidal, Simpson's, Midpoint, and Two-Point Rules). The routine parameters take in the integral parameters (i.e. the endpoints of the global interval $[a, b]$), the number of sub-intervals, N, and our chosen, pre-defined function $f(x)$. We also input the number of points ('n_points') we want to compute on each interval. The parameter 'closed' is set to 'True', meaning the routine will assume a closed interval. We set the parameter to 'False' when we want to run an open interval method, for the chosen number of interval points. The routine returns the composite quadrature approximation, $I_n(f)$.

Additionally, we create separate routines for the Composite Midpoint and Trapezoidal rules, where we implement an efficient adaptive step refinement to improve the results of the approximation. The adaptive Midpoint routine uses a size refinement factor of $\alpha = \frac{1}{3}$. Consider the following example: suppose we have an interval $[1, 3]$, which we wish to divide into $N = 2$ equal sub-intervals, where $H = \frac{3-1}{2} = 1$. The sub-intervals are $[1, 2]$ and $[2, 3]$. We evaluate the function at the midpoint of each sub-interval, such that

$$I(f) = (1)\left[f\left(1 + \frac{1}{2}\right) + f\left(2 + \frac{1}{2}\right)\right].$$

Now consider the refinement factor, $\alpha = \frac{1}{3}$. We partition each sub-interval into three *new* sub-intervals: $[1, 1 + \frac{1}{3}]$, $[1 + \frac{1}{3}, 1 + \frac{2}{3}]$,$[1 + \frac{2}{3}, 2]$, $\ldots$, $[2 + \frac{2}{3}, 3]$. We now have $N = 2 * 3 = 6$ total sub-intervals with $H = 1 * \frac{1}{3} = \frac{1}{3}$. The midpoint approximation becomes

$$\begin{aligned} I(f) = \left(\frac{1}{3}\right)\Bigg[f\left(1 + \frac{1}{6}\right) + f\left(1 + \frac{1}{2}\right) + f\left(1 + \frac{5}{6}\right) \\ = f\left(2 + \frac{1}{6}\right) + f\left(2 + \frac{1}{2}\right) + f\left(2 + \frac{5}{6}\right)\Bigg]. \end{aligned}$$

Notice that the evaluations from the initial approximation, $f\left(1 + \frac{1}{2}\right)$ and $f\left(2 + \frac{1}{2}\right)$, are completely reused in the refinement step. We construct an efficient routine that iteratively refines each sub-interval and reevaluates the approximation, while also reusing points that have already been computed in the previous iteration.

Algorithm 2 Adaptive_midpoint(a, b, N, f, $tol = 10^{-6}$, $maxiter = 20$)

$H \leftarrow (b-a)/N$
$sum \leftarrow 0$
for $i \leftarrow 0$ to $N-1$ **do**
    $sum \leftarrow sum + f\Big(a + \Big(i + \frac{1}{2}\Big) \cdot H\Big)$
end for
$I \leftarrow H \cdot sum$
for $iteration \leftarrow 1$ to $maxiter$ **do**
    $H_{\text{new}} \leftarrow H/3$
    $N_{\text{new}} \leftarrow 3N$
    $sum_{\text{new}} \leftarrow 0$
    for $i \leftarrow 0$ to $N-1$ **do**
        $base \leftarrow a + i \cdot (3H_{\text{new}})$
        $x_{\text{left}} \leftarrow base + \frac{1}{2}H_{\text{new}}$
        $x_{\text{right}} \leftarrow base + \Big(\frac{1}{2} + 2\Big)H_{\text{new}}$
        $sum_{\text{new}} \leftarrow sum_{\text{new}} + f(x_{\text{left}}) + f(x_{\text{right}})$
    end for
    $sum \leftarrow sum + sum_{\text{new}}$
    $I_{\text{new}} \leftarrow H_{\text{new}} \cdot sum$
    if $|I_{\text{new}} - I| < tol$ **then**
        return I_{new}
    end if
    $I \leftarrow I_{\text{new}},\ N \leftarrow N_{\text{new}},\ H \leftarrow H_{\text{new}}$
end for
return I_{new}

The routine parameters accept the interval endpoints, a and b, the number of initial sub-intervals, N, and the pre-defined, chosen function $f(x)$. We fix the tolerance to 10^{-6} and the maximum number of iterations to 20. The routine refines the approximation until the difference between successive iterations is below the tolerance threshold, or the maximum number of iterations is reached. The routine returns the final approximation.

We also construct an adaptive refinement routine for the Composite Trapezoidal rule, setting the refinement factor to $\alpha = \frac{1}{2}$. We again provide an example to better explain the refinement steps: suppose we have an interval $[1, 3]$, which we wish to divide into $N = 2$ equal sub-intervals, where $H = \frac{3-1}{2} = 1$. The sub-intervals are $[1, 2]$ and $[2, 3]$. We evaluate the function at both endpoints of each sub-interval, such that

$$\begin{aligned} I(f) &= \left(\frac{1}{2}\right)[f(1) + f(2) + f(2) + f(3)] \\ &= \left(\frac{1}{2}\right)[f(1) + 2f(2) + f(3)]. \end{aligned}$$

Now consider the refinement factor, $\alpha = \frac{1}{2}$. We partition each sub-interval into two *new* sub-intervals: $[1, 1+\frac{1}{2}]$, $[1+\frac{1}{2}, 2]$, $[2, 2+\frac{1}{2}]$, $[2+\frac{1}{2}, 3]$. We now have $N = 2*2 = 4$ sub-intervals with $H = \frac{1}{2}*\frac{1}{2} = \frac{1}{4}$. The trapezoidal approximation becomes

$$I(f) = \left(\frac{1}{4}\right)\left[f(1) + 2f\left(1+\frac{1}{2}\right) + 2f(2) + 2f\left(2+\frac{1}{2}\right) + f(3)\right]$$

Notice that the evaluations from the initial approximation, $f(1)$, $2f(2)$, and $f(3)$, are completely reused in the refinement step. We construct an efficient routine that iteratively refines each sub-interval and reevaluates the approximation, while also reusing points that have already been computed in the previous iteration.

Algorithm 3 Adaptive_midpoint(a, b, N, f, $tol = 10^{-6}$, $maxiter = 20$)

```
H ← (b − a)/N
sum ← (f(a) + f(b))/2
for i ← 1 to N − 1 do
    sum ← sum + f(a + i · H)
end for
I ← H · sum
for iteration ← 1 to maxiter do
    H_new ← H/2
    N_new ← 2N
    for i ← 1 to N_new − 1 step 2 do
        sum ← sum + f(a + i · H_new)
    end for
    I_new ← H_new · sum
    if |I_new − I| < tol then
        return I_new
    end if
    I ← I_new, N ← N_new, H ← H_new
end for
return I
```

Just as in the adaptive midpoint algorithm, the routine parameters accept the interval endpoints, a and b, the number of initial sub-intervals, N, and the pre-defined, chosen function $f(x)$. We again fix the tolerance to 10^{-6} and the maximum number of iterations to 20. The routine refines the approximation until the difference between successive iterations is below the tolerance threshold, or the maximum number of iterations is reached. The final approximation is returned.

Lastly, we construct a routine for the Composite Gauss-Legendre method:

Algorithm 4 Composite_Gauss_Legendre(a, b, N, f)

$H \leftarrow (b-a)/N$
$x_1 \leftarrow 1/\sqrt{3}$
$sum \leftarrow 0$
for $i \leftarrow 0$ **to** $N-1$ **do**
    $a_i \leftarrow a + i \cdot H$
    $b_i \leftarrow a + (i+1) \cdot H$
    $term1 \leftarrow (b_i - a_2)/2$
    $term2 \leftarrow (b_i + a_2)/2$
    $sum \leftarrow sum + f((x_1 \cdot term1) + term2) + f((-x_1 \cdot term1) + term2)$
end for
return $(H/2) \cdot sum$

Similar to the previous routines, the algorithm parameters accept the interval endpoints, a and b, the number of sub-intervals, N, and the pre-defined functions $f(x)$. The algorithm applies the composite two-point Gauss-Legendre quadrature method and returns the resulting approximation.
Prior to testing the performance and behavior of these methods on various function types, we first demonstrate the correctness of these codes. Suppose we want to integrate the linear function $f(x) = 2x$ on the interval $[-1, 1]$. We know that

$$\int_{-1}^{1} 2x dx = \frac{1}{2}x^2\Big|_{-1}^{1} = \frac{1}{2}(1-1) = 0.$$

Since this is a linear function, only one point, n, is required to achieve an exact value. Therefore each method should approximate the integral exactly, using the global interval (i.e. only one sub-interval N, such that $H = \frac{b-a}{1} = b-a$). We demonstrate this with the following output for the composite open, composite closed, and adaptive quadrature methods:

Table 1: Approximations using Open Interval Methods

N	Midpoint ($n=1$)	$E(f)$	2-Point ($n=2$)	$E(f)$	Gauss-Legendre ($n=2$)	$E(f)$
1	0.00000000	0.00000000	-0.00000000	0.00000000	0.00000000	0.00000000

Table 2: Approximations using Closed Interval Methods

N	Left Rectangle ($n=1$)	$E(f)$	Trapezoidal ($n=2$)	$E(f)$	Simpson's ($n=3$)	$E(f)$
1	-4.00000000	4.00000000	0.00000000	0.00000000	0.00000000	0.00000000

Table 3: Approximations using Adaptive Midpoint Method

N	I_n	$I - I_n$	Error Estimate
1	0.00000000	-0.00000000	0.00000000

Table 4: Approximations using Adaptive Trapezoidal Method

N	I_n	$I - I_n$	Error Estimate
1	0.00000000	-0.00000000	0.00000000

The above outputs confirm the reliability and functionality of the constructed programs, since they determine the exact value of the linear function, using at least $n = 1$ points. This output serves as a basic outline for the formal function testing performed in the following section.

5 Description of the Experimental Design and Results

This section focuses on testing the performance of the composite quadrature (and adaptive-step composite quadrature) algorithms. We choose functions for which the value of definite integral on the given interval is known, and test the accuracy in the approximation of each of the routines against the true value of the integration. We use the composite error bounds for each method to determine the number of intervals needed so that the desired tolerance will be 10^{-6} when approximating the true value of $\mathcal{I}(f) = \int_a^b f(x)dx$. The chosen functions, integral values, and the necessary approximation results are shown below.

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

For the composite closed interval methods, the desired number of sub-intervals (derived from the composite error expressions) to achieve an error $\leq 10^{-6}$ are:

$$\begin{aligned} |E_{clr}(f)| = \frac{1}{2}(b-a)H|f'(\xi)| &\leq 10^{-6} \\ \frac{1}{2}(3)\frac{3}{N}e^3 &\leq 10^{-6} \qquad (|f'(x)| \leq e^3) \\ \frac{1}{2}3^2e^3 10^6 &\leq N \\ \implies N_{clr} &\gtrsim 90384916 \end{aligned}$$

Testing the Composite Left Rectangle method for $N = 90384916$ sub-intervals, the routine outputs the approximation $I(f) = 19.085536606455577$, with error

estimate $E(f) = 3.1673208766846983e-07$. This value is within our estimated error bounds.

$$
\begin{aligned}
|E_{ctr}(f)| = \frac{1}{12}(b-a)H^2|f''(\xi)| &\leq 10^{-6} \\
\frac{1}{12}(3)\frac{3^2}{N^2}e^3 &\leq 10^{-6} \qquad (|f''(x)| \leq e^3) \\
\frac{1}{12}3^3e^3 10^6 &\leq N^2 \\
\implies N_{ctr} &\gtrsim 6723
\end{aligned}
$$

Testing the Composite Trapezoidal method for $N = 6723$ sub-intervals, the routine outputs the approximation $I(f) = 19.085537239881344$, with error estimate $E(f) = -3.166936792808883e-07$. This value is within our estimated error bounds.

$$
\begin{aligned}
|E_{csfr}(f)| = \frac{1}{90}(b-a)H^4|f^{(4)}(\xi)| &\leq 10^{-6} \\
\frac{1}{90}(3)\frac{3^4}{N^4}e^3 &\leq 10^{-6} \qquad (|f^{(4)}(x)| \leq e^3) \\
\frac{1}{90}3^5e^3 10^6 &\leq N^4 \\
\implies N_{csfr} &\gtrsim 86
\end{aligned}
$$

Testing the Composite Simpson's method for $N = 86$ sub-intervals, the routine outputs the approximation $I(f) = 19.085537080173395$, with error estimate $E(f) = -1.5698573108124947e-07$. This value is within our estimated error bounds.

For the composite open interval quadrature methods, the desired number of sub-intervals (derived from the composite error expressions) to achieve an error of $\leq 10^{-6}$ are:

$$
\begin{aligned}
|E_{cmpr}(f)| = \frac{1}{24}(b-a)H^2|f''(\xi)| &\leq 10^{-6} \\
\frac{1}{24}(3)\frac{3^2}{N^2}e^3 &\leq 10^{-6} \qquad (|f''(x)| \leq e^3) \\
\frac{1}{24}3^3e^3 10^6 &\leq N^2 \\
\implies N_{cmpr} &\gtrsim 4754
\end{aligned}
$$

Testing the Composite Midpoint Rule for $N = 4754$ sub-intervals, the routine outputs the approximation $I(f) = 19.085536606510136$, with error estimate $E(f) = 3.166775286445045e-07$. As observed with the previous quadrature

methods, this value is within our estimated error bounds.

$$
\begin{aligned}
|E_{c2pr}(f)| = \frac{1}{36}(b-a)H^2|f''(\xi)| &\leq 10^{-6} \\
\frac{1}{36}(3)\frac{3^2}{N^2}e^3 &\leq 10^{-6} \qquad (|f''(x)| \leq e^3) \\
\frac{1}{36}3^3e^3 10^6 &\leq N^2 \\
\implies N_{ctr} &\gtrsim 915
\end{aligned}
$$

Testing the Composite Two-Point Rule for $N = 915$ sub-intervals, the routine outputs the approximation $I(f) = 19.085531224142635$, with error estimate $E(f) = 5.699045029672334e-06$. This value is within our estimated error bounds, meaning we have estimated a valid bound for our error terms.

$$|E_{cgl}| = \frac{1}{270}(b-a)|f^{(4)}(\xi)| \leq \frac{1}{270}(3)e^3 \approx 0.22317263$$

Notice that the composite Gauss-Legendre error term gives us an exact estimate for the error, regardless of the H or N. We test the Gauss-Legendre two-point method for $N = 1$ (i.e. on the global interval $[a, b]$, where $a = 0$, $b = 3$ for this problem). The approximation for the Composite Gauss-Legendre method is $I(f) = 18.810070539739872$, with exact error estimate given to be $E(f) = 0.2754663834477924$. This error term falls outside of our theoretical estimate for the error bound, though it is close in value to our approximated error bound. We test the composite (and adaptive composite) methods, including the desired number of sub-intervals (if possible):

Table 5: Approximations using Open Interval Methods

N	Midpoint ($n = 1$)	$E(f)$	2-Point ($n = 2$)	$E(f)$	Gauss-Legendre ($n = 2$)	$E(f)$
5	18.80223158	0.28330535	18.89632110	0.18921582	19.08497084	0.00056608
10	19.01415359	0.07138333	19.03792623	0.04761069	19.08550124	0.00003568
20	19.06765597	0.01788096	19.07361492	0.01192200	19.08553469	0.00000223
40	19.08106448	0.00447244	19.08255521	0.00298171	19.08553678	0.00000014
80	19.08441868	0.00111825	19.08479142	0.00074550	19.08553691	0.00000001

Table 6: Approximations using Closed Interval Methods

N	Left Rectangle ($n = 1$)	$E(f)$	Trapezoidal ($n = 2$)	$E(f)$	Simpson's ($n = 3$)	$E(f)$
5	13.92903574	5.15650118	19.65469682	-0.56915989	16.25183772	2.83369921
10	16.36563366	2.71990326	19.22846420	-0.14292727	19.08638666	-0.00084973
20	17.68989362	1.39564330	19.12130889	-0.03577197	19.08559046	-0.00005353
40	18.37877480	0.70676213	19.09448243	-0.00894551	19.08554028	-0.00000335
80	18.72991964	0.35561728	19.08777346	-0.00223653	19.08553713	-0.00000021

From the above tables, we observe the error terms decrease exponentially as the number of sub-intervals, N, increase from 5 to 80. For each method, the approximation for the definite integral improves when a larger number of sub-intervals are chosen. Comparing the composite open interval methods, the Composite Gauss-Legendre method provides the best approximation for the integral, for each chosen value of N. The Composite Midpoint Rule provides the least accurate approximation of the examined open interval methods, as the error between the true and estimated value is highest for each chosen value of N. Observing the composite closed interval methods, the Trapezoidal Rule provides the best approximation when $N = 5$, but Simpson's First Rule provides the best approximation for the larger sub-interval sizes ($N = 10, 20, 40, 80$). For each chosen value of N, the Left Rectangle Rule results in the highest error in the approximation, for both the open and closed interval methods. This corresponds to our calculations of N, where the required number of sub-intervals to achieve an error approximation within the 10^{-6} tolerance value was highest among all the observed methods.
We now examine the performance of the adaptive Midpoint and Trapezoidal methods:

Table 7: Approximations using Adaptive Midpoint Method

N	I_n	$I_f - I_c$	E(f)
1	13.44506721	5.47631673	5.64046971
3	18.31290430	0.77012906	0.77263262
9	18.99746347	0.08804174	0.08807346
27	19.07572279	0.00981374	0.00981413
81	19.08444612	0.00109080	0.00109081
243	19.08541572	0.00012121	0.00012121
729	19.08552346	0.00001347	0.00001347
2187	19.08553543	0.00000150	0.00000150
6561	19.08553676	0.00000017	0.00000017

Table 8: Approximations using Adaptive Trapezoidal Method

N	I_n	$I_f - I_c$	E(f)
1	31.62830538	-12.12215878	-12.54276846
2	22.53668630	-3.41972168	-3.45114937
4	19.97189504	-0.88429591	-0.88635812
8	19.30867311	-0.22300568	-0.22313618
16	19.14141885	-0.05587374	-0.05588192
32	19.09951354	-0.01397611	-0.01397662
64	19.08903146	-0.00349451	-0.00349454
128	19.08641058	-0.00087366	-0.00087366
256	19.08575534	-0.00021842	-0.00021842
512	19.08559153	-0.00005460	-0.00005460
1024	19.08555057	-0.00001365	-0.00001365
2048	19.08554034	-0.00000341	-0.00000341
4096	19.08553778	-0.00000085	-0.00000085

In the above tables, the first column indicates the total number of subintervals, N, used at each step in the refinement process. For each method, we begin with the global interval (setting $N = 1$), and iterate until the appropriate tolerance is reach. The value in the second column indicates the integral approximation, I_n. In the third column, we compare the difference between the estimate for in the previous refinement step (I_f) to the estimate in the current refinement step (I_c). This is the value that is compared to our set tolerance (10^{-6}). To measure these differences accurately, we must apply an appropriate scalar to the difference. For the Adaptive Midpoint Method, we find

$$\begin{aligned}\hat{E}_f &= \frac{1}{3^r}\hat{E}_c \\ &\implies \frac{3^r}{3^r-1}(I_f - I_c) = \frac{3^2}{3^2-1}(I_f - I_c) = \frac{9}{8}(I_f - I_c) = \hat{E}_c + O(H^3),\end{aligned}$$

where r is the order of infinitesimal (the order of H in the composite error term - for the Composite Midpoint method, $r = 2$). Similarly, for the Adaptive Trapezoidal Method, we find

$$\begin{aligned}\hat{E}_f &= \frac{1}{2^r}\hat{E}_c \\ &\implies \frac{2^r}{2^r-1}(I_f - I_c) = \frac{2^2}{2^2-1}(I_f - I_c) = \frac{4}{3}(I_f - I_c) = \hat{E}_c + O(H^3),\end{aligned}$$

where r is the order of infinitesimal (the order of H in the composite error term - for the Composite Trapezoidal method, $r = 2$). We the appropriate scalars are applied to compute the difference in the refinement at each step. The final column observes the difference between the true value of the definite integral and the value of the approximation in the current refinement step. We observe similar results in the following functions tested.

5.2 Function 2: $\int_0^{\frac{\pi}{3}} e^{sin(2x)}\cos(2x)dx = \frac{1}{2}\left(-1+e^{\frac{\sqrt{3}}{2}}\right) \approx 0.68872134$

$$
\begin{aligned}
|E_{clr}(f)| = \frac{1}{2}(b-a)H|f'(\xi)| &\leq 10^{-6} \\
\frac{1}{2}\left(\frac{\pi}{3}\right)\frac{\left(\frac{\pi}{3}\right)}{N}(2) &\leq 10^{-6} \qquad (|f'(x)| \leq 2) \\
\frac{1}{2}\left(\frac{\pi}{3}\right)^2(2)10^6 &\leq N \\
\implies N_{clr} &\gtrsim 1096623
\end{aligned}
$$

Testing the Composite Left Rectangle Rule for $N = 1096623$ sub-intervals, the routine outputs the approximation $I(f) = 0.6887223826548905$, with error estimate $E(f) = -1.0450368082004502e-06$.

$$
\begin{aligned}
|E_{ctr}(f)| = \frac{1}{12}(b-a)H^2|f''(\xi)| &\leq 10^{-6} \\
\frac{1}{12}\left(\frac{\pi}{3}\right)\frac{\left(\frac{\pi}{3}\right)^2}{N^2}(6-\sqrt{2})e^{-\sqrt{2}/2} &\leq 10^{-6} \qquad (|f''(x)| \leq (6-\sqrt{2})e^{-\sqrt{2}/2}) \\
\frac{1}{12}\left(\frac{\pi}{3}\right)^3(6-\sqrt{2})e^{-\sqrt{2}/2}10^6 &\leq N^2 \\
\implies N_{ctr} &\gtrsim 466
\end{aligned}
$$

Testing the Composite Trapezoidal Rule for $N = 466$ sub-intervals, the routine outputs the approximation $I(f) = 0.6887192633029027$, with error estimate $E(f) = 2.0743151796231984e-06$.

$$
\begin{aligned}
|E_{csfr}(f)| = \frac{1}{90}(b-a)H^4|f^{(4)}(\xi)| &\leq 10^{-6} \\
\frac{1}{90}\left(\frac{\pi}{3}\right)\frac{\left(\frac{\pi}{3}\right)^4}{N^4}(80+30\sqrt{2})e^{-\sqrt{2}/2} &\leq 10^{-6} \qquad (|f^{(4)}(x)| \leq (80+30\sqrt{2})e^{-\sqrt{2}/2}) \\
\frac{1}{90}\left(\frac{\pi}{3}\right)^5(80+30\sqrt{2})e^{-\sqrt{2}/2}10^6 &\leq N^4 \\
\implies N_{csfr} &\gtrsim 31
\end{aligned}
$$

Testing the Composite Simpson's Rule for $N = 31$ sub-intervals, the routine outputs the approximation $I(f) = 0.7015494718760034$, with error estimate $E(f) = -0.012828134257921064$.
For the composite open interval quadrature methods, the desired number of sub-intervals (derived from the composite error expressions) to achieve an error

of $\leq 10^{-6}$ are:

$$|E_{cmpr}(f)| = \frac{1}{24}(b-a)H^2|f''(\xi)| \leq 10^{-6}$$
$$\frac{1}{24}\left(\frac{\pi}{3}\right)\frac{\left(\frac{\pi}{3}\right)^2}{N^2}(6-\sqrt{2})e^{-\sqrt{2}/2} \leq 10^{-6} \qquad (|f''(x)| \leq (6-\sqrt{2})e^{-\sqrt{2}/2})$$
$$\frac{1}{24}\left(\frac{\pi}{3}\right)^3(6-\sqrt{2})e^{-\sqrt{2}/2}10^6 \leq N^2$$
$$\implies N_{cmpr} \gtrsim 329$$

Testing the Composite Midpoint Rule for $N = 329$ sub-intervals, the routine outputs the approximation $I(f) = 0.6887234183951166$, with error estimate $E(f) = -2.0807770343411747e-06$.

$$|E_{c2pr}(f)| = \frac{1}{36}(b-a)H^2|f''(\xi)| \leq 10^{-6}$$
$$\frac{1}{36}\left(\frac{\pi}{3}\right)\frac{\left(\frac{\pi}{3}\right)^2}{N^2}(6-\sqrt{2})e^{-\sqrt{2}/2} \leq 10^{-6} \qquad (|f''(x)| \leq (6-\sqrt{2})e^{-\sqrt{2}/2})$$
$$\frac{1}{36}\left(\frac{\pi}{3}\right)^3(6-\sqrt{2})e^{-\sqrt{2}/2}10^6 \leq N^2$$
$$\implies N_{c2pr} \gtrsim 64$$

Testing the Composite Two-Point Rule for $N = 64$ sub-intervals, the routine outputs the approximation $I(f) = 0.6887579972584774$, with error estimate $E(f) = -3.665964039512026e-05$.

$$|E_{cgl}| = \frac{1}{270}(b-a)|f^{(4)}(\xi)| \leq \frac{1}{270}\left(\frac{\pi}{3}\right)(80+30\sqrt{2})e^{-\sqrt{2}/2} \approx 0.23412478$$

We test the Gauss-Legendre two-point method for $N = 1$ sub-interval. The approximation for the Composite Gauss-Legendre method is $I(f) = 0.6113183619819696$, with exact error estimate $E(f) = 0.07740297563611265$.
We test the composite (and adaptive composite) methods, including the desired number of sub-intervals (if possible):

Table 9: Approximations using Open Interval Methods

N	Midpoint ($n=1$)	$E(f)$	2-Point ($n=2$)	$E(f)$	Gauss-Legendre ($n=2$)	$E(f)$
5	0.69782405	-0.00910271	0.69477896	-0.00605762	0.68870355	0.00001779
10	0.69097952	-0.00225818	0.69022610	-0.00150476	0.68872021	0.00000113
20	0.68928477	-0.00056343	0.68909692	-0.00037558	0.68872127	0.00000007
40	0.68886213	-0.00014079	0.68881519	-0.00009386	0.68872133	0.00000000
80	0.68875653	-0.00003519	0.68874480	-0.00002346	0.68872134	0.00000000

Table 10: Approximations using Closed Interval Methods

N	Left Rectangle ($n = 1$)	$E(f)$	Trapezoidal ($n = 2$)	$E(f)$	Simpson's ($n = 3$)	$E(f)$
5	0.89979843	-0.21107709	0.67059607	0.01812527	0.75032332	-0.06160198
10	0.79881124	-0.11008990	0.68421006	0.00451128	0.68874806	-0.00002672
20	0.74489538	-0.05617404	0.68759479	0.00112655	0.68872303	-0.00000169
40	0.71709008	-0.02836874	0.68843978	0.00028156	0.68872144	-0.00000011
80	0.70297610	-0.01425476	0.68865095	0.00007038	0.68872134	-0.00000001

Table 11: Approximations using Adaptive Midpoint Method

N	I_n	$I - I_n$	Error Estimate
1	1.24482607	-0.59669127	-0.55610474
3	0.71443384	-0.02578828	-0.02571250
9	0.69151092	-0.00279059	-0.00278959
27	0.68903040	-0.00030907	-0.00030906
81	0.68875567	-0.00003433	-0.00003433
243	0.68872515	-0.00000381	-0.00000381
729	0.68872176	-0.00000042	-0.00000042

Table 12: Approximations using Adaptive Trapezoidal Method

N	I_n	$I - I_n$	Error Estimate
1	-0.09881426	0.89576022	0.78753560
2	0.57300591	0.11640433	0.11571543
4	0.66030915	0.02847651	0.02841219
8	0.68166654	0.00705893	0.00705480
16	0.68696073	0.00176086	0.00176061
32	0.68828138	0.00043997	0.00043996
64	0.68861136	0.00010998	0.00010998
128	0.68869384	0.00002749	0.00002749
256	0.68871446	0.00000687	0.00000687
512	0.68871962	0.00000172	0.00000172
1024	0.68872091	0.00000043	0.00000043

5.3 Function 3: $\int_{-2}^{1} \tanh(x)dx = \ln\left(\frac{\cosh(1)}{\cosh(2)}\right) \approx -0.89122192$

$$
\begin{aligned}
|E_{clr}(f)| = \frac{1}{2}(b-a)H|f'(\xi)| &\leq 10^{-6} \\
\frac{1}{2}(3)\frac{3}{N} &\leq 10^{-6} \qquad (|f'(x)| \leq 1) \\
\frac{1}{2}3^2 10^6 &\leq N \\
\implies N_{clr} &\gtrsim 4500000
\end{aligned}
$$

Testing the Composite Left Rectangle Rule for $N = 4500000$ sub-intervals, the routine outputs the approximation $I(f) = -0.8912224920820421$, with error estimate $E(f) = 5.75207204756012e-07$.

$$
\begin{aligned}
|E_{ctr}(f)| = \frac{1}{12}(b-a)H^2|f''(\xi)| &\leq 10^{-6} \\
\frac{1}{12}(3)\frac{3^2}{N^2}\frac{4\sqrt{3}}{9} &\leq 10^{-6} \qquad (|f''(x)| \leq \frac{4\sqrt{3}}{9}) \\
\frac{1}{12}3^3\frac{4\sqrt{3}}{9}10^6 &\leq N^2 \\
\implies N_{ctr} &\gtrsim 1317
\end{aligned}
$$

Testing the Composite Trapezoidal Rule for $N = 1317$ sub-intervals, the routine outputs the approximation $I(f) = -0.8912217658259614$, with error estimate $E(f) = -1.5104887596262273e-07$.

$$
\begin{aligned}
|E_{csfr}(f)| = \frac{1}{90}(b-a)H^4|f^{(4)}(\xi)| &\leq 10^{-6} \\
\frac{1}{90}(3)\frac{3^4}{N^4}(4.084) &\leq 10^{-6} \qquad (|f^{(4)}(x)| \leq\approx 4.084) \\
\frac{1}{90}3^5(4.084)10^6 &\leq N^4 \\
\implies N_{csfr} &\gtrsim 58
\end{aligned}
$$

Testing the Composite Simpson's Rule for $N = 58$ sub-intervals, the routine outputs the approximation $I(f) = -0.8912219021249569$, with error estimate $E(f) = -1.4749880428155393e-08$.
For the composite open interval quadrature methods, the desired number of sub-intervals (derived from the composite error expressions) to achieve an error

of $\leq 10^{-6}$ are:

$$|E_{cmpr}(f)| = \frac{1}{24}(b-a)H^2|f''(\xi)| \leq 10^{-6}$$
$$\frac{1}{24}(3)\frac{3^2}{N^2}\frac{4\sqrt{3}}{9} \leq 10^{-6} \qquad (|f''(x)| \leq \frac{4\sqrt{3}}{9})$$
$$\frac{1}{24}3^3\frac{4\sqrt{3}}{9}10^6 \leq N^2$$
$$\implies N_{cmpr} \gtrsim 931$$

Testing the Composite Midpoint Rule for $N = 931$ sub-intervals, the routine outputs the approximation $I(f) = -0.8912220680079368$, with error estimate $E(f) = 1.5113309947967224e-07$.

$$|E_{c2pr}(f)| = \frac{1}{36}(b-a)H^2|f''(\xi)| \leq 10^{-6}$$
$$\frac{1}{36}(3)\frac{3^2}{N^2}\frac{4\sqrt{3}}{9} \leq 10^{-6} \qquad (|f''(x)| \leq \frac{4\sqrt{3}}{9})$$
$$\frac{1}{36}3^3\frac{4\sqrt{3}}{9}10^6 \leq N^2$$
$$\implies N_{c2pr} \gtrsim 180$$

Testing the Composite Two-Point Rule for $N = 180$ sub-intervals, the routine outputs the approximation $I(f) = -0.8912246122533023$, with error estimate $E(f) = 2.6953784649785817e-06$.

$$|E_{cgl}| = \frac{1}{270}(b-a)|f^{(4)}(\xi)| \leq \frac{1}{270}(3)(4.084) \approx 0.04537778$$

We test the Gauss-Legendre two-point method for $N = 1$ sub-interval. The approximation for the Composite Gauss-Legendre method is $I(f) = -0.7909093792346076$, with exact error estimate $E(f) = -0.10031253764022974$.
We test the composite (and adaptive composite) methods, including the desired number of sub-intervals (if possible):

Table 13: Approximations using Open Interval Methods

N	Midpoint ($n=1$)	$E(f)$	2-Point ($n=2$)	$E(f)$	Gauss-Legendre ($n=2$)	$E(f)$
5	-0.89639369	0.00517177	-0.89467806	0.00345614	-0.89123528	0.00001336
10	-0.89252810	0.00130619	-0.89209315	0.00087124	-0.89122264	0.00000072
20	-0.89154918	0.00032726	-0.89144012	0.00021820	-0.89122196	0.00000004
40	-0.89130378	0.00008186	-0.89127649	0.00005457	-0.89122192	0.00000000
80	-0.89124238	0.00002047	-0.89123556	0.00001364	-0.89122192	0.00000000

Table 14: Approximations using Closed Interval Methods

N	Left Rectangle ($n=1$)	$E(f)$	Trapezoidal ($n=2$)	$E(f)$	Simpson's ($n=3$)	$E(f)$
5	-1.39850541	0.50728350	-0.88081889	-0.01040302	-1.02030338	0.12908147
10	-1.14744955	0.25622763	-0.88860629	-0.00261563	-0.89120209	-0.00001983
20	-1.01998883	0.12876691	-0.89056720	-0.00065472	-0.89122083	-0.00000108
40	-0.95576900	0.06454709	-0.89105819	-0.00016373	-0.89122185	-0.00000007
80	-0.92353639	0.03231447	-0.89118098	-0.00004094	-0.89122191	-0.00000000

Table 15: Approximations using Adaptive Midpoint Method

N	I_n	$I - I_n$	Error Estimate
1	-1.38635147	0.54135362	0.49512955
3	-0.90514825	0.01385427	0.01392634
9	-0.89283335	0.00161078	0.00161143
27	-0.89140154	0.00017962	0.00017962
81	-0.89124188	0.00001996	0.00001997
243	-0.89122414	0.00000222	0.00000222
729	-0.89122216	0.00000025	0.00000025

Table 16: Approximations using Adaptive Trapezoidal Method

N	I_n	$I - I_n$	Error Estimate
1	-0.30365014	-0.72180089	-0.58757178
2	-0.84500080	-0.04003111	-0.04622111
4	-0.87502414	-0.01615313	-0.01619778
8	-0.88713898	-0.00408022	-0.00408293
16	-0.89019915	-0.00102261	-0.00102277
32	-0.89096610	-0.00025580	-0.00025581
64	-0.89115796	-0.00006396	-0.00006396
128	-0.89120593	-0.00001599	-0.00001599
256	-0.89121792	-0.00000400	-0.00000400
512	-0.89122092	-0.00000100	-0.00000100

5.4 Function 4: $\int_0^{3.5} x\cos(2\pi x)dx = -\frac{1}{2\pi^2} \approx -0.05066059$

$$
\begin{aligned}
|E_{clr}(f)| = \frac{1}{2}(b-a)H|f'(\xi)| &\leq 10^{-6} \\
\frac{1}{2}(3.5)\frac{3.5}{N}\frac{11\pi}{2} &\leq 10^{-6} \qquad (|f'(x)| \leq \frac{11\pi}{2}) \\
\frac{1}{2}3.5^2\left(\frac{11\pi}{2}\right)10^6 &\leq N \\
\implies N_{clr} &\gtrsim 105845000
\end{aligned}
$$

Testing the Composite Left Rectangle Rule for $N = 105845000$ sub-intervals, the routine outputs the approximation $I(f) = -0.05066053395366144$, with error estimate $E(f) = -5.7867507445574784e-08$.

$$
\begin{aligned}
|E_{ctr}(f)| = \frac{1}{12}(b-a)H^2|f''(\xi)| &\leq 10^{-6} \\
\frac{1}{12}(3.5)\frac{3.5^2}{N^2}14\pi^2 &\leq 10^{-6} \qquad (|f''(x)| \leq 14\pi^2) \\
\frac{1}{12}3.5^3 14\pi^2 10^6 &\leq N^2 \\
\implies N_{ctr} &\gtrsim 22181
\end{aligned}
$$

Testing the Composite Trapezoidal Rule for $N = 22181$ sub-intervals, the routine outputs the approximation $I(f) = -0.05066059597092315$, with error estimate $E(f) = 4.1497542624391315e-09$.

$$
\begin{aligned}
|E_{csfr}(f)| = \frac{1}{90}(b-a)H^4|f^{(4)}(\xi)| &\leq 10^{-6} \\
\frac{1}{90}(3.5)\frac{3.5^4}{N^4}48\pi^4 &\leq 10^{-6} \qquad (|f^{(4)}(x)| \leq 48\pi^4) \\
\frac{1}{90}3.5^5 48\pi^4 10^6 &\leq N^4 \\
\implies N_{csfr} &\gtrsim 405
\end{aligned}
$$

Testing the Composite Simpson's Rule for $N = 405$ sub-intervals, the routine outputs the approximation $I(f) = -0.04059073641914929$, with error estimate $E(f) = -0.0100698554020196$.
For the composite open interval quadrature methods, the desired number of sub-intervals (derived from the composite error expressions) to achieve an error of $\leq 10^{-6}$ are:

$$
\begin{aligned}
|E_{cmpr}(f)| = \frac{1}{24}(b-a)H^2|f''(\xi)| &\leq 10^{-6} \\
\frac{1}{24}(3.5)\frac{3.5^2}{N^2}14\pi^2 &\leq 10^{-6} \qquad (|f''(x)| \leq 14\pi^2) \\
\frac{1}{24}3.5^3 14\pi^2 10^6 &\leq N^2 \\
\implies N_{cmpr} &\gtrsim 15698
\end{aligned}
$$

Testing the Composite Midpoint Rule for $N = 15698$ sub-intervals, the routine outputs the approximation $I(f) = -0.050660587678633476$, with error estimate

$E(f) = -4.142535411921777e-09.$

$$|E_{c2pr}(f)| = \frac{1}{36}(b-a)H^2|f''(\xi)| \leq 10^{-6}$$
$$\frac{1}{36}(3.5)\frac{3.5^2}{N^2}14\pi^2 \leq 10^{-6} \qquad (|f''(x)| \leq 14\pi^2)$$
$$\frac{1}{36}3.5^3 14\pi^2 10^6 \leq N^2$$
$$\implies N_{c2pr} \gtrsim 3019$$

Testing the Composite Two-Point Rule for $N = 3019$ sub-intervals, the routine outputs the approximation $I(f) = -0.050660517152394516$, with error estimate $E(f) = -7.466877437189989e-08.$

$$|E_{cgl}| = \frac{1}{270}(b-a)|f^{(4)}(\xi)| \leq \frac{1}{270}(3.5)(48\pi^4) \approx 6.14108718$$

We test the Gauss-Legendre two-point method for $N = 1$ sub-interval. The approximation for the Composite Gauss-Legendre method is $I(f) = 0.23009262672111655$, with exact error estimate $E(f) = -0.2807532185422854$.
We test the composite (and adaptive composite) methods, including the desired number of sub-intervals (if possible):

Table 17: Approximations using Open Interval Methods

N	Midpoint ($n = 1$)	$E(f)$	2-Point ($n = 2$)	$E(f)$	Gauss-Legendre ($n = 2$)	$E(f)$
5	0.22002371	-0.27068430	0.09596372	-0.14662431	-0.10171453	0.05105394
10	-0.03502604	-0.01563455	-0.04091133	-0.00974926	-0.05173889	0.00107830
20	-0.04782354	-0.00283706	-0.04880294	-0.00185765	-0.05071546	0.00005487
40	-0.05000547	-0.00065512	-0.05022584	-0.00043475	-0.05066386	0.00000327
80	-0.05050003	-0.00016056	-0.05055367	-0.00010692	-0.05066079	0.00000020

Table 18: Approximations using Closed Interval Methods

N	Left Rectangle ($n = 1$)	$E(f)$	Trapezoidal ($n = 2$)	$E(f)$	Simpson's ($n = 3$)	$E(f)$
5	0.85067331	-0.90133390	-0.37432669	0.32366610	0.44233998	-0.49300057
10	0.53534851	-0.58600910	-0.07715149	0.02649090	0.02190691	-0.07256750
20	0.25016123	-0.30082182	-0.05608877	0.00542818	-0.04906786	-0.00159273
40	0.10116885	-0.15182944	-0.05195615	0.00129556	-0.05057861	-0.00008198
80	0.02558169	-0.07624228	-0.05098081	0.00032022	-0.05065570	-0.00000489

Table 19: Approximations using Adaptive Midpoint Method

N	I_n	$I - I_n$	Error Estimate
1	-0.00000000	2.65220280	-0.05066059
3	2.35751360	-2.68515266	-2.40817419
9	-0.02928877	-0.02237377	-0.02137182
27	-0.04917656	-0.00149336	-0.00148403
81	-0.05050399	-0.00015671	-0.00015660
243	-0.05064329	-0.00001730	-0.00001730
729	-0.05065867	-0.00000192	-0.00000192
2187	-0.05066038	-0.00000021	-0.00000021

Table 20: Approximations using Adaptive Trapezoidal Method

N	I_n	$I - I_n$	Error Estimate
1	-6.12500000	4.08333333	6.07433941
2	-3.06250000	0.59799032	3.01183941
4	-2.61400726	3.35269004	2.56334667
8	-0.09948973	0.05338683	0.04882914
16	-0.05944961	0.00899629	0.00878901
32	-0.05270239	0.00205384	0.00204179
64	-0.05116200	0.00050215	0.00050141
128	-0.05078539	0.00012484	0.00012480
256	-0.05069176	0.00003117	0.00003116
512	-0.05066838	0.00000779	0.00000779
1024	-0.05066254	0.00000195	0.00000195
2048	-0.05066108	0.00000049	0.00000049

5.5 Function 5: $\int_{0.1}^{2.5} \left(x + \frac{1}{x}\right) dx = \frac{2.5^2 - 0.1^2}{2} + \ln\left(\frac{2.5}{0.1}\right) \approx 6.33887582$

$$
\begin{aligned}
|E_{clr}(f)| = \frac{1}{2}(b-a)H|f'(\xi)| &\leq 10^{-6} \\
\frac{1}{2}(2.4)\frac{2.4}{N}99 &\leq 10^{-6} \qquad (|f'(x)| \leq 99) \\
\frac{1}{2}2.4^2(99)10^6 &\leq N \\
\implies N_{clr} &\gtrsim 285120000
\end{aligned}
$$

Testing the Composite Left Rectangle Rule for $N = 285120000$ sub-intervals, the routine outputs the approximation $I(f) = 6.338875855167809$, with error

estimate $E(f) = -3.0299609043993314e-08$.

$$
\begin{aligned}
|E_{ctr}(f)| = \frac{1}{12}(b-a)H^2|f''(\xi)| &\leq 10^{-6} \\
\frac{1}{12}(2.4)\frac{2.4^2}{N^2}2000 &\leq 10^{-6} \qquad (|f''(x)| \leq 2000) \\
\frac{1}{12}2.4^3(2000)10^6 &\leq N^2 \\
\implies N_{ctr} &\gtrsim 48000
\end{aligned}
$$

Testing the Composite Trapezoidal Rule for $N = 48000$ sub-intervals, the routine outputs the approximation $I(f) = 6.338875845668192$, with error estimate $E(f) = -2.07999919510371e-08$.

$$
\begin{aligned}
|E_{csfr}(f)| = \frac{1}{90}(b-a)H^4|f^{(4)}(\xi)| &\leq 10^{-6} \\
\frac{1}{90}(2.4)\frac{2.4^4}{N^4}(2.4 \cdot 10^6) &\leq 10^{-6} \qquad (|f^{(4)}(x)| \leq 2.4 \cdot 10^6) \\
\frac{1}{90}2.4^5(2.4 \cdot 10^6)10^6 &\leq N^4 \\
\implies N_{csfr} &\gtrsim 971
\end{aligned}
$$

Testing the Composite Simpson's Rule for $N = 971$ sub-intervals, the routine outputs the approximation $I(f) = 6.336487403185955$, with error estimate $E(f) = 0.0023884216822454007$.

For the composite open interval quadrature methods, the desired number of sub-intervals (derived from the composite error expressions) to achieve an error of $\leq 10^{-6}$ are:

$$
\begin{aligned}
|E_{cmpr}(f)| = \frac{1}{24}(b-a)H^2|f''(\xi)| &\leq 10^{-6} \\
\frac{1}{24}(2.4)\frac{2.4^2}{N^2}2000 &\leq 10^{-6} \qquad (|f''(x)| \leq 2000) \\
\frac{1}{24}2.4^3(2000)10^6 &\leq N^2 \\
\implies N_{cmpr} &\gtrsim 33941
\end{aligned}
$$

Testing the Composite Midpoint Rule for $N = 33941$ sub-intervals, the routine outputs the approximation $I(f) = 6.338875804068028$, with error estimate $E(f) = 2.08001722512563e-08$.

$$
\begin{aligned}
|E_{c2pr}(f)| = \frac{1}{36}(b-a)H^2|f''(\xi)| &\leq 10^{-6} \\
\frac{1}{36}(2.4)\frac{2.4^2}{N^2}2000 &\leq 10^{-6} \qquad (|f''(x)| \leq 2000) \\
\frac{1}{36}2.4^3(2000)10^6 &\leq N^2 \\
\implies N_{c2pr} &\gtrsim 6534
\end{aligned}
$$

Testing the Composite Two-Point Rule for $N = 6534$ sub-intervals, the routine outputs the approximation $I(f) = 6.338875450701253$, with error estimate $E(f) = 3.741669472034914e - 07$.

$$|E_{cgl}| = \frac{1}{270}(b-a)|f^{(4)}(\xi)| \leq \frac{1}{270}(2.4)(2.4 \cdot 10^6) \approx 21333.33333$$

We test the Gauss-Legendre two-point method for $N = 1$ sub-interval. The approximation for the Composite Gauss-Legendre method is $I(f) = 5.698512396694215$, with exact error estimate $E(f) = 0.640363428173985$.
We test the composite (and adaptive composite) methods, including the desired number of sub-intervals (if possible):

Table 21: Approximations using Open Interval Methods

N	Midpoint ($n = 1$)	$E(f)$	2-Point ($n = 2$)	$E(f)$	Gauss-Legendre ($n = 2$)	$E(f)$
5	5.96841363	0.37046219	6.05899541	0.27988041	6.27348958	0.06538624
10	6.18711240	0.15176343	6.23054585	0.10832997	6.32583938	0.01303645
20	6.28868401	0.05019182	6.30443701	0.03443881	6.33719876	0.00167706
40	6.32471009	0.01416573	6.32934217	0.00953366	6.33872639	0.00014944
80	6.33518832	0.00368751	6.33641100	0.00246482	6.33886516	0.00001066

Table 22: Approximations using Closed Interval Methods

N	Left Rectangle ($n = 1$)	$E(f)$	Trapezoidal ($n = 2$)	$E(f)$	Simpson's ($n = 3$)	$E(f)$
5	9.17372847	-2.83485264	7.44572847	-1.10685264	6.40173533	-0.06285950
10	7.57107105	-1.23219523	6.70707105	-0.36819523	6.46085191	-0.12197609
20	6.87909172	-0.54021590	6.44709172	-0.10821590	6.36043195	-0.02155612
40	6.58388787	-0.24501204	6.36788787	-0.02901204	6.34148658	-0.00261076
80	6.45429898	-0.11542316	6.34629898	-0.00742316	6.33910269	-0.00022686

Table 23: Approximations using Adaptive Midpoint Method

N	I_n	$I - I_n$	Error Estimate
1	4.96615385	0.84395604	1.37272198
3	5.71633700	0.50197476	0.62253883
9	6.16253678	0.16524395	0.17633905
27	6.30942029	0.02908937	0.02945554
81	6.33527751	0.00359236	0.00359831
243	6.33847072	0.00040502	0.00040510
729	6.33883075	0.00004508	0.00004508
2187	6.33887082	0.00000501	0.00000501
6561	6.33887527	0.00000056	0.00000056

Table 24: Approximations using Adaptive Trapezoidal Method

N	I_n	$I - I_n$	Error Estimate
1	15.60000000	-7.08923077	-9.26112418
2	10.28307692	-3.21147484	-3.94420110
4	7.87447079	-1.33782901	-1.53559497
8	6.87109903	-0.49300506	-0.53222321
16	6.50134524	-0.15710474	-0.16246942
32	6.38351668	-0.04412792	-0.04464086
64	6.35042074	-0.01150674	-0.01154492
128	6.34179069	-0.00291234	-0.00291487
256	6.33960643	-0.00073045	-0.00073061
512	6.33905860	-0.00018276	-0.00018277
1024	6.33892153	-0.00004570	-0.00004570
2048	6.33888725	-0.00001143	-0.00001143
4096	6.33887868	-0.00000286	-0.00000286
8192	6.33887654	-0.00000071	-0.00000071

5.6 Function 6: $\int_0^4 2^x dx = \frac{15}{\ln(2)} \approx 21.64042561$

$$
\begin{aligned}
|E_{clr}(f)| = \frac{1}{2}(b-a)H|f'(\xi)| &\leq 10^{-6} \\
\frac{1}{2}(4)\frac{4}{N}2^4\ln(2) &\leq 10^{-6} \qquad (|f'(x)| \leq 2^4\ln(2)) \\
\frac{1}{2}4^2 2^4\ln(2)10^6 &\leq N \\
\implies N_{clr} &\gtrsim 88704000
\end{aligned}
$$

Testing the Composite Left Rectangle Rule for $N = 88704000$ sub-intervals, the routine outputs the approximation $I(f) = 21.640425275139684$, with error estimate $E(f) = 3.3819476641383517e-07$.

$$
\begin{aligned}
|E_{ctr}(f)| = \frac{1}{12}(b-a)H^2|f''(\xi)| &\leq 10^{-6} \\
\frac{1}{12}(4)\frac{4^2}{N^2}2^4\ln^2(2) &\leq 10^{-6} \qquad (|f''(x)| \leq 2^4\ln^2(2)) \\
\frac{1}{12}4^3 2^4\ln^2(2)10^6 &\leq N^2 \\
\implies N_{ctr} &\gtrsim 6403
\end{aligned}
$$

Testing the Composite Trapezoidal Rule for $N = 6403$ sub-intervals, the routine outputs the approximation $I(f) = 21.640425951468153$, with error estimate

$E(f) = -3.3813370237112395e-07.$

$$
\begin{aligned}
|E_{csfr}(f)| = \frac{1}{90}(b-a)H^4|f^{(4)}(\xi)| &\leq 10^{-6} \\
\frac{1}{90}(4)\frac{4^4}{N^4}2^4\ln^4(2) &\leq 10^{-6} \qquad (|f^{(4)}(x)| \leq 2^4\ln^4(2)) \\
\frac{1}{90}4^5 2^4\ln^4(2)10^6 &\leq N^4 \\
\implies N_{csfr} &\gtrsim 81
\end{aligned}
$$

Testing the Composite Simpson's Rule for $N = 81$ sub-intervals, the routine outputs the approximation $I(f) = 21.381558439908844$, with error estimate $E(f) = 0.2588671734256067$.
For the composite open interval quadrature methods, the desired number of sub-intervals (derived from the composite error expressions) to achieve an error of $\leq 10^{-6}$ are:

$$
\begin{aligned}
|E_{cmpr}(f)| = \frac{1}{24}(b-a)H^2|f''(\xi)| &\leq 10^{-6} \\
\frac{1}{24}(4)\frac{4^2}{N^2}2^4\ln^2(2) &\leq 10^{-6} \qquad (|f''(x)| \leq 2^4\ln^2(2)) \\
\frac{1}{24}4^3 2^4\ln^2(2)10^6 &\leq N^2 \\
\implies N_{cmpr} &\gtrsim 4527
\end{aligned}
$$

Testing the Composite Midpoint Rule for $N = 4527$ sub-intervals, the routine outputs the approximation $I(f) = 21.640425275110385$, with error estimate $E(f) = 3.3822406564354424e-07$.

$$
\begin{aligned}
|E_{c2pr}(f)| = \frac{1}{2}(b-a)H^2|f''(\xi)| &\leq 10^{-6} \\
\frac{1}{2}(4)\frac{4^2}{N^2}2^4\ln^2(2) &\leq 10^{-6} \qquad (|f''(x)| \leq 2^4\ln^2(2)) \\
\frac{1}{2}4^3 2^4\ln^2(2)10^6 &\leq N^2 \\
\implies N_{ctr} &\gtrsim 218
\end{aligned}
$$

Testing the Composite Two-Point Rule for $N = 218$ sub-intervals, the routine outputs the approximation $I(f) = 21.64032837902739$, with error estimate $E(f) = 9.723430705932401e-05$.

$$|E_{cgl}| = \frac{1}{270}(b-a)|f^{(4)}(\xi)| \leq \frac{1}{270}(4)(2^4\ln^4(2)) \approx 0.05471647$$

We test the Gauss-Legendre two-point method for $N = 1$ sub-interval. The approximation for the Composite Gauss-Legendre method is $I(f) = 21.404323724978326$, with exact error estimate $E(f) = 0.23610188835612433$.

We test the composite (and adaptive composite) methods, including the desired number of sub-intervals (if possible):

Table 25: Approximations using Open Interval Methods

N	Midpoint ($n = 1$)	$E(f)$	2-Point ($n = 2$)	$E(f)$	Gauss-Legendre ($n = 2$)	$E(f)$
5	21.36563333	0.27479229	21.45694436	0.18348125	21.63995656	0.00046905
10	21.57126599	0.06915962	21.59430116	0.04612446	21.64039608	0.00002953
20	21.62310664	0.01731897	21.62887850	0.01154711	21.64042376	0.00000185
40	21.63609405	0.00433156	21.63753783	0.00288778	21.64042550	0.00000012
80	21.63934261	0.00108300	21.63970361	0.00072201	21.64042561	0.00000001

Table 26: Approximations using Closed Interval Methods

N	Left Rectangle ($n = 1$)	$E(f)$	Trapezoidal ($n = 2$)	$E(f)$	Simpson's ($n = 3$)	$E(f)$
5	16.19212219	5.44830342	22.19212219	-0.55169658	18.53828561	3.10214001
10	18.77887776	2.86154785	21.77887776	-0.13845215	21.64112962	-0.00070400
20	20.17507188	1.46535374	21.67507188	-0.03464626	21.64046992	-0.00004430
40	20.89908926	0.74133635	21.64908926	-0.00866365	21.64042839	-0.00000277
80	21.26759165	0.37283396	21.64259165	-0.00216604	21.64042579	-0.00000017

Table 27: Approximations using Adaptive Midpoint Method

N	I_n	$I - I_n$	Error Estimate
1	16.00000000	5.50015418	5.64042561
3	20.88902593	0.74932001	0.75139968
9	21.55508817	0.08531120	0.08533745
27	21.63092035	0.00950494	0.00950527
81	21.63936918	0.00105643	0.00105643
243	21.64030823	0.00011738	0.00011738
729	21.64041257	0.00001304	0.00001304
2187	21.64042416	0.00000145	0.00000145
6561	21.64042545	0.00000016	0.00000016

Table 28: Approximations using Adaptive Trapezoidal Method

N	I_n	$I - I_n$	Error Estimate
1	34.00000000	-12.00000000	-12.35957439
2	25.00000000	-3.33333333	-3.35957439
4	22.50000000	-0.85786438	-0.85957439
8	21.85660172	-0.21606808	-0.21617610
16	21.69455065	-0.05411827	-0.05412504
32	21.65396195	-0.01353591	-0.01353634
64	21.64381002	-0.00338438	-0.00338440
128	21.64127173	-0.00084612	-0.00084612
256	21.64063714	-0.00021153	-0.00021153
512	21.64047850	-0.00005288	-0.00005288
1024	21.64043883	-0.00001322	-0.00001322
2048	21.64042892	-0.00000331	-0.00000331
4096	21.64042644	-0.00000083	-0.00000083

6 Appendix

6.1 Derivation of Error Expressions

Detailed mathematical explanation of the error expressions for chosen numerical approximation methods. For closed interval methods, let $h = \frac{b-a}{n}$. For open interval methods, let $h = \frac{b-a}{n+2}$.

6.1.1 Composite Left Rectangle Rule Error

This is a closed interval method. The Left Rectangle Rule over an interval $[a, b]$ requires only the left endpoint, $x = a$. Let $h = b-a$. Applying Taylor expansion around $x = a$, the exact definite integral for the method is defined by

$$\begin{aligned}\mathcal{I}(f) &= \int_a^b f(x)dx \\ &= \int_a^b [f(a) + (x-a)f'(\eta)]dx \\ &= (b-a)f(a) + \frac{(b-a)^2}{2}f'(\zeta) \\ &= hf(a) + \frac{h^2}{2}f'(a) + O(h^3)\end{aligned}$$

The numerical approximation for the method is

$$I_0(f) = (b-a)f(a) = hf(a). \tag{30}$$

Therefore

$$E_0 = \mathcal{I}(f) - I_0(f) = \frac{h^2}{2}f'(a) + O(h^3). \tag{31}$$

For the Composite Left Rectangle Rule over N sub-intervals $[a, b]$, we take $H = \frac{b-a}{N}$ and sum the error over each sub-interval:

$$\begin{aligned}E = \sum_{i=0}^{N-1} E_0 &= \sum_{i=0}^{N-1} \left[\frac{H^2}{2}f'(a) + O(H^3)\right] \\ &= N\frac{H^2}{2}f'(\xi) + O(NH^3) \\ &= \frac{1}{2}N\left(\frac{b-a}{N}\right)^2 f'(\xi) + O\left(N\left(\frac{b-a}{N}\right)^3\right) \\ &= \frac{1}{2}(b-a)Hf'(\xi) + O(H^2).\end{aligned}$$

6.1.2 Composite Open Newton-Cotes Two-Point Rule Error

This is an open interval method. The Newton-Cotes Two-Point Rule over a closed interval $[a, b]$ requires the two equally spaced points (i.e. trisection

points). Let $h = \frac{b-a}{3}$ and $x_1 = a + h$, $x_2 = a + 2h$. Define $x = x_1 + sh$, for $-1 \leq s \leq 2$. Denote $f(x_i) = f_i$ and apply Taylor expansion around x_1. The exact definite integral for the method is

$$\begin{aligned}
\mathcal{I}(f) &= \int_a^b \left[f_1 + (x - x_1) f_1' + \frac{(x - x_1)^2}{2} f_1'' + \frac{(x - x_1)^3}{3!} f'''(\eta) \right] dx \\
&= h \int_{-1}^2 \left[f_1 + sh f_1' + \frac{s^2 h^2}{2} f_1'' + \frac{s^3 h^3}{3!} f'''(\eta) \right] ds \\
&= h f_1 [s]_{-1}^2 + \frac{h^2}{2} f_1' [s^2]_{-1}^2 + \frac{h^3}{6} f_1'' [s^3]_{-1}^2 + O(h^4) \\
&= 3h f_1 + \frac{3h^2}{2} f_1' + \frac{9h^3}{6} f_1'' + O(h^4) \\
&= 3h f_1 + \frac{3h^2}{2} f_1' + \frac{3h^3}{2} f_1'' + O(h^4).
\end{aligned}$$

The numerical approximation for the method is

$$\begin{aligned}
I_1(f) &= \frac{3h}{2} [f_1 + f_2] \\
&= \frac{3h}{2} [f_1 + f_1 + h f_1' + \frac{h^2}{2} f_1'' + \frac{h^3}{6} f'''(\eta)] \\
&= 3h f_1 + \frac{3h^2}{2} f_1' + \frac{3h^3}{4} f_1'' + O(h^4).
\end{aligned}$$

Therefore

$$E_1 = \mathcal{I}(f) - I_1(f) = \left(\frac{3}{2} - \frac{3}{4} \right) \left(\frac{b-a}{3} \right)^3 f''(x_i) + O(h^4) \tag{32}$$

$$= \frac{1}{36} (b-a)^3 f''(x_i) + O((b-a)^4). \tag{33}$$

For the Composite Newton-Cotes Two-Point Rule over N sub-intervals $[a, b]$, we take $H = \frac{b-a}{N}$ and sum the error over each sub-interval:

$$\begin{aligned}
E = \sum_{i=1}^N E_1 &= \sum_{i=1}^N \left[\frac{1}{36} H^3 f''(x_i) + O(H^4) \right] \\
&= \frac{1}{36} N H^3 f''(\xi) + O(N H^4) \\
&= \frac{1}{36} (b-a) H^2 f''(\xi) + O(H^3).
\end{aligned}$$

6.1.3 Composite Two-Point Gauss-Legendre Error

For the $2n$-point Gauss-Legendre quadrature rule on $[-1, 1]$, one standard result for the error is given by the following formula:

$$E_{[-1,1]}(f) = \int_{-1}^1 f(t) dt - \sum_{j=1}^{2n} \gamma_j f(t_j) = \frac{2^{2n+3} [(n+1)!]^4}{(2n+3)[(2n+2)!]^3} f^{(2n+2)}(\xi), \tag{34}$$

for some $t \in [-1, 1]$. This rule is exact for all polynomials up to degree $2n + 1$. We want to derive the formula for approximating the definite integral over an arbitrary interval $[a, b]$, i.e. we want to approximate

$$\int_a^b f(x)dx.$$

Applying the linear change of variables (24), we can set

$$f(x) = f\left(\frac{b-a}{2}t + \frac{b+a}{2}\right),$$

such that $t \mapsto x$ for $x \in [a, b]$, $t \in [-1, 1]$. Define the transformed function by $g(t) = f\left(\frac{b-a}{2}t + \frac{b+a}{2}\right)$. The error on the interval $[a, b]$ is therefore given by

$$E_{[a,b]}(f) = \frac{b-a}{2}\left[\int_{-1}^{1} g(t)dt - \sum_{j=0}^{2n} \gamma_j g(t_i)\right] = \frac{b-a}{2}E_{[-1,1]}(g). \tag{35}$$

Since we observe the two-point Gauss-Legendre quadrature rule, we choose $n = 1$ and simplify to get

$$\begin{aligned} E_{[a,b]}(f) &= \left(\frac{b-a}{2}\right)\frac{2^5(2!)^4}{(5)(4!)^3}f^{(4)}(\xi) \\ &= \frac{1}{270}(b-a)f^{(4)}(\xi) \end{aligned}$$

To derive the Composite Gauss-Legendre quadrature error, we partition $[a, b]$ into N sub-intervals (a_i, b_i), $i = 1, 2, \ldots, N$. Define the size of each sub-interval to be $H = \frac{b-a}{N}$ and sum the error expressions over each interval. The composite error expression is

$$\begin{aligned} E &= \sum_{i=1}^{N}\left[\frac{H}{270}f^{(4)}(\xi)\right] \\ &= \frac{1}{270}NHf^{(4)}(\eta) \\ &= \frac{1}{270}(b-a)f^{(4)}(\eta). \end{aligned}$$

References

[1] Alfio Quarteroni and Fausto Saleri, "Numerical Methods for Scientists and Engineers," *Springer*, 2000. https://link.springer.com/book/10.1007/978-3-662-04166-0.

[2] M. B. Rivas, *On the Convergence of Some Cubic Spline Interpolation Schemes*, *SIAM Journal on Numerical Analysis*, Vol. 23, No. 4 (Aug. 1986), pp. 903–912.

[3] Dr. Kyle A. Gallivan, "Set 23: Interpolatory Numerical Quadrature Part 1", Class Lecture, MAD5403: Foundations of Computational Mathematics II, Florida State University.

[4] Dr. Kyle A. Gallivan, "Set 24: Interpolatory Numerical Quadrature Part 2", Class Lecture, MAD5403: Foundations of Computational Mathematics II, Florida State University.

[5] Dr. Kyle A. Gallivan, "Set 25: Interpolatory Numerical Quadrature Part 3", Class Lecture, MAD5403: Foundations of Computational Mathematics II, Florida State University.

[6] Dr. Kyle A. Gallivan, "Set 26: Interpolatory Numerical Quadrature Part 4", Class Lecture, MAD5403: Foundations of Computational Mathematics II, Florida State University.

[7] "Newton-Cotes Formulas," *Wikipedia, The Free Encyclopedia*, https://en.wikipedia.org/wiki/Newton-Cotes_formulas.

[8] "Numerical Integration," *Wikipedia, The Free Encyclopedia*, https://en.wikipedia.org/wiki/Numerical_integration.

[9] 'Riemann Sum," *Wikipedia, The Free Encyclopedia*, https://en.wikipedia.org/wiki/Riemann_sum.

[10] 'Gaussian Quadrature," *Wikipedia, The Free Encyclopedia*, https://en.wikipedia.org/wiki/Gaussian_quadrature#Change_of_interval.

[11] LibreTexts. (n.d.). *7.05: Gauss quadrature rule of integration.* In *Numerical Methods with Applications (Kaw)*. Retrieved April 13, 2025, from https://math.libretexts.org/Workbench/Numerical_Methods_with_Applications_(Kaw)/7%3A_Integration/7.05%3A_Gauss_Quadrature_Rule_of_Integration

[12] McQuarr, B. (n.d.). *Lecture: Composite quadrature.* Retrieved April 13, 2025, from https://personal.morris.umn.edu/~mcquarrb/teachingarchive/M4401/Lectures/LectureCompositeQuadrature.pdf

[13] Xu, Z. (n.d.). *Composite integration.* Retrieved April 13, 2025, from https://www3.nd.edu/~zxu2/acms40390F11/sec4-4-Composite-integration.pdf