Numerical Integration I

Mark A. Austin

University of Maryland

austin@umd.edu
ENCE 201 Fall Semester 2023

April 10, 2024

Overview

Mathematical Questions

1. How do we evaluate:

$$I = \int_a^b f(x)dx? \tag{1}$$

2. Calculus tells us that the antiderivative of a function f(x) over an interval [a,b] is:

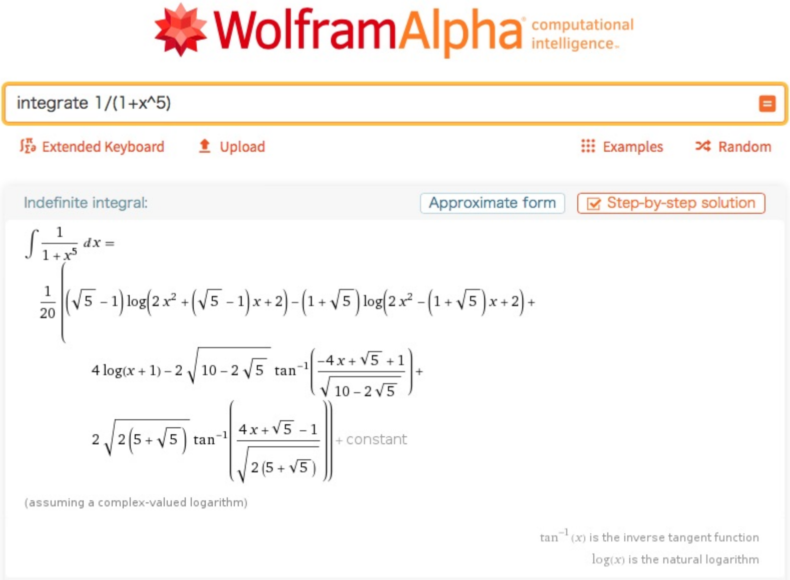
$$I = \int_a^b f(x)dx = [F(x)]_a^b = F(b) - F(a). \tag{2}$$

3. Many integrals cannot be evaluated using this approach, e.g.,

$$I = \int_0^1 \frac{1}{1+x^5}dx \tag{3}$$

has a very complicated antiderivative.

Mathematical Questions



Mathematical Questions

Idea: Let's replace the original function by a new function that is much easier to work with, i.e.,

$$I = \int_a^b f(x)dx \approx \int_a^b \tilde{f}(x)dx = \tilde{I}. \tag{4}$$

We want $\tilde{f}(x)$ to be a good approximation of $f(x)$.

Basic Questions:

1. What strategies exist for choosing and integrating $\tilde{f}(x)$?
2. How much computational work is needed to obtain a required level of accuracy?

General Framework

The approximation error is as follows:

$$\text{Error} = \int_a^b \left[f(x) - \tilde{f}(x) \right] dx \leq (b-a) \max_{a \leq \xi \leq b} \| f(\xi) - \tilde{f}(\xi) \|.$$

This inequality tells us the approximation error E depends on two factors:

- The width of the integration interval (b-a).
- The maximum difference between $f(\xi)$ and $\tilde{f}(\xi)$ within the interval $a \leq \xi \leq b$.

Basic

Numerical Methods

Basic Numerical Methods

Basic approaches to numerical integration:

1. Polynomial Approximation
2. Rectangular and Midpoint Rules
3. Trapezoid Rule
4. Simpson's Rule

Composite methods:

1. Composite Trapezoid Rule
2. Composite Simpson's Rule

Polynomial Approximation

Strategy: Choose an approximation $\tilde{f}(x)$ to $f(x)$ that is easily integrable and a good approximation to $f(x)$

Two candidate schemes:

1. Interpolation polynomials approximating f(x).
2. Taylor series approximation of f(x).

Note: In order for the Taylor series approximation to work, we need the functional derivatives at "a" to exist.

Polynomial Interpolation

Example 1: Consider the integral: $I = \int_0^{\pi} sin(x)dx$.

Analytic Solution.

$$I = \int_0^{\pi} sin(x)dx = [-cos(x)]_0^{\pi} = 2.0. \tag{5}$$

Polynomial Interpolation

Consider the data set (3 data points):

x	0.0	$\pi/2$	π
sin(x)	0.0	1.0	0.0

A quadratic fit will have roots at $x = 0$ and $x = \pi$, and pass through the point $sin(\pi/2) = 1.0$.

Polynomial Interpolation

So let:

$$p(x) = Ax\,(x-\pi)\,. \tag{6}$$

and determine the value of A by applying the constraint $sin(\pi/2)$ $= 1.0$.

$$p(\pi/2) = A\pi/2\,(\pi/2-\pi) = 1.0 \rightarrow A = -4/\pi^2. \tag{7}$$

Integration

$$I = \int_0^{\pi} sin(x)dx \approx \left[\frac{-4}{\pi^2}\right] \int_0^{\pi} x\,(x-\pi)\,dx = 2.09. \tag{8}$$

The relative error is 4.5%. Not bad.

Polynomial Approximation

Example 2: Consider the integral:

$$I = \int_0^1 e^{x^2} dx \tag{9}$$

The Taylor series approximation of $f(x)$ is:

$$f(x) = 1 + t + \frac{t^2}{2!} + \frac{t^3}{3!} + \cdots \frac{t^{(n+1)}}{(n+1)!} e^c, \quad \text{where} \quad t = x^2. \tag{10}$$

and c is a constant $0 \leq c \leq 1$.

Polynomial Approximation

Solution:

$$I = \int_0^1 \left[1 + x^2 + \frac{x^4}{2!} + \frac{x^6}{3!} + \cdots \frac{x^{2n}}{n!}\right] dx + \int_0^1 \left[\frac{x^{2n+2}}{(n+1)!}\right] e^c dx. \tag{11}$$

Let n = 3. We have

$$I = 1 + \frac{1}{3} + \frac{1}{10} + \frac{1}{42} + Error = 1.4571 + Error. \tag{12}$$

An upper bound on the numerical error is:

$$\text{Error} \leq \frac{e}{24} \int_0^1 x^8 dx = \frac{e}{216} = 0.0126. \tag{13}$$

Polynomial Approximation

Difficulties with Polynomial Approximation:

- Taylor series approximations only work well when higher order derivatives exist.

  This excludes functions that are continuous, but are not continuously differentiable. (e.g., $f(x) = |x|$ is continuous, but not differentiable at x = 0).

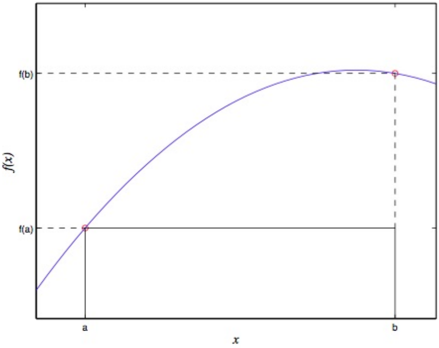
- Some Taylor series converge too slowly to get a reasonable approximation by just a few terms of the series.

  As a rule, if the series has a factorial in the denominator, this technique will work efficiently, otherwise, it will not.

Basic Interpolation Methods

Rectangular Interpolation

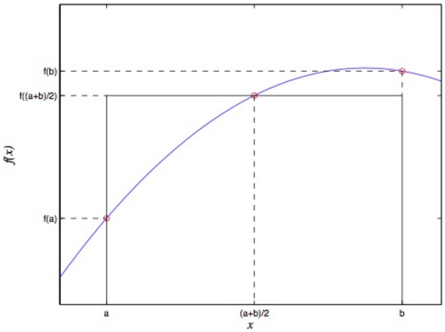
$$I = \int_a^b f(x)dx \approx (b-a)f(a) = \tilde{I}. \tag{14}$$



Basic Interpolation Methods

Midpoint Interpolation

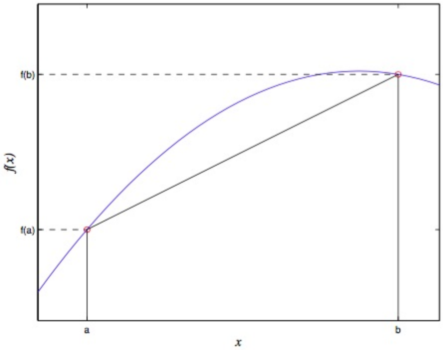
$$I = \int_a^b f(x)dx \approx (b-a)f(\frac{a+b}{2}) = \tilde{I}. \quad (15)$$



Basic Interpolation Methods

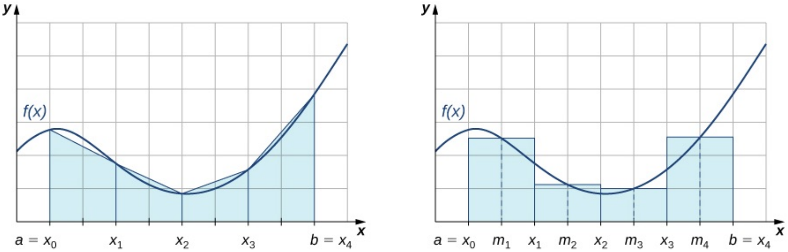
Trapezoid Interpolation

$$I = \int_a^b f(x)dx \approx \frac{(b-a)}{2}\left[f(a) + f(b)\right] = \tilde{I}. \tag{16}$$



Basic Interpolation Methods

Observation: The midpoint rule tends to be more accurate than the trapezoid rule:



When we get to error analysis we will see that, in fact, this is true!

Trapezoid Rule

Trapezoidal Rule

Sketch of Derivation: Let the interval of integration be defined by $h = (b - a)$, and two end points: $(a, f(a))$ and $(b, f(b))$.

Linear Polynomial Fit:

$$p(x) = f(a) + \left[\frac{f(b) - f(a)}{h}\right](x - a) \tag{17}$$

Integrate p(x), then simplify:

$$T = \int_a^b p(x)dx = |f(a)x + \left[\frac{f(b) - f(a)}{h}\right] \cdot \frac{(x-a)^2}{2}|_a^b$$
$$= \frac{h}{2}\left[f(a) + f(b)\right].$$

Composite Trapezoidal Rule

Definition: Assume that $f(x)$ is continuous over an interval [a,b]. Let n be a positive integer and $h = (b - a)/n$.

Next, let's divide [a,b] into n subintervals, each of length h, with endpoints at $P = [x_0, x_1, x_2, \cdots, x_n]$.

We set:

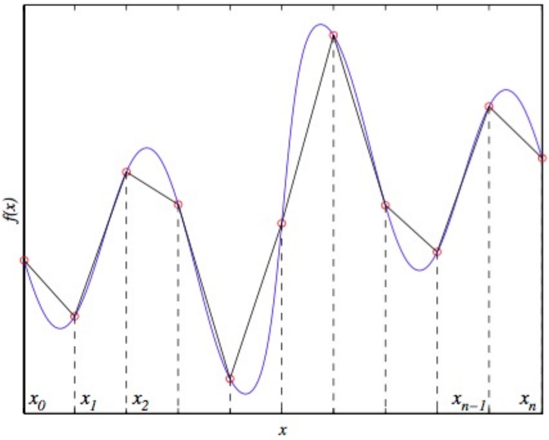
$$T_n = \frac{h}{2}\left[f(x_0) + 2f(x_1) + \cdots + 2f(x_{(n-1)}) + f(x_n)\right] \qquad (18)$$

Note: As n increases toward infinity,

$$\operatorname*{limit}_{n\to\infty} T_n = \int_a^b f(x)dx. \qquad (19)$$

Composite Trapezoidal Rule

Visual Representation



Composite Trapezoidal Rule

Error Analysis

$$I = \int_a^b f(x)dx = T_n - \frac{|f^2(\xi)|}{12} h^2(b-a). \qquad (20)$$

where $[a \leq \xi \leq b]$. The method is $O(h^2)$ accurate.

Example 1. Error Analysis for $\int_0^2 x^2 dx$. Does equation 20 work?

Analytical Solution:

$$I = \int_0^2 x^2 dx = \left[\frac{1}{3}x^3\right]_0^2 = \frac{8}{3}. \qquad (21)$$

Trapezoidal Rule

One Step of Trapezoid: (here h = 2, b-a = 2)

$$\int_0^2 x^2 dx \to T_1 = \frac{h}{2}[f(0) + f(2)] = 4.0. \tag{22}$$

Theoretical Error Estimate: f(x) = x^2, $\frac{df}{dx} = 2x$, $\frac{d^2 f}{dx^2} = 2$.

$$\text{Error} \leq \frac{|f^2(\xi)|}{12} h^2 (b-a) \to \frac{2 \cdot 2^2 \cdot 2}{12} = \frac{16}{12} = 1.33. \tag{23}$$

Actual Error:

$$\text{Absolute Error} = |\text{Exact} - \text{Trapezoid}| = |8/3 - 4| = 1.33. \checkmark \tag{24}$$

Trapezoidal Rule

Example 2. Evaluate $I = \int_0^4 xe^{2x}dx$.

Analytic Solution.

$$I = \int_0^4 xe^{2x}dx = \left[\frac{x}{2}e^{2x} - \frac{1}{4}e^{2x}\right]_0^4 = 5,216.92. \qquad (25)$$

One Step of Trapezoid Rule (n = 1).

$$T_1 = \left[\frac{4-0}{2}\right][f(0) + f(4)] = 23,847.66. \qquad (26)$$

Not very accurate at all!

Composite Trapezoidal Rule

Example 3. Evaluate $I = \int_0^4 xe^{2x}dx$ with two segments (n = 2).

Solution. We have:

$$
\begin{aligned}
I &= \int_0^4 xe^{2x}dx = \int_0^2 xe^{2x}dx + \int_2^4 xe^{2x}dx \\
&\approx \left[\frac{2-0}{2}\right][f(0)+f(2)] + \left[\frac{4-2}{2}\right][f(2)+f(4)] \\
&= [f(0)+2f(2)+f(4)] = [0+4e^4+4e^8] \\
T_2 &= [0+4e^4+4e^8] = 12,142.22.
\end{aligned}
$$

Answer is much better than one step, but still very poor accuracy.

Composite Trapezoidal Rule

Test Program Source Code:

```
# ============================================================================
# TestIntegrationTrapezoid01.py: Use trapezod algorithm to integrate functions.
#
# Written By: Mark Austin                                            July 2023
# ============================================================================

import math;
import Integration;

def f2(x):
    return x*math.exp(2 * x)

# main method ...

def main():
    print("--- ");
    print("--- Case Study 2: Integrate x*math.exp(2x) over [0, 4] ... ");
    print("--- ================================================== ... ");

    # Initialize problem setup ...

    a = 0.0;
    b = 4.0
    nointervals = 2

    print("--- Inputs:")
```

Composite Trapezoidal Rule

Test Program Source Code: Continued ...

```
    print("---   a = {:9.4f} ...".format(a) )
    print("---   b = {:9.4f} ...".format(b) )
    print("---   no intervals = {:d} ...".format(nointervals) )

    # Compute numerical solution to integral ..

    print("--- Execution:")
    xi = Integration.trapezoid( f2, a, b, nointervals )

    # Summary of computations ...

    print("--- Output:")
    print("---   integral = {:12.4f} ...".format( xi ) )

# call the main method ...

main()
```

Composite Trapezoidal Rule

Abbreviated Output:

```
--- Case Study 2: Integrate x*math.exp(2x) over [0, 4] ...
--- ======================================== ...
--- Inputs:
---   a =    0.0000 ...
---   b =    4.0000 ...
---   no intervals = 2 ...
--- Execution:
--- Output:
---   integral =   12142.2245 ...

--- Case Study 2: Integrate x*math.exp(2x) over [0, 4] ...
--- ======================================== ...
--- Inputs:
---   a =    0.0000 ...
---   b =    4.0000 ...
---   no intervals = 4 ...
--- Execution:
--- Output:
---   integral =   7288.7877 ...
```

Composite Trapezoidal Rule

Systematic Refinement: $T_1, T_2, \cdots, T_{512}$:

No Intervals	h	Integral T_n
1	4.0	$T_1 = 23{,}847.66$
2	2.0	$T_2 = 12{,}142.22$
4	1.0	$T_4 = 7{,}288.79$
8	0.5	$T_8 = 5{,}764.76$
16	0.25	$T_{16} = 5{,}355.94$
32	0.125	$T_{32} = 5{,}251.81$
64	0.0625	$T_{64} = 5{,}225.81$
128	0.0312	$T_{128} = 5{,}219.10$
256	0.0156	$T_{256} = 5{,}217.47$
512	0.0078	$T_{512} = 5{,}217.06$

Key Takeaway: Trapezoid works, but convergence is very slow.

Composite Trapezoidal Rule

Example 4. Use the Trapezoid rule with $n = 10$ segments to approximate $\int_0^\pi e^x \cos x dx$.

Determine the absolute true error $|E_t|$, and compare it with the true-error bound provided above.

Solution. We have

$$\Delta x = (b - a)/n = (\pi - 0)/10 = 0.314159,$$

and $\Delta x/2 = 0.157080$.

Moreover $x_0, x_1, \ldots, x_{10}$ satisify $x_i = a + i\Delta x = i\Delta x$, forall $i = 0, 1, \ldots, 10$.

Hence,

$$x_0 = 0, x_1 = 0.314159, x_2 = 0.628319, \cdots, x_{10} = 3.14159.$$

Trapezoidal Rule

Solution Continued. Therefore,

$$\int_0^{\pi} e^x \cos x dx \approx \frac{\Delta x}{2}(e^{x_0}\cos(x_0) + \cdots + e^{x_{10}}\cos(x_{10})) = -12.2695. \tag{27}$$

Error Analysis. The analytical solution is:

$$\int_0^{\pi} e^x \cos x dx = -(1+e^{\pi})/2 = -12.0703. \tag{28}$$

This gives $|E_t| = 0.199199$. Finally, we note $f''(x)$ reaches an absolute minimum value of -14.9210 at $x = 3\pi/4$. And so

$$\text{Worst case error} \leq (14.9210)\pi^3/(12)(10)^2 = 0.385537. \tag{29}$$

Trapezoidal Rule

Example 5. How many intervals are needed to compute:

$$I = \int_0^1 \left[\frac{sin(x)}{x}\right] dx \tag{30}$$

to an accuracy 10^{-8}?

Solution. First, we note $|f^2(\xi)|_{max} = 1/3$.

For the trapezium rule:

$$\text{Error} \leq \frac{1}{12} h^2 |f^2(\xi)|_{max} = \frac{h^2}{36} \leq \frac{10^{-8}}{2}. \tag{31}$$

Hence, $h \leq \sqrt{18} \times 10^{-4}$. We also have $nh = 1$.

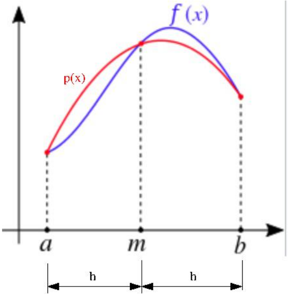
Number of required intervals: $n \geq 2, 357$.

Simpson's Rule

(Thomas Simpson, 1710-1761)

Simpson's Rule

Objective: Approximate the integral of a function by fitting a quadratic function q(x) through three equally spaced points: $[a, f(a)]$, $[m, f(m)]$ and $[b, f(b)]$.



Interval of integration: $[b-a]=2\text{h}$. Midpoint $m=[(a+b)/2]$.

Simpson's Rule

Sketch of Derivation: Suppose that:

$$q(x) = q_0 + q_1(x-a) + q_2(x-a)(x-m) \qquad (32)$$

fits through $[a, f(a)]$, $[m, f(m)]$ and $[b, f(b)]$.

We can use the method of divided differences to show:

$$q_0 = f(a)$$
$$q_1 = (f(m) - f(a))/h$$
$$q_2 = (f(b) - 2f(m) + f(a))/2h^2$$

Simpson's Rule

Sketch of Derivation:

Next, integrate q(x) and simplify. This gives:

$$S = \int_a^b q(x)dx = \frac{h}{3}[f(a) + 4f(m) + f(b)]. \qquad (33)$$

For a single step of Simpon's rule,

$$I = \int_a^b f(x)dx = \int_a^b q(x)dx - \frac{1}{90}\left[\frac{(b-a)}{2}\right]^5 f^4(\xi), \qquad (34)$$

where $[a \leq \xi \leq b]$.

Simpson's Rule

Important Point

Notice that the error depends on the fourth derivative of f(x).

Thus, if f(x) happens to be a polynomial of degree three or less,

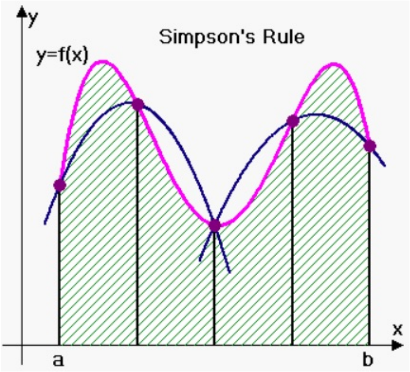
$$f(x) = f_0 + f_1 x + f_2 x^2 + f_2 x^3 \tag{35}$$

then Simpsons rule will give an exact answer, i.e,

$$I = \int_a^b f(x)dx = \int_a^b q(x)dx. \tag{36}$$

Composite Simpson's Rule

Objective: Simply chain together a sequence of simpson rule approximations:



Composite Simpson's Rule

Numerical Formula

$$S_n = \frac{h}{3}\sum_{j=1}^{n/2}[f(x_{2j-1}) + 4f(x_{2j}) + f(x_{2j+1})] \qquad (37)$$

Error Analysis

$$I = \int_a^b f(x)dx = S_n - \frac{h^4}{180}(b-a)|f^4(\xi)| \qquad (38)$$

where $[a \leq \xi \leq b]$ and $h = (b-a)/n$ is the step length. The method is $O(h^4)$ accurate.

Simpson's Rule

Example 1. Consider the integral: $\int_0^\pi sin(x)dx$.

Applying Simpson's Rule to the data set:

x	0.0	$\pi/2$	π
sin(x)	0.0	1.0	0.0

gives:

$$S = \frac{\pi/2}{3}[f(0) + 4f(\pi/2) + f(\pi)] = \frac{\pi}{6}[0 + 4 * 1 + 0]. \quad (39)$$

which, by coincidence, is identical to the quadratic polynomial approximation.

Simpson's Rule

Now let's extend the data set from 3 to 5 points:

x	0.0	$\pi/4$	$\pi/2$	$3\pi/4$	π
sin(x)	0.0	$1/\sqrt{2}$	1.0	$1/\sqrt{2}$	0.0

Applying Simpson's Rule for four intervals:

$$
\begin{aligned}
S_4 &= \frac{\pi/4}{3}[f(0) + 4f(\pi/4) + 2f(\pi/2) + 4f(3\pi/4) + f(\pi)] \\
&= \frac{\pi}{12}\left[0.0 + 4/\sqrt{2} + 2 * 1.0 + 4/\sqrt{2} + 0.0\right] \\
&= \frac{\pi}{12}\left[2.0 + 8/\sqrt{2}\right] \\
&= 2.0045.
\end{aligned}
$$

Simpson's Rule

Estimate of Maximum Absolute Error:

$$\text{Maximum Error} \leq \frac{h^4}{180}(b-a)|f^4(\xi)| \tag{40}$$

We have: $f(x) = sin(x) \rightarrow f^4(\xi) = sin(\xi) \leq 1.0.$

The interval $(b-a) = \pi$ and $h = \pi/4$. Thus, we estimate:

$$\text{Maximum Error} \leq \frac{(\pi/4)^4}{180}\pi = \left[\frac{\pi^5}{16x16x180}\right] = 0.0066. \tag{41}$$

Actual error $= 0.0045.$

Composite Simpson's Rule

Systematic Refinement: $S_2, S_4, \cdots, S_{32}$:

No Intervals	h	Integral S_n
2	$\pi/2$	$S_2 = 2.0944$
4	$\pi/4$	$S_4 = 2.0045$
8	$\pi/8$	$S_8 = 2.00027$
16	$\pi/16$	$S_{16} = 2.00002$
32	$\pi/32$	$S_{32} = 2.000001$

Key Takeaway: Simpson's Rule converges much faster than Trapezoid ...

Simpson's Rule

Example 2. Evaluate $I = \int_0^4 xe^{2x} dx$.

Analytic Solution.

$$I = \int_0^4 xe^{2x} dx = \left[\frac{x}{2}e^{2x} - \frac{1}{4}e^{2x}\right]_0^4 = 5,216.92. \tag{42}$$

Systematic Refinement: S_2, S_4, $\cdots$, S_{32}:

No Intervals	h	Integral S_n
2	2	S_2 = 8,240.41
4	1	S_4 = 5,670.97
8	0.5	S_8 = 5,256.75
16	0.25	S_{16} = 5,219.67
32	0.125	S_{32} = 5,217.10

Simpson's Rule

Example 3. How many intervals are needed to compute:

$$I = \int_0^1 \left[\frac{sin(x)}{x}\right] dx \qquad (43)$$

to an accuracy 10^{-8}?

Solution. For the Simpson's Rule:

$$\text{Error} \leq \frac{1}{180} h^4 |f^4(\xi)|_{max} \leq \frac{10^{-8}}{2}. \qquad (44)$$

Number of required intervals: $n \geq 20$.

This is significantly better than Trapezoidal Rule (n = 2,357), but still a lot of work. We need a more efficient method!

Python Code Listings

Code 1: Composite Trapezoid Rule

```
# ======================================================================
# Integration.trapezoid(): Numerical integration of f(x) with
#                          composite trapezoid rule.
#
# Args: f (function): the equation f(x).
#       a (float): the initial point.
#       b (float): the final point.
#       n (int): number of intervals.
#
# Returns:
#       xi (float): numerical approximation of the definite integral.
# ======================================================================

import math
import numpy as np

def trapezoid(f, a, b, n):
    h = (b - a) / n

    sum_x = 0

    for i in range(0, n - 1):
        x = a + (i + 1) * h
        sum_x += f(x)

    xi = h / 2 * (f(a) + 2 * sum_x + f(b))
    return xi
```

Code 2: Composite Simpson's Rule

```
# ================================================================================
# Integration.simpson(): Numerical integration of f(x) with 1/3 Simpson's Rule.
#
# Args: f (function): the equation f(x).
#       a (float): the initial point.
#       b (float): the final point.
#       n (int): number of intervals.
#
# Returns:
#       xi (float): numerical approximation of the definite integral.
# ================================================================================

import math
import numpy as np

def simpson(f, a, b, n):
    h = (b - a) / n

    sum_odd  = 0
    sum_even = 0

    for i in range(0, n - 1):
        x = a + (i + 1) * h
        if (i + 1) % 2 == 0:
            sum_even += f(x)
        else:
            sum_odd += f(x)

    xi = h / 3 * (f(a) + 2 * sum_even + 4 * sum_odd + f(b))
    return xi
```