PRACTICE 8: Numerical Integration II

1 Open Formulas for Numerical Integration

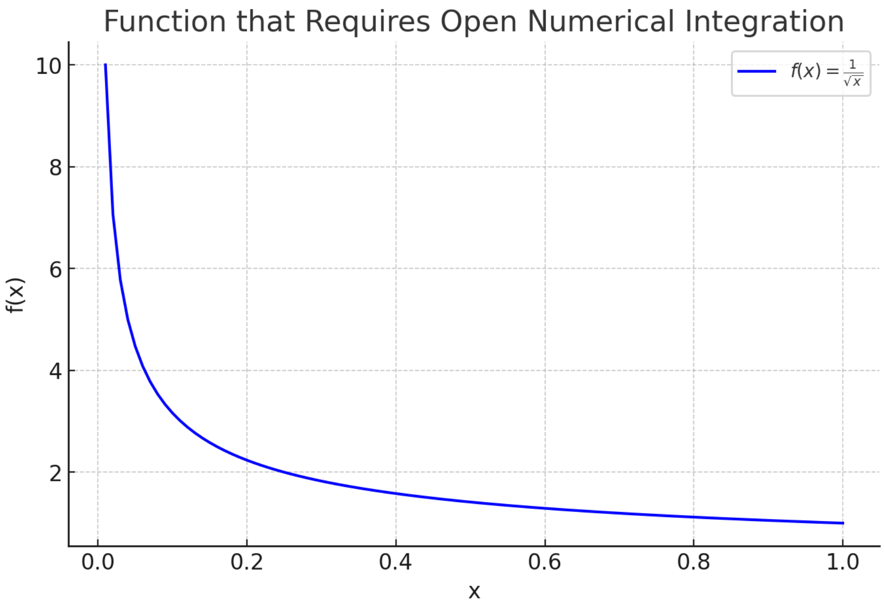
Open numerical integration formulas approximate the integral of a function $f(x)$ without using the function values at the endpoints of the integration interval. These formulas are useful when the values of $f(x)$ at the limits are undefined or difficult to evaluate.

Some functions cannot be numerically integrated using the standard Trapezoidal or Simpson's rule because they are undefined at the interval endpoints. One such example is:

$$f(x) = \frac{1}{\sqrt{x}}$$

in the interval $[0, 1]$. The function is not defined at $x = 0$, making the use of closed numerical integration methods impossible. In this case, **open numerical formulas** are required, as they do not evaluate the function at the endpoints.

The graph below illustrates the function $f(x) = \frac{1}{\sqrt{x}}$ in the interval $[0, 1]$, avoiding $x = 0$ to prevent singularity.



The Exact Value of the Integral: We evaluate the definite integral from 0 to 1:

$$I = \lim_{\epsilon \to 0^+} \int_{\epsilon}^{1} \frac{1}{\sqrt{x}}\, dx = \lim_{\epsilon \to 0^+} \int_{\epsilon}^{1} x^{-1/2}\, dx = \lim_{\epsilon \to 0^+} \left[\frac{x^{1/2}}{1/2} \right]_{\epsilon}^{1} = 2 \lim_{\epsilon \to 0^+} \left[\sqrt{x} \right]_{\epsilon}^{1} = 2.$$

This function demonstrates why open numerical integration is essential in cases where standard methods fail.

1.1 Simple Open Trapezoidal Formula

The open trapezoidal formula approximates the function $f(x)$ using a straight line (linear interpolation) within an internal interval and then computes the integral of this approximation.

To integrate $f(x)$ over the interval $[a, b]$ without evaluating the function at the endpoints, we use intermediate points x_1 and x_2:

$$\int_a^b f(x)dx \approx \frac{b-a}{2}[f(x_1) + f(x_2)]$$

where the points x_1 and x_2 are typically chosen as:

$$x_1 = a + \frac{l}{3}, \quad x_2 = b - \frac{l}{3}$$

with $l = b - a$.

This method has an error of order $O(h^3)$, making it more accurate than the midpoint rule.

The open trapezoidal rule is derived using interpolation over an internal set of points in the interval $[a, b]$, avoiding the endpoints.

1.1.1 Lagrange Interpolation Weights

We approximate the integral:

$$\int_0^1 \varphi(t)dt \approx A_0\varphi(1/3) + A_1\varphi(2/3)$$

Solving for the weights:

$$A_0 = \frac{1}{2}, \quad A_1 = \frac{1}{2}$$

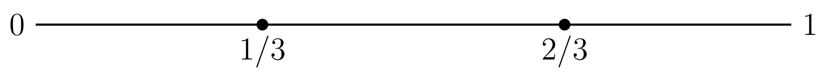
Using the change of variable $t = \dfrac{x-a}{b-a}$ (so $x = a + t(b-a)$ and $dx = b - a$), we transform the integral to:

$$\int_a^b f(x)dx = (b-a)\int_0^1 f(a + (b-a)t)dt$$

Now we select the points as follows:

$$x_0 = a + h,$$
$$x_1 = a + 2h.$$

with $h = \frac{b-a}{3}$, as in the bellow diagram:



which leads to

$$I \approx \frac{b-a}{2}\left[f(x_0) + f(x_1)\right]$$

1.2 Composite Open Trapezoidal Rule

When applying the open trapezoidal rule over multiple subintervals, we divide $[a, b]$ into n equal subintervals of width $h = \frac{b-a}{n}$. The integral is then approximated as:

$$\int_a^b f(x)dx \approx \frac{(b-a)}{2n} \sum_{i=1}^{n} [f(x_{i,1}) + f(x_{i,2})]$$

where for each subinterval $[x_i, x_{i+1}]$, the intermediate points are chosen as:

$$x_{i,1} = x_i + \frac{h}{3}, \quad x_{i,2} = x_{i+1} - \frac{h}{3},$$

with $h = \dfrac{x_{i+1} - x_i}{3}$. This composite approach improves the accuracy of the numerical integration.

1.2.1 Examples in MATLAB

Example 1: Compute the integral of $f(x) = \frac{1}{\sqrt{x}}$ in $[0, 1]$ using the open trapezoidal rule.

Open Trapezoidal Rule Example

```
f = @(x) 1./sqrt(x);
a = 0; b = 1;
fplot(f,[a,b],'m-'),hold on

% Select interior points
h = b-a;
x1 = a + h/3;
x2 = b - h/3;

% Compute integral approximation
I = (b-a)/2 * (f(x1) + f(x2));

fprintf('Integral approximation using Open Trapezoidal Rule: %f\n', I);

% Mark the selected nodes on the plot
plot([x1, x2], f([x1, x2]), 'ro', 'MarkerFaceColor', 'r');
```

Example 2: Compute the integral of $f(x) = e^x$ in $[0, 2]$ using the composite open trapezoidal rule with $n = 5$.

1.3 Simple Open Simpson's Formula

The open **Simple Simpson's Rule** is derived using quadratic interpolation over an internal set of points in the interval $[a, b]$, avoiding the endpoints.

We approximate the integral:

$$\int_0^1 \varphi(t)dt \approx A_0\varphi(1/4) + A_1\varphi(1/2) + A_2\varphi(3/4)$$

Solving for the weights using the conditions:

$$\begin{aligned}
\varphi(t) = 1 \quad &\Rightarrow \quad 1 = A_0 + A_1 + A_2 \\
\varphi(t) = t \quad &\Rightarrow \quad 1/2 = \frac{A_0}{4} + \frac{A_1}{2} + \frac{3A_2}{4} \\
\varphi(t) = t^2 \quad &\Rightarrow \quad 1/3 = \frac{A_0}{16} + \frac{A_1}{4} + \frac{9A_2}{16}
\end{aligned}$$

Solving these equations:

$$A_0 = \frac{2}{3}, \quad A_1 = -\frac{1}{3}, \quad A_2 = \frac{2}{3}$$

For a general interval $[a, b]$, we transform the integral:

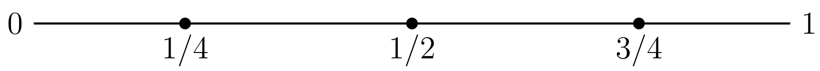
$$\int_a^b f(x)dx = (b-a)\int_0^1 f(a + (b-a)t)dt$$

$$\int_a^b f(x)dx \approx (b-a)\left[\frac{2}{3}f(x_0) - \frac{1}{3}f(x_1) + \frac{2}{3}f(x_2)\right]$$

where the selected points are:

$$\begin{aligned}
x_0 &= a + h, \\
x_1 &= a + 2h, \\
x_2 &= a + 3h.
\end{aligned}$$

with $h = \frac{b-a}{4}$, as in the following diagram:



1.4 Composite Open Simpson's Rule

When higher accuracy is needed, the **Composite Open Simpson's Rule** is used. Instead of applying the open Simpson formula over the entire interval $[a, b]$, we divide it into n equal subintervals and apply the formula in each subinterval:

$$\int_a^b f(x)dx \approx \frac{(b-a)}{n}\sum_{i=1}^{n}\left[\frac{2}{3}f(x_{i,0}) - \frac{1}{3}f(x_{i,1}) + \frac{2}{3}f(x_{i,2})\right]$$

1.4.1 Examples in MATLAB

Example 1: Compute the integral of $f(x) = \frac{1}{\sqrt{x}}$ in $[0, 1]$ using the composite open Simpson's rule.

Open Simpson's Rule Example

```
f = @(x) 1./sqrt(x);
a = 0; b = 1;
h = (b-a)/4;
x0 = a + h;
x1 = a + 2*h;
x2 = a + 3*h;
I = 4*h * ( (2/3)*f(x0) - (1/3)*f(x1) + (2/3)*f(x2) );
fprintf('Integral approximation: %f\n', I);
```

Example 2: Compute the integral of $f(x) = \ln(x)$ in $[0, 2]$ using the composite open Simpson's rule with $n = 100$.

Using the composite method significantly enhances the accuracy of numerical integration.

2 Gauss-Legendre Quadrature

Gauss-Legendre quadrature is a numerical integration method that provides high accuracy by selecting optimal integration points (nodes x_i) and weights w_i as follows:

$$I = \int_{-1}^{1} f(x)dx \approx \sum_{i=1}^{n} w_i f(x_i)$$

- x_i are the Legendre polynomial roots
- w_i are the corresponding weights

Gauss-Legendre Quadrature Weights

The weights w_i in Gauss-Legendre quadrature are designed to ensure that the integral is evaluated exactly for polynomials of degree up to $2n-1$. They are computed using the relationship with Legendre polynomials.

The weights w_i are calculated using the formula:

$$w_i = \frac{2}{(1-x_i^2)[P_n'(x_i)]^2}$$

where:

- x_i are the roots of the Legendre polynomial $P_n(x)$ of degree n.
- $P_n'(x)$ is the derivative of the Legendre polynomial evaluated at x_i.

Change of Variable for a General Interval $[a, b]$

If the integral is defined over a general interval $[a, b]$, instead of $[-1, 1]$, we use the change of variable:

$$x = \frac{b-a}{2}\xi + \frac{a+b}{2}, \quad \text{where } \xi \in [-1, 1].$$

Applying this transformation, the integral becomes:

$$\int_a^b f(x)dx = \frac{b-a}{2}\int_{-1}^{1} f\left(\frac{b-a}{2}\xi + \frac{a+b}{2}\right) d\xi.$$

This transformation affects the weights as well. The new weights for the interval $[a, b]$ are given by:

$$w_i' = \frac{b-a}{2} w_i.$$

Thus, the integral approximation over $[a, b]$ is:

$$I \approx \sum_{i=1}^{n} w_i' f(x_i') = \frac{b-a}{2}\sum_{i=1}^{n} w_i f\left(\frac{b-a}{2}x_i + \frac{a+b}{2}\right).$$

Orthonormality of Legendre Polynomials

Legendre polynomials $P_n(x)$ form an *orthonormal basis* in the interval $[-1, 1]$, meaning they satisfy the orthogonality condition:

$$\int_{-1}^{1} P_m(x)P_n(x)dx = 0, \quad \text{for } m \neq n.$$

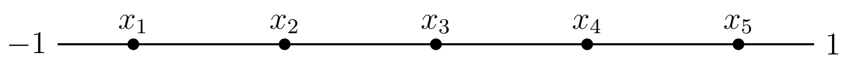
This property ensures that Gauss-Legendre quadrature selects nodes optimally for numerical integration.

Gauss-Legendre Quadrature and Polynomial Degree

The precision of Gauss-Legendre quadrature depends on the number of nodes n. Specifically, using n nodes allows exact integration of polynomials up to degree $2n - 1$.

For example, using **1 node** in Gauss-Legendre quadrature is equivalent to the **midpoint rule**, which is the simplest numerical integration method.

The following diagram illustrates the placement of Gauss-Legendre nodes



2.0.1 MATLAB Examples

Example 1: Gauss-Legendre Quadrature with Two Nodes

Approximating $\int_{-1}^{1} e^x dx$ using Gauss-Legendre quadrature with two nodes:

Example 2: Gauss-Legendre Quadrature with Four Nodes

Approximating $\int_{0}^{1} \ln(1 + x)dx$ using Gauss-Legendre quadrature with four nodes: