Notes on Numerical Integration

Unbloated Contributors

Contents

Preface v

1 Numerical Integration 1

1.1 The Midpoint Rule 1

2 Implementation 3

2.1 A Reference Implementation 3

2.2 Checking the Result 3

A Notation 5

Preface

This short book introduces the midpoint rule and its implementation. The examples use semantic headings, numbered equations, and references that remain valid when chapters move.

Chapter 1

Numerical Integration

1.1 The Midpoint Rule

For a continuous function f, approximate its integral by sampling the midpoint of each subinterval [1]. Appendix A collects the notation.

Definition 1.1 (Composite midpoint rule).
For $n \geq 1$ and $h = (b-a)/n$, define

$$M_n(f) = h \sum_{k=0}^{n-1} f\left(a + \left(k + \frac{1}{2}\right) h\right). \tag{1.1}$$

Theorem 1.2 (Exactness for affine functions). If $f(x) = cx + d$, then $M_n(f) = \int_a^b f(x)\,\mathrm{d}x$.

Proof. On each subinterval, the average of an affine function equals its value at the midpoint. Multiplying by h gives the exact integral on that subinterval. Summing proves the result. □

Example 1.3. For $f(x) = x$ on $[0, 1]$, Equation (1.1) gives $M_n(f) = 1/2$ for every positive integer n.

Exercise 1.1.
Apply the midpoint rule to $f(x) = x^2$ on $[0, 1]$ with $n = 2$. Compare the result with the exact integral $1/3$.

Chapter 2

Implementation

2.1 A Reference Implementation

Listing 2.1 implements Equation (1.1). It rejects nonpositive subinterval counts before computing the step size.

```
def midpoint(f, a, b, n):
    if n < 1:
        raise ValueError("n must be positive")
    h = (b - a) / n
    return h * sum(f(a + (k + 0.5) * h) for k in range(n))
```

Listing 2.1: Composite midpoint rule.

2.2 Checking the Result

Theorem 1.2 provides an exact test case. Table 2.1 instead uses $f(x) = x^2$ to show how the approximation changes when the number of subintervals doubles.

Table 2.1: Midpoint estimates for the integral of x^2 on $[0, 1]$.

n	Estimate	Absolute error
1	0.250000	0.083333
2	0.312500	0.020833
4	0.328125	0.005208

Remark. These values check the implementation for one integrand. They do not establish an error bound for arbitrary continuous functions.

Appendix A

Notation

The interval endpoints are a and b. Split this interval into n equal parts with step size $h = (b - a)/n$.

Bibliography

[1] OpenStax. *Calculus Volume 2*. Rice University, 2016. Publisher's edition.