6.1 Introduction

If F is a *primitive function* of f [i.e., $F'(x) = f(x)$], one has for the definite integral:

$$I(f) = \int_a^b f(x)\mathrm{d}x = F(b) - F(a). \tag{6.1}$$

However, since not all functions f have analytic primitives we need numerical methods for integration (which is also called *numerical quadrature*). The most simple approach would be to calculate, for a rather small x step size (or, abscissa spacing), the classical upper sum (or lower sum) as introduced in the analytical concept of an integral. However, this would not be efficient since we would need to evaluate the integrand many times. There are better methods (better in the sense of being faster and more accurate), which mainly use simple polynomial interpolation of the integrand, taken at some sampling points, and deliver the integral in terms of the analytical result known for the interpolating polynomial.

For simplicity, here we consider only methods based on equally-spaced abscissas. In general, when the function to be integrated shows a strongly peaked behavior or has variations on different length scales, adaptive methods are needed which vary the x step size by themselves.[1] Such methods are common for numerically solving differential equations, and since every numerical integration is just a special case of solving a differential equation, namely,

$$\frac{\mathrm{d}y}{\mathrm{d}x} = f(x) \tag{6.2}$$

with boundary condition $y(a) = 0$ [then $y(b)$ will deliver the above integral $I(f)$], those methods can be used if not available otherwise.

[1]For example, the MATLAB routine `quadgk` uses an adaptive method.