Numerical Integration using the Trapezoidal Rule

Prof. Naga Kandasamy
ECE Department, Drexel University

Numerical integration is used to solve integrals that lack a closed-form solution. For example, one cannot evaluate the definite integral

$$f(x) = \int_0^1 \sqrt{\frac{1+x^2}{1+x^4}}\,\mathrm{d}x$$

symbolically. So, how does one integrate such functions?

A widely-used technique is numerical integration that approximates the value of the actual integral. Given a function $f(x)$ and the limit points a and b, where $a < b$, we wish to estimate the area under this curve; that is, we wish to determine $\int_a^b f(x)\ \mathrm{d}x$. The trapezoidal rule is one way to perform numerical integration.

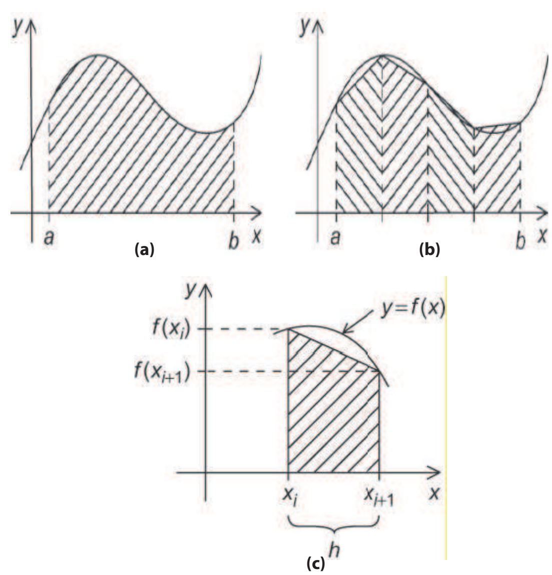

Figure 1: Illustration of the trapezoidal rule: (a) area to be estimated; (b) approximate area using trapezoids; and (c) area under one trapezoid.

The area between the graph of $f(x)$, the vertical lines $x = a$ and $x = b$, and the x-axis can be estimated as shown in Fig. 1 (b) by dividing the interval $[a, b]$ into n subintervals and approximating the area over each subinterval by the area of a trapezoid. Fig. 1(c) shows one such trapezoid where the base of the trapezoid is the subinterval, its vertical sides are the vertical lines through the endpoints of the subinterval, and the fourth side is the secant line joining the points where the vertical lines cross the graph. If the endpoints of the subinterval are x_i and x_{i+1}, then the length of the subinterval is $h = x_{i+1} - x_i$, and if the lengths of the two vertical segments are $f(x_i)$ and $f(x_{i+1})$, then the area of a single trapezoid is

$$\frac{h}{2}[f(x_i) + f(x_{i+1})].$$

If each subinterval has the same length then $h = (b - a)/n$. Also, if we call the leftmost endpoint x_0 and the rightmost endpoint x_n, we have

$$x_0 = a, x_1 = a + h, x_2 = a + 2h, \ldots, x_{n-1} = a + (n-1)h, x_n = b,$$

and our approximation of the total area under the curve will be

$$\int_a^b f(x)\,\mathrm{d}x = h[f(x_0)/2 + f(x_1) + f(x_2) + \cdots + f(x_{n-1}) + f(x_n)/2].$$

Thus, the pseudo-code for a serial algorithm might look something like the following.

```
procedure TRAP(a, b, n)
h := (b − a)/n;
sum := (f(a) + f(b))/2.0;
for i := 1 to n − 1 step 1 do
    x_i := a + i × h;
    sum := sum + f(x_i);
end for
sum := h × sum;
```