Introduction to Numerical Integration

James R. Nagel
Department of Electrical and Computer Engineering
University of Utah, Salt Lake City, Utah
February 4, 2012

1 Introduction

By definition, the integral of some function $f(x)$ between the limits a and b may be thought of as the area A between the curve and the x-axis. This configuration is shown in Figure 1, and is written mathematically as

$$A = \int_a^b f(x)\,dx \ . \tag{1}$$

By the fundamental theorem of calculus, we know that continuous functions have an exact analytical solution. Using $F(x)$ to denote the antiderivative of $f(x)$, this solution is given by

$$\int_a^b f(x)\,dx = F(b) - F(a) \ . \tag{2}$$

In practice, it is actually common for $F(x)$ to be extremely difficult (if not impossible) to obtain, thus forcing us to settle for a numerical approximation instead. This commonly occurs with applications that require definite integration of highly complex physical models or discrete samples of random data. Consequently, numerical integration has become an indispensable tool for processing sophisticated engineering designs. It is therefore important to gain an appreciation for the scope of numerical integration and its power to solve real engineering problems.

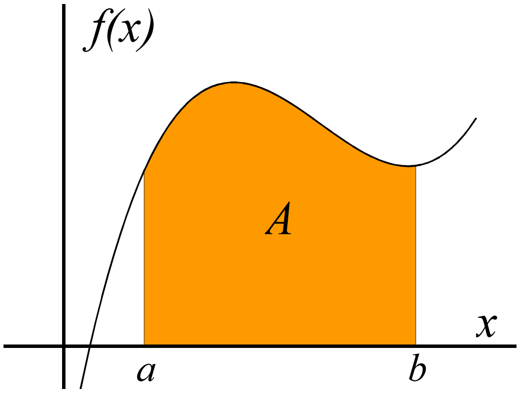


Figure 1: The integral of $f(x)$ from a to b represented as the area under the curve.

In general, there are three major trade-offs to consider when choosing a particular algorithm for definite integration. The first, and most important, is obviously the accuracy of a given numerical approximation. Fortunately, no integration scheme is so inaccurate that it cannot be compensated for by iterating the integration over a greater volume of data samples. Thus, the second metric to consider is the computational time required to achieve a given accuracy. This is usually measured in terms of function evaluations, or *fevals*, that the computer needs to perform while implementing the algorithm. Many advanced schemes also exist to provide very high degrees of accuracy with few fevals, but generally require far more time just to write the code than to simply grind on a more primitive version. The complexity of the algorithm is therefore the third major trade-off to consider. The "correct" choice of algorithm must then be determined by the nature of the application, the available computational resources, and the technical experience of the designer.

2 Riemann Sums

The most straightforward way to obtain a numerical approximation of a definite integral is through the use of a *Riemann sum*. In principle, this can be defined in several different ways, but one of the simplest methods is the *right-point* rule depicted in Figure 2(a). In this case, the area of the function is approximated by dividing the domain into n little subintervals and then adding up the combined areas of the resulting rectangles. The width Δx of each rectangle is then simply given by

$$\Delta x = \frac{b-a}{n} \; . \tag{3}$$

The height of each rectangle is found by evaluating the function at the points x_i along the domain between a and b, where x_i is defined as

$$x_i = a + i\Delta x \; . \tag{4}$$

The total area A is therefore approximated by

$$A \approx \sum_{i=1}^{n} f(x_i)\, \Delta x \; . \tag{5}$$

Another equally valid Riemann sum is the *left-point* rule depicted in Figure 2(b). In this case, the x_i points are evaluated using

$$x_i = a + (i-1)\Delta x \; , \tag{6}$$

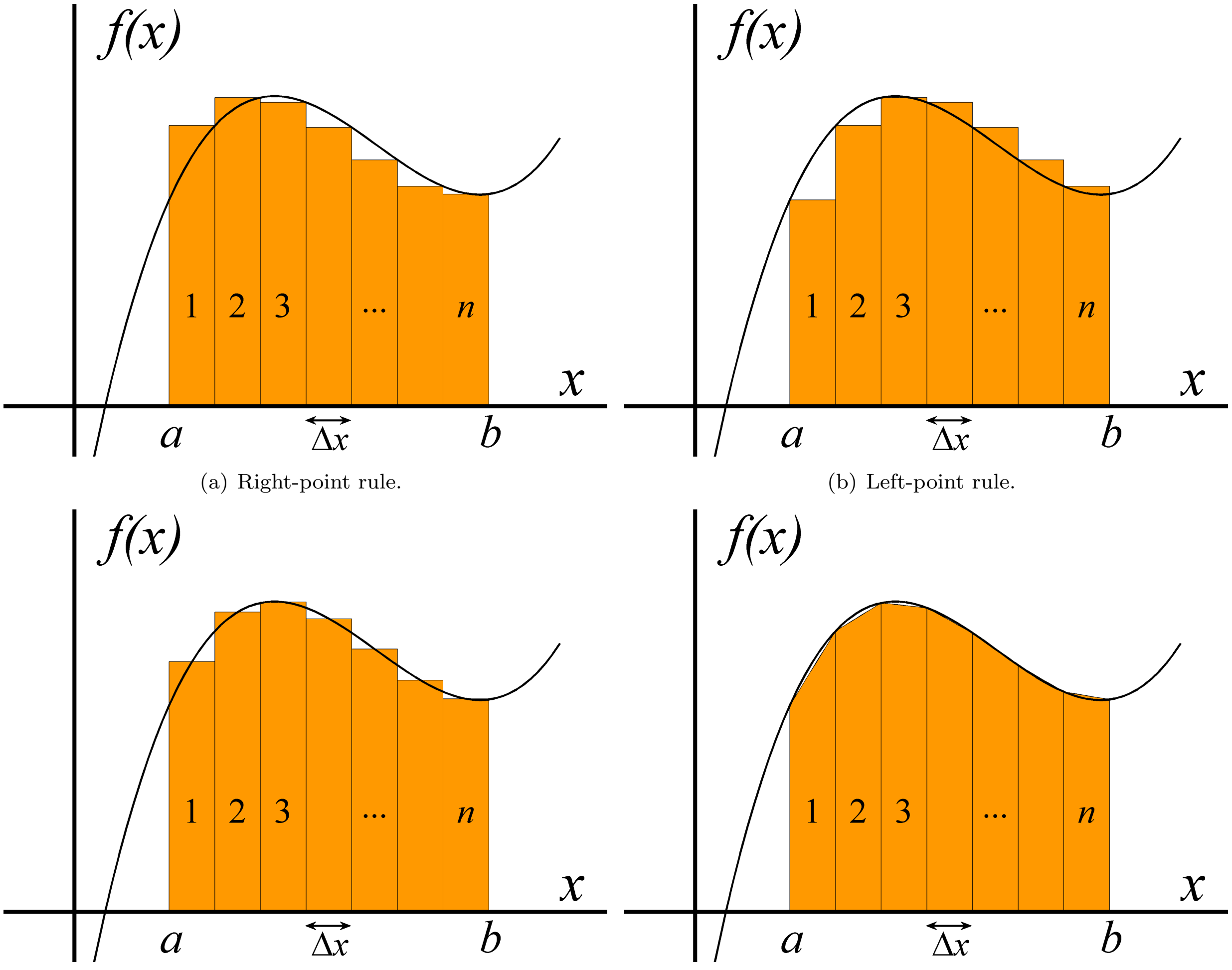
and A is again approximated by adding up the appropriate number of rectangles. In fact, there is a huge variety of techniques for approximating the area under a curve, and the only necessary condition is that a Riemann sum must approach the true value of A as $n \to \infty$:

$$\lim_{n\to\infty} \sum_{i=1}^{n} f(x_i)\, \Delta x = \int_a^b f(x)\, dx = A \; . \tag{7}$$

The importance of this condition is that it assures us any arbitrary degree of accuracy by simply adding up the appropriate number of rectangles.

3 Error Bounds

When considering a potential rule for numerical integration, it is helpful to establish some sort of metric to represent its effectiveness. This is usually accomplished by taking the difference between the value obtained



(a) Right-point rule.

(b) Left-point rule.

(c) Midpoint rule.

(d) Trapezoidal rule.

Figure 2: Various methods for calculating a Riemann sum.

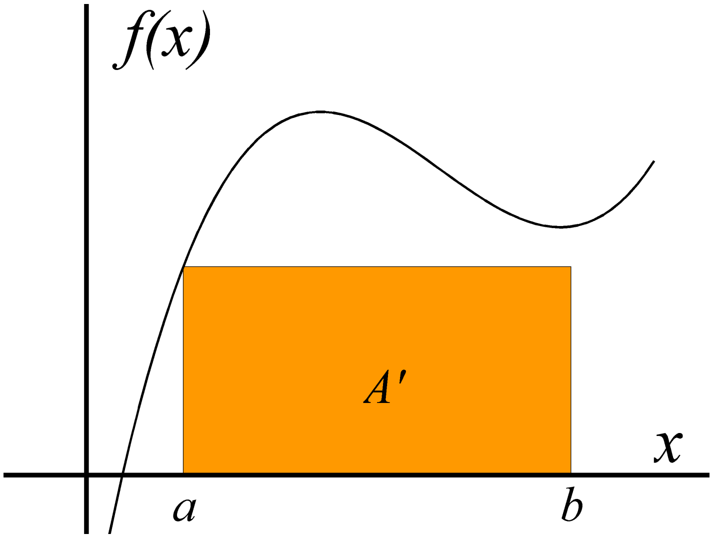


Figure 3: Left-point approximation using $n = 1$ rectangle.

from the numerical integration and the true, analytical result. Letting A' denote the numerical approximation to the true area A, the numerical error of integration is therefore given by

$$\epsilon = A' - A \ . \tag{8}$$

Naturally, it is not possible to evaluate ϵ without knowledge of A itself (and if we knew A, there would be no need to perform the numerical integral). We can, however, establish a sort of upper limit that determines the effectiveness of the numerical integration technique. To see how, remember that the exact area A is given from Equation (2) as

$$A = F(b) - F(a) \ . \tag{9}$$

If we now take the number $F(b)$ and express it as a Taylor series expansion around a, we find

$$F(b) = F(a) + F'(a)(b-a) + \frac{1}{2!}F''(a)(b-a)^2 + \frac{1}{3!}F^{(3)}(a)(b-a)^3 + \cdots \ . \tag{10}$$

Next, we note that the function $F'(a)$ is simply $f(a)$, and that $\Delta x = (b - a)$. Subtracting $F(a)$ therefore gives

$$A = F(b) - F(a) = f(a)\Delta x + \frac{1}{2!}f'(a)\Delta x^2 + \frac{1}{3!}f^{(2)}(a)\Delta x^3 + \cdots \ . \tag{11}$$

So as this expression shows, the area A under a curve can be represented as its own Taylor series expansion.

Now consider the left-point rule evaluated with a single ($n = 1$) rectangle as shown in Figure 3. Clearly, this scenario represents the absolute worst-case accuracy for approximating the area of the curve under f. The error for $n = 1$ rectangle therefore represents the *error bound* of the left-point rule, since the error can only improve as more rectangles are added. Thus, if we take $\Delta x = (b - a)$ and $x_i = a$, the approximate area is found to be

$$A' = f(a)\Delta x \ . \tag{12}$$

Comparison with Equation (11) shows that this is simply the first term in the Taylor series expansion for A. For this reason, the left-point rule is referred to as a *first order* approximation to the area.

Let us now compute the error term for a single rectangle. Using $\epsilon = A' - A$, it is easy to show that

$$\epsilon = -\left(\frac{1}{2!}f'(a)\Delta x^2 + \frac{1}{3!}f^{(2)}(a)\Delta x^3 + \cdots\right) \ . \tag{13}$$

Note that as the number of subintervals grows, the value of Δx naturally shrinks. This means that as $\Delta x \to 0$, the higher-order powers in Δx become negligible and the total error will be dominated by the first term. It is therefore common to drop all but the lowest order error term and approximate the net error as

$$\epsilon \approx -\frac{1}{2} f'(a) \Delta x^2 \ . \tag{14}$$

The importance of Equation (14) is that it tells us the relative severity of error we can expect by approximating the area of a curve with a single rectangle over a very small value of Δx. In particular, it shows how smaller values of Δx will have less error and how functions with a small derivative are better approximated. For this reason, constant-valued functions such as $f(x) = c$ are perfectly evaluated by the left-point rule because their derivatives are identically zero.

To calculate the error-bound of the right-point rule, we begin with the Taylor series expansion of $F(a)$ around the point b:

$$F(a) = F(b) - F'(b)(a-b) + \frac{1}{2!} F''(b)(a-b)^2 - \frac{1}{3!} F^{(3)}(b)(a-b)^3 + \cdots \ . \tag{15}$$

Solving for A in the same manner as before then leads to

$$A = f(b)\Delta x - \frac{1}{2!} f'(b) \Delta x^2 + \frac{1}{3!} f^{(2)}(b) \Delta x^3 - \cdots \ . \tag{16}$$

Again, we immediately recognize the first term in the series as the right-point rule, thus proving it is likewise a first-order approximation to the area. It is also important to note the alternating signs in the series expansion, which will be important later. The total error is then found to be

$$\epsilon = \frac{1}{2!} f'(b) \Delta x^2 - \frac{1}{3!} f^{(2)}(b) \Delta x^3 + \cdots \ , \tag{17}$$

with alternating signs out to infinity. For small values of Δx, this approximates into

$$\epsilon \approx \frac{1}{2} f'(b) \Delta x^2 \ , \tag{18}$$

which is the same the left-point rule, but with a positive sign in front. The significance of the sign is that functions with positive slope will give positive error (ie, overestimated) when using the right-point rule, and negative error (underestimated) when using the left-point rule.

4 Midpoint Rule

In practice, a far more useful integration technique is the *midpoint rule* shown in Figure 2(c). The rule works the same as before, but simply replaces the x_i terms with

$$x_i = a + \frac{i \Delta x}{2} \ . \tag{19}$$

The benefit to this method becomes apparent after computing the error bound. Consider again the $n = 1$ case shown in Figure 4. The approximate area A' is first divided into two regions around the point $c = \frac{b+a}{2}$ such that

$$A' = A_1 + A_2 \ . \tag{20}$$

The area defined by A_1 represents a right-point rule at the point c, while the area A_2 represents a left-point rule defined at the same point. Both of these give the same value of area, which is simply

$$A_1 = A_2 = f(c) \frac{\Delta x}{2} \ . \tag{21}$$

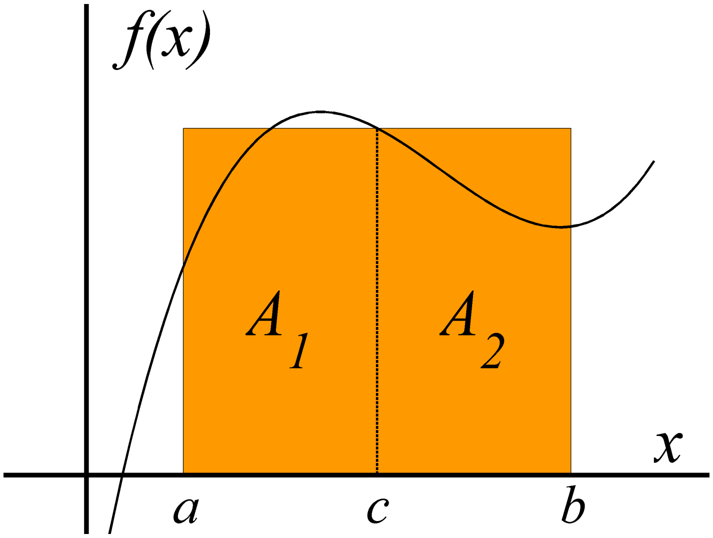


Figure 4: Midpoint approximation using $n = 1$ rectangle. The point c is defined by $\frac{b+a}{2}$.

The total error in A' is then found by simply adding the individual errors from A_1 and A_2. Adding Equations (13) and (17) with a width of $\Delta x/2$ therefore gives

$$\epsilon = -2\left(\frac{1}{3!}f^{(2)}(c)\frac{\Delta x^3}{8} + \frac{1}{5!}f^{(4)}(c)\frac{\Delta x^5}{32} + ...\right) . \tag{22}$$

For small values of Δx, this simplifies into

$$\epsilon \approx -\frac{1}{24}f^{(2)}(c)\Delta x^3 . \tag{23}$$

As this expression shows, the midpoint rule is a *second order* approximation for A because the error bound is dependent on the second derivative of f. This means linear functions such as $f(x) = mx + b$ are perfectly evaluated by the midpoint rule. As a result, the midpoint rule provides an extra order of accuracy simply by redefining the position of each x_i. Although this might seem odd at first, it should make perfect intuitive sense. As Figure 2(c) shows, the midpoint rule behaves as if the interval were actually subdivided into two rectangular regions rather than one. It is therefore natural that it should provide greater accuracy than either the left- or right-point rules alone.

5 Trapezoidal Rule

Another useful integration rule is the *trapezoidal rule* depicted in Figure 2(d). Under this rule, A' is evaluated by dividing the total area into little trapezoids rather than rectangles. We therefore begin by calculating the area of a single trapezoid like the one shown in Figure 5. Given a width Δx, a left-point height $f(a)$ and right-point height $f(b)$, the area of a trapezoid is simply the average area between the left-point rectangle and the right-point rectangle:

$$A' = \frac{\Delta x}{2}\Big[f(a) + f(b)\Big] . \tag{24}$$

From this expression, it is straightforward to show that for a series of trapezoids subdivided along $x_i = a+i\Delta x$ intervals, the approximate area of a function along the interval $[a, b]$ is

$$A' = \Delta x\left[\frac{f(a)}{2} + \frac{f(b)}{2} + \sum_{i=1}^{n-1} f(x_i)\right] . \tag{25}$$

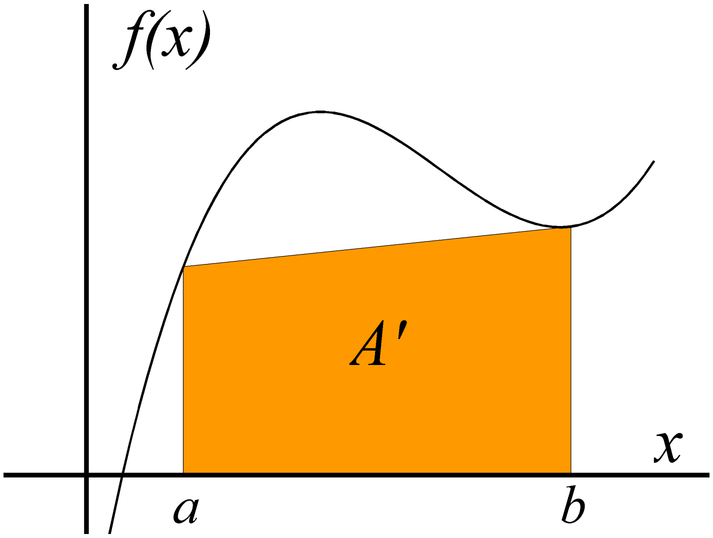


Figure 5: Trapezoidal approximation using $n = 1$ rectangle.

Now let us calculate the error bound for the trapezoidal rule. We begin by remembering the Taylor series expansions from Equation (11) and Equation (16). If these two expressions are averaged together, the total area can be expressed using

$$A = \frac{1}{2}f(a)\Delta x + \frac{1}{2}f(b)\Delta x + \frac{1}{4}f'(a)\Delta x^2 - \frac{1}{4}f'(b)\Delta x^2 + \frac{1}{12}f^{(2)}(a)\Delta x^3 + \frac{1}{12}f^{(2)}(b)\Delta x^3 + \cdots \ . \tag{26}$$

Upon inspection, we see that the first two terms in this expression represent the area A' of a single trapezoid found through Equation (24). Calculating the total error using $\epsilon = A' - A$ therefore leaves us with

$$\epsilon = -\frac{1}{4}f'(a)\Delta x^2 + \frac{1}{4}f'(b)\Delta x^2 - \frac{1}{12}f^{(2)}(a)\Delta x^3 - \frac{1}{12}f^{(2)}(b)\Delta x^3 - \cdots \ . \tag{27}$$

Although it may appear from this expression that the error bound of the trapezoidal rule is only first-order, it actually is second-order. To see how, consider what happens as Δx shrinks to a very small value. Using this assumption, it is reasonable to make the approximation

$$\Delta x f^{(2)}(c) \approx f'(b) - f'(a) \ , \tag{28}$$

which is simply the central-difference approximation to the derivative around the point $c = (b + a)/2$. Using similar logic we can also note that

$$f^{(2)}(c) \approx \frac{1}{2}\left[f^{(2)}(b) + f^{(2)}(a)\right] \ . \tag{29}$$

Substitution back into the original error expression therefore produces a second-order error bound:

$$\epsilon \approx \frac{1}{12}f^{(2)}(c)\Delta x^3 \ . \tag{30}$$

6 Simpson's Rule

A final integration scheme worth discussing briefly is *Simpson's rule.* In this case, the area under a curve is approximated through parabolic interpolation. The derivation of this rule is reasonably straightforward, though very tedious. We shall therefore skip to the end result by simply giving Simpson's rule as

$$A' = \frac{\Delta x}{3}\Big[f(a) + 4f(c) + f(b)\Big] \ , \tag{31}$$

where again $c = (b+a)/2$ is the midpoint between a and b. In this instance, we use $\Delta x = (c-a)$ to represent the subinterval between a and c. Because of this convention, it is useful to think of each approximate area A' as being comprised of two distinct subintervals. When carried out over a larger series of subintervals, Simpson's rule is expressed in summation form as

$$A' = \frac{\Delta x}{3}\left[f(a) + 2\sum_{n=1}^{n/2-1} f(x_{2n}) + 4\sum_{n=1}^{n/2} f(x_{2n-1}) + f(b)\right] \ , \tag{32}$$

where $x_i = a + \Delta x$. Note also that this expression requires an even number of subintervals in order to properly implement (or equivalently, an odd number of x_i terms).

Using similar methods as before, it is possible to show that the error bound for Simpson's rule satisfies

$$\epsilon \approx \frac{1}{180} f^{(4)}(c)\Delta x^5 \ . \tag{33}$$

Thus, Simpson's rule is a *fourth order* approximation to the true area under a curve. Of course, Simpson's rule is also significantly more complicated to implement than any of the previous methods. This reiterates the fundamental trade-off that is inherent throughout the entire field of numerical integration. Greater accuracy per function evaluation can only be achieved through greater complexity of the algorithm. Eventually, it is up to us to decide how much accuracy is truly needed for a given application and how much time we are willing spend to achieve it.

7 2D Integration

Integration in two dimensions works very much the same as in one dimension, only now it represents the volume V between some function $f(x, y)$ and a bounded region R in the x-y plane:

$$V = \iint_R f(x, y)dydx \ . \tag{34}$$

For introductory purposes, it helps to limit R to the simple rectangular area depicted in Figure 6. We shall therefore define R as the region bounded by $a \leq x \leq b$ and $c \leq y \leq d$, leaving the generalized case for an advanced course. The volume is then expressed in the double integral

$$V = \int_a^b \int_c^d f(x, y)\, dydx \ . \tag{35}$$

Just as there exists a wide variety of techniques for evaluating integrals in one dimension, there is an equally wide variety for two dimensions and beyond. We shall therefore limit the discussion to an introduction of the two simplest methods, which are the 2D midpoint rule and the 2D trapezoidal rule.

7.1 2D Midpoint Rule

The 2D midpoint rule begins by dividing the region R into a set of little rectangular subdomains, each with length Δx and width Δy. As illustrated in Figure 7, the x-dimension is divided into n subdomains while the y-dimension is divided into m, with the center point of each rectangle given by (x_i, y_j). The volume of

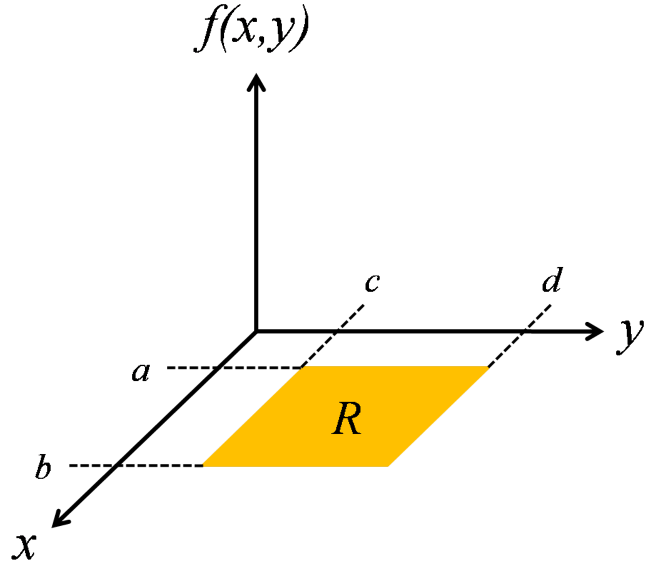


Figure 6: 2D region R defined by $a \leq x \leq b$ and $c \leq y \leq d$.

the box at this subdomain is then simply given by $f(x_i, y_j)\Delta x\Delta y$, and the total volume V is approximated by adding up all the little box elements:

$$V' = \sum_{i=1}^{n}\sum_{j=1}^{m} f(x_i, y_j)\Delta x\Delta y \ , \tag{36}$$

where

$$x_i = a + \frac{i\Delta x}{2} \quad \text{and} \quad y_j = c + \frac{j\Delta y}{2} \ . \tag{37}$$

7.2 2D Trapezoidal Rule

The division of R by the 2D trapezoidal rule is shown in Figure 8, where the feval points are given by

$$x_i = a + i\Delta x \quad \text{and} \quad y_j = c + j\Delta y \ , \tag{38}$$

with $i = 0, 1, \ ... \ , n$ and $j = 0, 1, \ ... \ , m$. In other words, the vertices represent the four corners of each subdomain. The volume of each 3D trapezoidal element is found by simply taking the average of the height at all four vertices and treating this as the equivalent "height" of a box. Each sub-volume may then be written as

$$V_{ij} = \frac{1}{4}\Delta x\Delta y\Big[f(x_{i-1}, y_{j-1}) + f(x_i, y_{j-1}) + f(x_{i-1}, y_j) + f(x_i, y_j)\Big] \ . \tag{39}$$

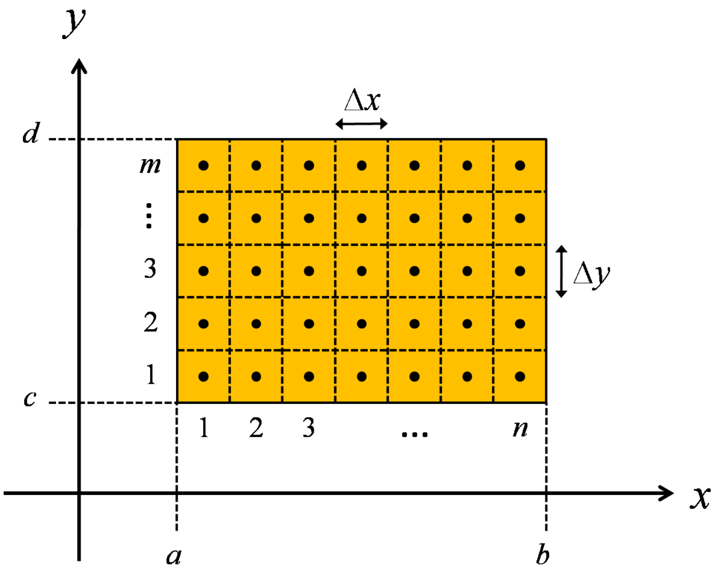


Figure 7: Subdomains for the 2D midpoint rule and their corresponding feval points.

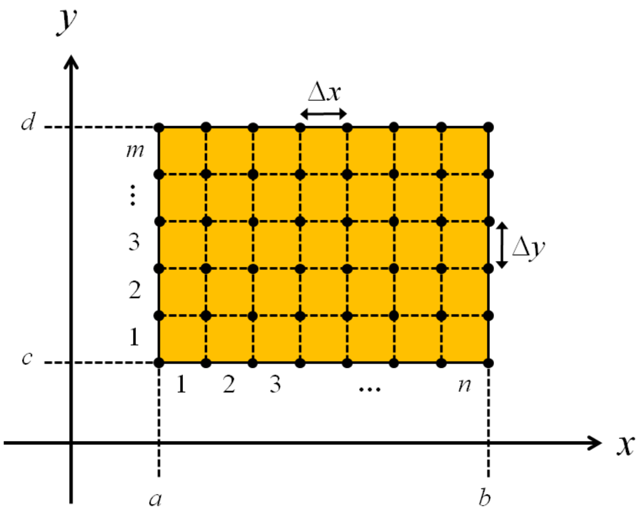


Figure 8: Subdomains for the 2D trapezoidal rule and their corresponding feval points.

Extending this single volume out to the entire region R therefore gives

$$\begin{aligned} V' = {} & \frac{\Delta x \Delta y}{4}\Big[f(a,c) + f(a,d) + f(b,c) + f(b,d)\Big] \\ & + \frac{1}{2}\Delta x \Delta y \left[\sum_{i=1}^{n-1} f(x_i, c) + \sum_{i=1}^{n-1} f(x_i, d) + \sum_{j=1}^{m-1} f(a, y_j) + \sum_{j=1}^{m-1} f(b, y_j)\right] \\ & + \Delta x \Delta y \sum_{i=1}^{n-1}\sum_{j=1}^{m-1} f(x_i, y_j) \ . \end{aligned} \tag{40}$$

Although this algorithm may appear complicated, it is really nothing more than an expression of how many averages each vertex contributes to. Namely, the four corners only contribute to one subdomain each, while the outer edges contribute to two subdomains, and the inner points all contribute to four subdomains. This expression also demonstrates the difference in complexity between various integration methods. From a purely computational perspective, both the 2D midpoint rule and trapezoidal rule require the same number of fevals to complete. Yet from an implementation perspective, the 2D trapezoidal rule requires much more thought and attention to physically program and debug the code. Which method one chooses to employ will therefore depend on the nature of the application.

8 Further Study

The field of numerical integration is very rich, and great efforts have been made to achieve the highest possible accuracy with each numerical computation. The trade-off with such rules is usually with the complexity of the algorithm, and eventually one must decide where it is worthwhile to simply perform more computations on a simpler algorithm rather than try to squeeze the most out of each iteration. Singularities also present an interesting problem when using Riemann sums, and a special form of numerical integration called *Gaussian quadrature* is the preferred method for dealing with such functions. Discontinuities are also a major concern and must be carefully considered for numerical methods as well. Although we have not touched on such topics in this paper, there is a fast wealth of literature available for students who wish to pursue this subject further.