Chapter 6

Integration of Functions

In this chapter we will

- introduce the concept of digitization
- learn how to evaluate definite integrals using the
  - Rectangle rule
  - Trapezoid rule
  - Newton-Cotes formulae
  - Gaussian quadratures
- identify Runge's phenomenon of interpolation
- devise techniques to handle improper integrals

In this chapter, we will study numerical methods that allow us to evaluate the definite integrals of real functions. Such integrals appear in many physical problems and are invariably related to the calculation of a path length, a surface area, or a volume with different weight functions. The numerical evaluation of a definite integral is often called numerical integration or **numerical quadrature**.

Because of their key place in calculations of physical quantities, numerical methods for the evaluation of definite integrals were developed in the very early days of calculus by scientists such as Newton, Euler, and Gauss. Even today, most of the methods used are based on these early works that build on the geometric interpretation of an integral as the area between a curve and the $x-$axis or the volume enclosed within a closed surface.

In order to study the different methods of numerical integration, however, we will need first to digress slightly and discuss the procedure that makes it possible to perform on a digital computer calculations that involve continuous functions of real numbers.

6.1 Discretization

In our discussion of numerical methods, so far, we focused on problems that have discrete solutions. For example, the frequencies of the normal modes of N coupled

Squaring the Circle

The term "quadrature" arises from the ancient problem of finding the quadrature of a circle, or squaring the circle, as is most commonly known. This problem was posed in various different forms from the Babylonian times, but in ancient Greece it referred to the construction of a square that has the same area as that of a circle, using only a compass and a straightedge. The impossibility of such a construction was proven in 1882, when it was shown that π is a transcendental number. However, as late as 1897, a bill was submitted to the House of Representatives of the state of Indiana, aiming to legislate the value of π to the rational number $16/5 = 3.2$, thus proving that a circle can be squared![1]

harmonic oscillators comprise a set of N real or imaginary numbers. In many physical problems, however, the solution is a continuous function. For example, in order to model the trajectory of a projectile fired from a cannon in the atmosphere, we need to describe the position of the particle as a function of time. In other cases, although the solution to the problem is a single real number, the procedure might involve the manipulation of continuous functions. In the previous example, if we want to calculate the total distance traveled by the projectile, we will need to integrate the continuous function that describes its trajectory.

Digital computers are not designed to handle continuous functions. As we have seen many times so far, even the representation of a single real number with floating point arithmetic involves several approximations and requires special care. The potential for pitfalls only becomes greater when dealing with continuous functions. The fundamental approach with which we manipulate a function $f(x)$ with a digital computer is by tabulating its value over a range of discrete values of the argument x. This is called **discretization**.

Consider, for example, a projectile that is shot from a cannot at an angle $\theta =$ 45 degrees with respect to the surface of the Earth and with an initial speed of $u_0 = 100$ m s^{-1}. If we assume that the gravitational field of the Earth is constant with an acceleration equal to $g = 10$ m s^{-2} and that there is no air resistance, then the horizontal and vertical displacements of the projectile as a function of time are given by the functions

$$x(t) = u_0 \cos(\theta) t = 50 \left(\frac{t}{1\ \mathrm{s}}\right)\ \mathrm{m} \tag{6.1}$$

and

$$z(t) = u_0 \sin(\theta) t - \frac{1}{2} g t^2 = \left[50\left(\frac{t}{1\ \mathrm{s}}\right) - 5\left(\frac{t}{1\ \mathrm{s}}\right)^2\right]\ \mathrm{m}\ , \tag{6.2}$$

respectively. The projectile will return back to the surface of the Earth after a time

$$t_0 = \frac{2u_0 \sin\theta}{g} = 10\ \mathrm{s} \tag{6.3}$$

and at a distance

$$x_{\max} = \frac{u_0^2 \sin(2\theta)}{g} = 500\ \mathrm{m} \tag{6.4}$$

from the cannon. In its trajectory, the projectile will reach a height of

$$z_{\max} = \frac{u_0^2 \sin^2(\theta)}{2g} = 125\ \mathrm{m}\ . \tag{6.5}$$

In order to perform any calculation with the functions $x(t)$ and $z(t)$, we will need to discretize them by tabulating their values over a discrete **grid** of times,

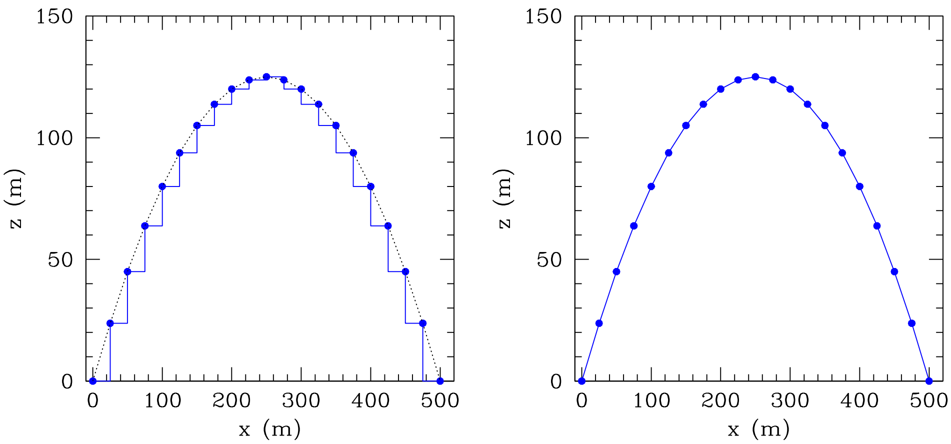

Figure 6.1: Two examples of discretizing the path of a projectile as a series of *(left)* steps or *(right)* line elements between the grid points.

e.g.,

i	t_i (s)	x_i (m)	z_i (m)
1	1	50	45
2	1.5	75	63.75
...	...	...	...
20	10	500	0

The index i counts the number of grid points.

It is important to emphasize here that by discretizing a continuous function, we loose all information about the behavior of the function between the grid points. As a result, the accuracy of performing calculations with discretized functions depends on the spacing between the neighboring grid points. In this example, we used a constant spacing of 0.5 s. This is not neccesary, as we could have used variable spacing as well. Our goal in discretizing a continuous function is to place enough grid points where the function changes rapidly and not as many where the function remains flat.

We can now use the discretized values of $x(t)$ and $z(t)$ in order to perform numerical calculations. We need to be very careful, however, and show explicitly that the calculation that we will perform with the discrete values gives the same answer as the calculation with the continuous functions when the number of discretization points goes to infinity. The following, rather obvious, example illustrates the potential problems.

Consider again the motion of the projectile discussed above and let us use the discretized values to calculate the total path length traveled by the projectile until it returned to the surface of the Earth. We will visualize the motion of the projectile in the $x-z$ plane, as shown in Figure 6.1.

One way to calculate the total path is by visualizing a staircase connecting the discrete pairs of points in the $x-z$ plane. This is how a computer monitor would pixelize the curve. The total path length would then be the sum of the displacements along the $x-$ and the $z-$ axes of each step. (Imagine calculating the path traveled by an ant that climbs up and down the staircase.) In algebraic form, this means that we are trying to calculate the total path S by the sum

$$S = \sum_{i=1}^{N-1} \left(|x_{i+1} - x_i| + |z_{i+1} - z_i| \right) . \tag{6.6}$$

In this expression, N is the total number of grid points. The sum, however, ends at $N-1$ because there are only $N-1$ "steps" in the "staircase".

When we consider the limit of an infinite number of grid points, the discretized "staircase" becomes visually indistinguishable from the continuous curve. The same is not true, however, for the sum given by expression (6.6). Indeed, projecting the individual steps on the $x-$ and the $z-$axis, we can see that the total sum is

$$S = x_{\max} + 2z_{\max} \tag{6.7}$$

independent of the number of grid points!

The correct way to calculate the total path traveled by the projectile is to connect the discrete pairs of points in the $x-z$ plane by straight line elements and sum their lengths. Algebraically, this means that we will calculate the total path using the expression

$$S = \sum_{i=1}^{N-1} \left[(x_{i+1}-x_i)^2 + (z_{i+1}-z_i)^2\right]^{1/2} \ . \tag{6.8}$$

In order to show that this expression gives the correct answer when we use an infinite number of grid points, we first rewrite it as

$$S = \sum_{i=1}^{N-1} \left[1 + \left(\frac{z_{i+1}-z_i}{x_{i+1}-x_i}\right)^2\right]^{1/2} (x_{i+1}-x_i) \ . \tag{6.9}$$

In this last expression we made explicit use of the fact that $x_{i+1} - x_i > 0$ in the problem we are studying.

We now set $h \equiv x_{i+i} - x_i$ and will use the definitions of the derivative of a function $z(x)$,

$$\frac{dz}{dx} = \lim_{h\to 0} \frac{z(x+h) - z(h)}{h} \ , \tag{6.10}$$

and of the definite integral of a function $f(x)$,

$$\int_a^b f(x)dx = \lim_{h\to\infty} \sum_i f(a+ih)h \ . \tag{6.11}$$

Taking the limit of an infinite number of grid points is equivalent to taking the limit of the grid separation becoming zero, $h \to 0$, and therefore

$$\lim_{N\to\infty} S = \lim_{h\to 0} S = \int_0^{x_{\max}} \left[1 + \left(\frac{dz}{dx}\right)^2\right]^{1/2} dx \ . \tag{6.12}$$

This last integral is nothing but the path length of the trajectory as calculated with the continuous functions $x(t)$ and $z(t)$. It is worth emphasizing here that an apparently small change between expressions (6.6) and (6.8) made the difference between a numerical method that works and one that does not!

6.2 Integration of Functions

The definite integral of a function $f(x)$

$$I = \int_a^b f(x)dx \tag{6.13}$$

is equal to the surface area of the shape enclosed by the curve that represents the function, the x-axis, and the vertical lines at $x = a$ and $x = b$.

In order to evaluate this integral, we first need to discretize the function over a series of N grid points $x_1, x_2, ..., x_i, ..., x_N$, such that $x_1 = a$ and $x_N = b$. If we choose the spacing h between two successive grid points to be constant, then it must be equal to

$$h = \frac{b-a}{N-1} \ , \tag{6.14}$$

where the denominator $N-1$ reflects the fact that there are $N-1$ intervals between N grid points. In this case, the $i-$th grid point can be found at an abscissa

$$x_i = a + (i-1)h \ . \tag{6.15}$$

With these definitions, the integral of the function over the entire interval becomes equal to the sum of the $N-1$ integrals between successive grid points, i.e.,

$$I = \sum_{i=1}^{N-1} \int_{x_i}^{x_{i+1}} f(x)dx \ . \tag{6.16}$$

Our task, therefore, reduces to devising a numerical method to calculate approximately the integral between two succesive grid points.

6.2.1 The Rectangle Rule

If we choose the grid points such as their separation is very small, then the function $f(x)$ is expected to change only marginally in the interval $[x_i, x_{i+1}]$. As a result, we may approximate it to be constant and equal to its value at some point within the interval. For example, we may assume the function to be equal to its value at the leftmost point of the interval, i.e., we may set $f(x) = f(x_i)$ so that the integral between two grid points becomes

$$\int_{x_i}^{x_{i+1}} f(x)dx = f(x_i)(x_{i+1} - x_1) \tag{6.17}$$

and the integral over the entire interval becomes

$$I = \int_a^b f(x)dx = \sum_{i=1}^{N-1} f(x_i)(x_{i+1} - x_1) \ . \tag{6.18}$$

Rectangle Rule

A C function that implements the rectangle rule for a constant separation between grid points is shown in the following page. The algorithm requires three arguments, the lower and upper limit of integration as well as the number of grid points used. Upon completion, it returns the numerical value of the definite integral. This algorithm assumes that the integrand $f(x)$ is provided by the user as a C function of the form

```
double foo(double x);
```

Programming Tip

Note that, in this algorithm, we have used the integer counter `icounter` to step through the various grid points and evaluate the sum of the rectangle rule. Mathematically speaking, the `for` statement that we used in the algorithm is equivalent to the statement

```
for(xgrid=xlower;xgrid<xupper;xgrid+=step)
```

This last statement appears to be more efficient than the one we used because it uses one less variable and removes the need for the statement

```
xgrid+=step; // Go to next grid point
```

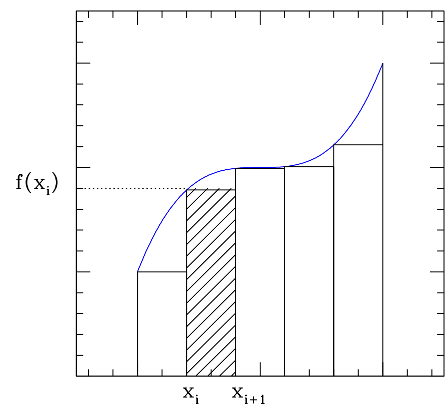

Figure 6.2: In the rectangle rule, the area between the curve $f(x)$ and the $x-$axis is approximated by a sum of rectangles between successive grid points.

within the body of the loop. However, it does not guarantee that the loop will terminate before the variable `xgrid` becomes approximately equal to `xupper`, because of rounding errors, as we discussed in Chapter 2.

Graphical Representation

Figure 6.2 shows that the approximation for the integral we devised is equivalent to replacing the surface area between the function $f(x)$ and the $x-$axis with a series of rectangles between successive grid points. Note here that we chose to approximate the function by its value at the leftmost point in each interval. We could have equally well used the rightmost or the middle point of each interval, without changing the level of approximation of the method.

It is important to discuss the fact that the rectangle method approximates the curve $f(x)$ as a "staircase" in a manner that is practically indistinguishable from the discretization shown in the left pannel of figure 6.1. As we discussed in the previous section, this discretization failed to provide us with an accurate measurement of the length of the curve, independent of the number of grid points used. As we will show below, however, this same discretization method allows us to evaluate the area under the curve with an error that diminishes as the number of grid points goes to infinity.

Error of Approximation

Our aim here is to evaluate the error of approximating the integral of a function with the rectangle rule, that is calculate the quantity

$$\epsilon \equiv \int_a^b f(x)dx - \sum_{i=1}^{N-1} f(x_i)(x_{i+1} - x_i) \ . \tag{6.19}$$

We begin by Taylor expanding the function around x_i,

$$f(x) = f(x_i) + f'(\xi)(x - x_i) \tag{6.20}$$

where $x_i \leq \xi \leq x_{i+1}$. We then integrate both sides of this equation from x_i to x_{i+1},

$$\int_{x_i}^{x_{i+1}} f(x)dx = f(x_i)(x_{i+1} - x_i) + \frac{1}{2}(x_{i+1} - x_i)^2 f'(\xi) \ . \tag{6.21}$$

The first term in the above sum is equal to the approximate value of the integral using the rectangle method. The second term in the sum is, therefore, the error in the approximation. Assuming for simplicity that the spacing between any two

```
double foo(double x);           // prototype for integrand

double rectangle(double xlower, double xupper, int Ngrids)
/* Uses the rectangle rule to calculate the definite
   integral of a function in the interval [xlower,xupper].
   The rectangle rule is applied on a grid of Ngrids
   values of the abscissa with a constant separation. The
   form of the integrand is provided by the function
   foo(x) and is supplied by the user.
 */
{
  double integral=0.0;          // the value of the integral
  int icounter;                 // Counter for grid points
  double xgrid;                 // Abscissa of grid point
  double step;                  // Separation between grid points

  step=(xupper-xlower)/(Ngrids-1);
  xgrid=xlower;                 // First grid point
                                // for all but last grid point
  for (icounter=1;icounter<Ngrids;icounter++)
    {
      integral+=foo(xgrid)*step;// Add the rectangle area
      xgrid+=step;              // Go to next grid point
    }

  return integral;
}
```

successive grid points is constant and equal to h, we can write the error introduced in calculating the integral between these two points as

$$\epsilon_i = \frac{1}{2} f'(\xi) h^2 \ . \tag{6.22}$$

We now need to take into account the fact that the rectangle method involves taking the sum of $N-1$ such intervals, and therefore the error of approximation can be as large as

$$\epsilon \simeq (N-1)\frac{1}{2}\langle f'(\xi)\rangle h^2 \simeq \frac{b-a}{2}\langle f'(\xi)\rangle h \ . \tag{6.23}$$

In this last expression, we used the fact that $N-1=(b-a)/h$ (see equation (6.14) and the symbol $\langle f'(\xi)\rangle$ to denote an appropriate average of the first derivative of the function $f(x)$ in the entire interval.

This last expression shows that decreasing the spacing between successive grid points by a factor of two reduces the error of the rectangle method by the same amount. The approximation, therefore, converges **linearly** to the true value of the integral, as the number of grid points becomes infinite.

6.2.2 The Trapezoid Rule

We can obtain a higher level of approximation in the evaluation of the integral by assuming that the function $f(x)$ varies linearly between any two successive grid points x_i and x_{i+1}. This is equivallent to assuming that, for $x_i \le x \le x_{i+1}$,

$$f(x) \simeq f(x_i) + \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}\,(x - x_i) \ . \tag{6.24}$$

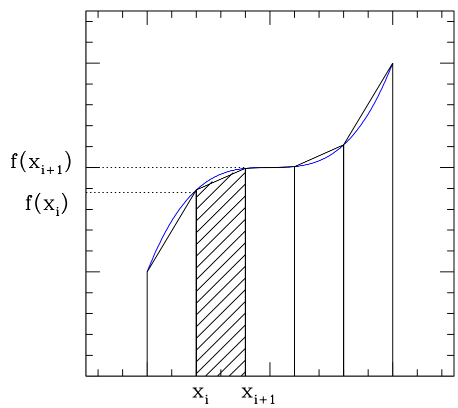

Figure 6.3: In the trapezoid rule, the area between the curve $f(x)$ and the $x-$axis is approximated by a sum of trapezoids between successive grid points.

A quick inspection of this last relation, which is linear in x, shows that it indeed gives the correct values of the function $f(x)$ at the two limits of the interval, x_i and x_{i+1}.

Integrating the above approximate relation between the two successive grid points, we obtain

$$\int_{x_i}^{x_{i+1}} f(x)dx \simeq \frac{1}{2}\left[f(x_{i+1})+f(x_i)\right](x_{i+1}-x_i) \ . \tag{6.25}$$

As a result, our approximation for the integral over the entire interval becomes

Trapezoid Rule

$$I=\int_a^b f(x)dx=\sum_{i=1}^{N-1}\frac{1}{2}[f(x_i)+f(x_{i+1})](x_{i+1}-x_i)\ . \tag{6.26}$$

A C function that implements the trapezoid rule is shown in the following page. As in the case of the algorithm for the rectangle rule, it requires three arguments, the upper and lower limit of integration as well as the number of grid points used. Note that we did not use the summation (6.26) in the implementation of our algorithm, because, albeit transparent, it is not optimal in evaluating the trapezoid rule. Indeed, the value of the function at each interior grid point x_i is calculated twice during the summation, once for the $(i-1)$-th term and once for the i-th term. We can avoid duplicating this effort by regrouping the terms in the summation as

$$\begin{aligned}
I &= \sum_{i=1}^{N-1}\frac{1}{2}[f(x_i)+f(x_{i+1})](x_{i+1}-x_i)+\\
&= \frac{1}{2}f(x_1)(x_2-x_1)+\frac{1}{2}f(x_2)(x_2-x_1)+\\
&\quad \frac{1}{2}f(x_2)(x_3-x_2)+\frac{1}{2}f(x_3)(x_3-x_2)+...+\\
&\quad \frac{1}{2}f(x_{N-1})(x_N-x_{N-1})+\frac{1}{2}f(x_N)(x_N-x_{N-1})\\
&= \frac{1}{2}f(x_1)(x_2-x_1)+\frac{1}{2}\sum_{i=2}^{N-1}f(x_i)(x_{i+1}-x_{i-1})+\\
&\quad \frac{1}{2}f(x_N)(x_N-x_{N-1})\ .
\end{aligned} \tag{6.27}$$

```
double foo(double x);          // prototype for integrand

double trapezoid(double xlower, double xupper, int Ngrids)
/* Uses the trapezoid rule to calculate the definite
   integral of a function in the interval [xlower,xupper].
   The trapezoid rule is applied on a grid of Ngrids
   values of the abscissa with a constant separation. The
   form of the integrand is provided by the function
   foo(x) and is supplied by the user.
 */
{
  double integral=0.0;              // the value of the integral
  int icounter;                     // Counter for grid points
  double xgrid;                     // Abscissa of grid point
  double step;                      // Separation between grid points

  step=(xupper-xlower)/(Ngrids-1);
  xgrid=xlower;                     // First grid point
  integral=foo(xgrid)*0.5*step;
  for (icounter=1;icounter<Ngrids-1;icounter++)
    {
      integral+=foo(xgrid)*step;   // Apply trapezoid rule
      xgrid+=step;                 // Go to next grid point
    }
  integral+=foo(xlower)*0.5*step;  // Last grid point

  return integral;
}
```

When the separation between two consecutive grid points is constant and equal to h, then the trapezoid rule becomes

$$I = \frac{1}{2}f(x_1)h + \sum_{i=2}^{N-1} f(x_i)h + \frac{1}{2}f(x_N)h\ . \tag{6.28}$$

Note that when $f(x_1) = -f(x_N)$ and the distance between two successive grid points is constant, the trapezoid and rectangle rules become identical!

Graphical Representation

Figure 6.3 shows that the approximation to the total integral given by equation (6.26) is equivalent to replacing the surface area between the function $f(x)$ and the $x-$axis with a series of trapezoids between successive grid points. Comparing this to Figure 6.2, it appears, at least graphically, that the trapezoid rule provides a better approximation to the continuous integral compared to the rectangle rule. We will indeed show below that the trapezoid rule converges at most **quadratically** to the true value of the integral.

Error of Approximation

Our aim is to evaluate the quantity

$$\epsilon \equiv \int_a^b f(x)dx - \sum_{i=1}^{N-1} \frac{1}{2}[f(x_i) + f(x_{i+1})](x_{i+1} - x_i)\ , \tag{6.29}$$

which measures the error between the true value of the integral and the one we calculate with the trapezoid rule. We begin again by Taylor expanding the function $f(x)$ around x_i, but we keep now terms up to second order in $(x_{i+1} - x_i)$,

$$f(x) = f(x_i) + f'(x_i)(x - x_i) + \frac{1}{2}f''(\xi)(x - x_i)^2\ , \tag{6.30}$$

where $x_i \le \xi \le x_{i+1}$. We then integrate both sides of this equation from x_i to x_{i+1} to obtain

$$\int_{x_i}^{x_{i+1}} f(x)dx = f(x_i)(x_{i+1}-x_i)+\frac{1}{2}(x_{i+1}-x_i)^2 f'(x_i)+\frac{1}{3!}(x_{i+1}-x_i)^3 f''(\xi)\ . \quad (6.31)$$

We continue by Taylor expanding again the function $f(x)$ but this time around x_{i+1} as

$$f(x) = f(x_{i+1}) + f'(x_{i+1})(x - x_{i+1}) + \frac{1}{2}f''(\xi')(x - x_{i+1})^2\ , \quad (6.32)$$

where again $x_i \le \xi' \le x_{i+1}$. We then integrate both sides of this equation from x_i to x_{i+1} to obtain

$$\int_{x_i}^{x_{i+1}} f(x)dx = f(x_{i+1})(x_{i+1} - x_i) - \frac{1}{2}(x_{i+1} - x_i)^2 f'(x_{i+1}) + \frac{1}{3!}(x_{i+1} - x_i)^3 f''(\xi')\ . \quad (6.33)$$

If we now average equations (6.31) and (6.33), we obtain

$$\begin{aligned}\int_{x_i}^{x_{i+1}} f(x)dx \quad = \quad & \frac{1}{2}[f(x_i) + f(x_{i+1})](x_{i+1} - x_i) - \\ & \frac{1}{4}(x_{i+1} - x_i)^2 \left[f'(x_{i+1}) - f'(x_i)\right] + \\ & \frac{1}{12}(x_{i+1} - x_i)^3 \left[f''(\xi) + f''(\xi')\right]\ . \end{aligned} \quad (6.34)$$

The second term in the sum appearing in the right-hand side of the last equation is actually a term of third order in the small difference $(x_{i+1} - x_i)$. In order to see this, we can differentiate both sides of equation (6.30) and evaluate it at $x = x_{i+1}$, i.e.,

$$f'(x_{i+1}) = f'(x_i) + f''(\xi)(x_{i+1} - x_i)\ . \quad (6.35)$$

We the insert this expression into equation (6.34) to obtain

$$\begin{aligned}\int_{x_i}^{x_{i+1}} f(x)dx \quad = \quad & \frac{1}{2}[f(x_i) + f(x_{i+1})](x_{i+1} - x_i) - \\ & \frac{1}{12}(x_{i+1} - x_i)^3 \left[4f''(\xi) + f''(\xi')\right]\ . \end{aligned} \quad (6.36)$$

Assuming for simplicity that the spacing between any two successive grid points is constant and equal to h, we can write the error introduced in calculating the integral between these two points as

$$\epsilon_i = \frac{1}{12}\left[4f''(\xi) + f''\right] h^3\ . \quad (6.37)$$

Taking into account the fact that the trapezoid method involves taking the sum of $N - 1$ such intervals, the error of approximation can be as large as

$$\epsilon \simeq (N - 1)\frac{5}{12}\langle f''(\xi)\rangle h^3 \simeq \frac{5(b - a)}{12}\langle f''(\xi)\rangle h^2\ . \quad (6.38)$$

In this last expression, we used the fact that $N - 1 = (b - a)/h$ (see equation (6.14) and the symbol $\langle f''(\xi)\rangle$ to denote an appropriate average of the second derivative of the function $f(x)$ in the entire interval. Decreasing the spacing between successive grid points by a factor of two reduces the error of the trapezoid rule by a factor of four. The approximation, therefore, converges quadratically to the true value of the integral, as the number of grid points becomes infinite.