LECTURE 22: *Newton–Cotes quadrature*

3.2 *Newton–Cotes quadrature*

You encountered the most basic method for approximating an integral when you learned calculus: the Riemann integral is motivated by approximating the area under a curve by the area of rectangles that touch that curve, which gives a rough estimate that becomes increasingly accurate as the width of those rectangles shrinks. This amounts to approximating the function f by a piecewise constant interpolant, and then computing the exact integral of the interpolant. When only one rectangle is used to approximate the entire integral, we have the most simple *Newton–Cotes* formula; see Figure 3.1.

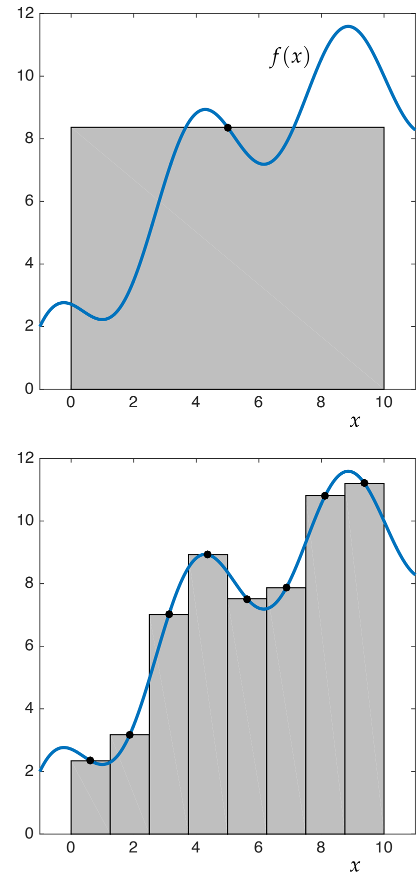

Figure 3.1: Estimates of $\int_0^{10} f(x)\,\mathrm{d}x$, shown in gray: the first approximates f by a constant interpolant; the second, a *composite rule*, uses a piecewise constant interpolant. You probably have encountered this second approximation as a *Riemann sum*.

Newton–Cotes formulas are interpolatory quadrature rules where the quadrature notes $x_0, \ldots, x_n$ are uniformly spaced over $[a, b]$,

$$x_j = j\left(\frac{b-a}{n}\right).$$

Given the lessons we learned about polynomial interpolation at uniformly spaced points in Section 1.6, you should rightly be suspicious of applying this idea with large n (i.e., high degree interpolants). A more reliable way to increase accuracy follows the lead of basic Riemann sums: partition $[a, b]$ into smaller subintervals, and use low-degree interpolants to approximate the integral on each of these smaller domains. Such methods are called *composite* quadrature rules.

In some cases, the function f may be fairly regular over most of the domain $[a, b]$, but then have some small region of rapid growth or oscillation. Modern *adaptive* quadrature rules are composite rules on which the subintervals of $[a, b]$ vary in size, depending on estimates of how rapidly f is changing in a given part of the domain. Such methods seek to balance the competing goals of highly accurate approximate integrals and as few evaluations of f as possible.We shall not dwell much on these sophisticated quadrature procedures here, but rather start by understanding some methods you were probably introduced to in your first calculus class.

3.2.1 *The trapezoid rule*

The trapezoid rule is a simple improvement over approximating the integral by the area of a single rectangle. A linear interpolant to f can be constructed, requiring evaluation of f at the interval end points $x_0 = a$ and $x_1 = b$. Using the interpolatory quadrature methodology described in the last section, we write

$$p_1(x) = f(a)\left(\frac{x-b}{a-b}\right) + f(b)\left(\frac{x-a}{b-a}\right),$$

and compute its integral as

$$\begin{aligned}\int_a^b p_1(x)\,dx &= \int_a^b f(a)\left(\frac{x-b}{a-b}\right) + f(b)\left(\frac{x-a}{b-a}\right)dx \\ &= f(a)\int_a^b \frac{x-x_1}{x_0-x_1}\,dx + f(b)\int_a^b \frac{x-x_1}{x_0-x_1}\,dx \\ &= f(a)\left(\frac{b-a}{2}\right) + f(b)\left(\frac{b-a}{2}\right).\end{aligned}$$

In summary,

Trapezoid rule:

$$\int_a^b f(x)\,dx \approx \frac{b-a}{2}\Big(f(a)+f(b)\Big).$$

The procedure behind the trapezoid rule is illustrated in Figure 3.2 where the area approximating the integral is colored gray.

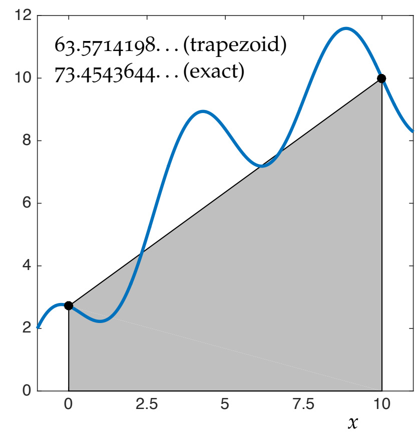

Figure 3.2: Trapezoid rule estimate of $\int_0^{10} f(x)\,dx$, shown in gray.

To derive an error bound for the trapezoid rule, simply integrate the fundamental interpolation error formula in Theorem 1.3. That gave, for each $x \in [a,b]$, some $\xi \in [a,b]$ such that

$$f(x) - p_1(x) = \tfrac{1}{2}f''(\xi)(x-a)(x-b).$$

Note that ξ will vary with x, which we emphasize by writing $\xi(x)$. Integrate this formula to obtain

$$\begin{aligned}\int_a^b f(x)\,dx - \int_a^b p_1(x)\,dx &= \int_a^b \frac{1}{2}f''(\xi(x))(x-a)(x-b)\,dx \\ &= \frac{1}{2}f''(\eta)\int_a^b (x-a)(x-b)\,dx \\ &= \frac{1}{2}f''(\eta)\left(\frac{1}{6}a^3 - \frac{1}{2}a^2b + \frac{1}{2}ab^2 - \frac{1}{6}b^3\right) \\ &= -\frac{1}{12}f''(\eta)(b-a)^3\end{aligned}$$

for some $\eta \in [a,b]$. The second step follows from the mean value theorem for integrals.

The mean value theorem for integrals states that if $h, g \in C[a,b]$ and *h does not change sign on $[a,b]$*, then there exists some $\eta \in [a,b]$ such that $\int_a^b g(t)h(t)\,dt = g(\eta)\int_a^b h(t)\,dt$. The requirement that h not change sign is essential. For example, if $g(t) = h(t) = t$ then $\int_{-1}^1 g(t)h(t)\,dt = \int_{-1}^1 t^2\,dt = 2/3$, yet $\int_{-1}^1 h(t)\,dt = \int_{-1}^1 t\,dt = 0$, so for all $\eta \in [-1,1]$, $g(\eta)\int_{-1}^1 h(t)\,dt = 0 \neq \int_{-1}^1 g(t)h(t)\,dt = 2/3$.

In a forthcoming lecture we shall develop a much more general theory, based on the *Peano kernel*, from which we can derive this error bound, plus bounds for more complicated schemes, too. For now, we summarize the bound in the following Theorem.

Theorem 3.2. Let $f \in C^2[a,b]$. The error in the trapezoid rule is

$$\int_a^b f(x)\,\mathrm{d}x - \left(\frac{b-a}{2}\Big(f(a)+f(b)\Big)\right) = -\frac{1}{12}f''(\eta)(b-a)^3$$

for some $\eta \in [a,b]$.

This bound has an interesting feature: if we are integrating over the small interval, $b-a = h \ll 1$, then the error in the trapezoid rule approximation is $\mathcal{O}(h^3)$ as $h \to 0$, while the error in the linear interpolant upon which this quadrature rule is based is only $\mathcal{O}(h^2)$ (from Theorem 1.3).

Example 3.1 ($f(x) = \mathrm{e}^x(\cos x + \sin x)$). Here we demonstrate the difference between the error for linear interpolation of a function, $f(x) = \mathrm{e}^x(\cos x + \sin x)$, between two points, $x_0 = 0$ and $x_1 = h$, and the trapezoid rule applied to the same interval. The theory reveals that linear interpolation will have an $\mathcal{O}(h^2)$ error as $h \to 0$, while the trapezoid rule has $\mathcal{O}(h^3)$ error, as confirmed in Figure 3.3.

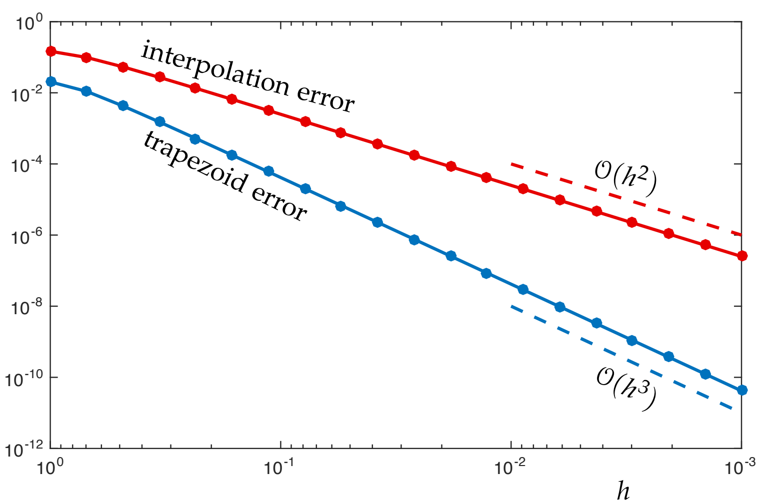

Figure 3.3: Error of linear interpolation and trapezoid rule approximation for $f(x) = \mathrm{e}^x(\cos x + \sin x)$ for $x \in [0,h]$ as $h \to 0$.

3.2.2 *Simpson's rule*

To improve the accuracy of the trapezoid rule, increment the degree of the interpolating polynomial. This will increase the number of evaluations of f (often very costly), but hopefully will significantly decrease the error. Indeed it does – by an even greater margin than we might expect.

Simpson's rule integrates the quadratic interpolant $p_2 \in \mathcal{P}_2$ to f at the uniformly spaced points

$$x_0 = a, \qquad x_1 = (a+b)/2, \qquad x_2 = b.$$

Using the interpolatory quadrature formulation of the last section,

$$\int_a^b p_2(x)\,\mathrm{d}x = w_0 f(a) + w_1 f(\tfrac{1}{2}(a+b)) + w_2 f(c),$$

where

$$w_0 = \int_a^b \left(\frac{x-x_1}{x_0-x_1}\right)\left(\frac{x-x_2}{x_0-x_2}\right)\mathrm{d}x = \frac{b-a}{6}$$

$$w_1 = \int_a^b \left(\frac{x-x_0}{x_1-x_0}\right)\left(\frac{x-x_2}{x_1-x_2}\right)\mathrm{d}x = \frac{2(b-a)}{3}$$

$$w_2 = \int_a^b \left(\frac{x-x_0}{x_2-x_0}\right)\left(\frac{x-x_1}{x_2-x_1}\right)\mathrm{d}x = \frac{b-a}{6}.$$

In summary:

Simpson's rule:

$$\int_a^b f(x)\,\mathrm{d}x \approx \frac{b-a}{6}\Big(f(a) + 4f(\tfrac{1}{2}(a+b)) + f(b)\Big).$$

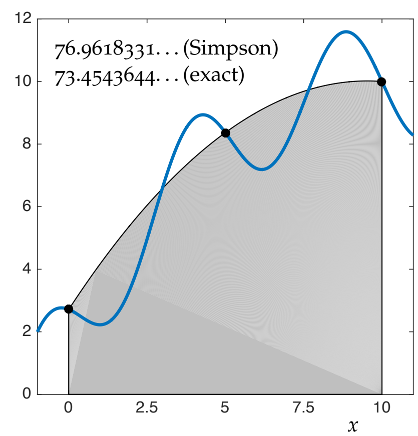

Figure 3.4: Simpson's rule estimate of $\int_0^{10} f(x)\,\mathrm{d}x$, shown in gray.

Simpson's rule enjoys a remarkable feature: though it only approximates f by a quadratic, *it integrates any cubic polynomial exactly*! One can verify this by directly applying Simpson's rule to a generic cubic polynomial. Write $f(x) = \alpha x^3 + q(x)$, where $q \in \mathcal{P}_2$. Let $I(f) = \int_a^b f(x)\,\mathrm{d}x$ and let $I_2(f)$ denote the Simpson's rule approximation. Then, by linearity of the integral,

$$I(f) = \alpha I(x^3) + I(q)$$

and, by linearity of Simpson's rule,

$$I_2(f) = \alpha I_2(x^3) + I_2(q).$$

Since Simpson's rule is an interpolatory quadrature rule based on quadratic polynomials, its degree of exactness must be at least 2 (Theorem 3.1), i.e., it exactly integrates q: $I_2(q) = I(q)$. Thus

$$I(f) - I_2(f) = \alpha\Big(I(x^3) - I_2(x^3)\Big).$$

So Simpson's rule will be exact for all cubics if it is exact for x^3. A simple computation gives

$$\begin{aligned} I_2(x^3) &= \frac{b-a}{6}\left(a^3 + 4\Big(\frac{a+b}{2}\Big)^3 + b^3\right) \\ &= \frac{b-a}{12}\Big(3a^3 + 3a^2b + 3ab^2 + 3b^3\Big) = \frac{b^4 - a^4}{4} = I(x^3), \end{aligned}$$

confirming that Simpson's rule is exact for x^3, and hence for all cubics. For now we simply state an error bound for Simpson's rule, which we will prove in a future lecture.

In fact, Newton–Cotes formulas based on approximating f by an even-degree polynomial always exactly integrate polynomials one degree higher.

Theorem 3.3. Let $f \in C^4[a,b]$. The error in the Simpson's rule is

$$\int_a^b f(x)\,\mathrm{d}x - \left(\frac{b-a}{6}\Big(f(a) + 4f((a+b)/2) + f(b)\Big)\right) = -\frac{1}{90}f^{(4)}(\eta)(b-a)^5$$

for some $\eta \in [a,b]$.

This error formula captures the fact that Simpson's rule is exact for cubics, since it features the fourth derivative $f^{(4)}(\eta)$, two derivatives greater than $f''(\eta)$ in the trapezoid rule bound, even though the degree of the interpolant has only increased by one. Perhaps it is helpful to visualize the exactness of Simpson's rule for cubics. Figure 3.5 shows $f(x) = x^3$ (blue) and its quadratic interpolant (red). On the left, the area under f is colored gray: its area is the integral we seek. On the right, the area under the interpolant is colored gray. Accounting area below the x axis as negative, both integrals give an identical value even though the functions are quite different. It is remarkable that this is the case for *all* cubics.

Typically one does not see Newton–Cotes rules based on polynomials of degree higher than two (i.e., Simpson's rule). Because it can be fun to see numerical mayhem, we give an example to emphasize why high-degree Newton–Cotes rules can be a bad idea. Recall that Runge's function $f(x) = 1/(1+x^2)$ gave a nice example for which the polynomial interpolant at uniformly spaced points over $[-5,5]$ fails to converge uniformly to f. This fact suggests that Newton–Cotes quadrature will also fail to converge as the degree of the interpolant grows. The exact value of the integral we seek is

Integrating the cubic interpolant at four uniformly spaced points is called *Simpson's three-eighths rule*.

$$\int_{-5}^{5} \frac{1}{1+x^2}\,\mathrm{d}x = 2\tan^{-1}(5) = 2.75680153\ldots.$$

Just as the interpolant at uniformly spaced points diverges, so too does the Newton–Cotes integral. Figure 3.6 illustrates this divergence, and shows that integrating the interpolant at Chebyshev

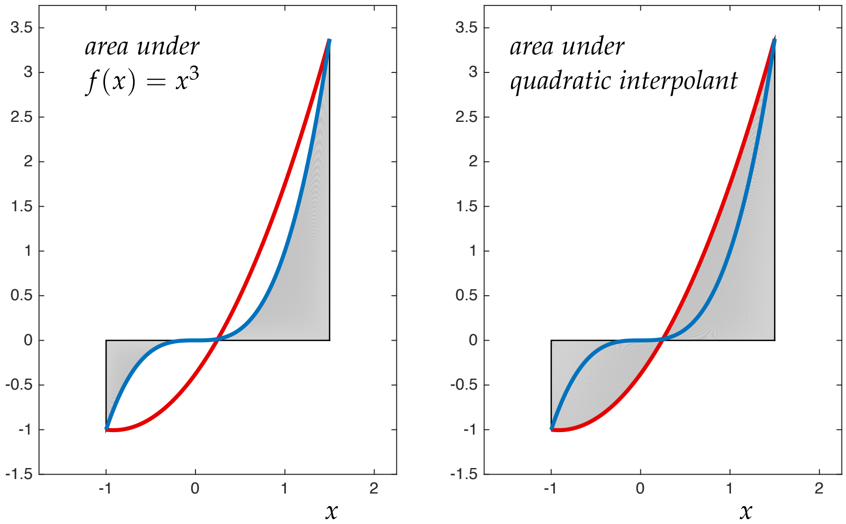

Figure 3.5: Simpson's rule applied to $f(x) = x^3$ on $x \in [-1, 3/2]$. The areas under $f(x)$ (blue) and its quadratic interpolant (red) are the same, even though the functions are quite different.

points, called *Clenshaw–Curtis quadrature*, does indeed converge. Section 3.4 describes this latter quadrature in more detail. Before discussing it, we describe a way to make Newton–Cotes rules more robust: integrate low-degree polynomials over subintervals of $[a, b]$.

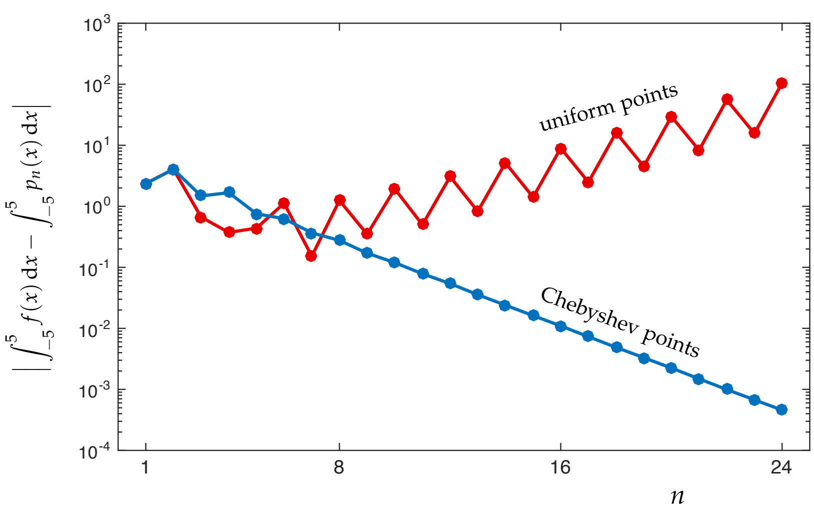

Figure 3.6: Integrating interpolants p_n at $n+1$ uniformly spaced points (red) and at Chebyshev points (blue) for Runge's function, $f(x) = 1/(1+x^2)$ over $x \in [-5, 5]$.

3.2.3 *Composite rules*

As an alternative to integrating a high-degree polynomial, one can pursue a simpler approach that is often very effective: Break the interval $[a, b]$ into subintervals, then apply a standard Newton–Cotes rule (e.g., trapezoid or Simpson) on each subinterval. Applying the trapezoid rule on n subintervals gives

$$\int_a^b f(x)\,\mathrm{d}x = \sum_{j=1}^{n} \int_{x_{j-1}}^{x_j} f(x)\,\mathrm{d}x \approx \sum_{j=1}^{n} \frac{(x_j - x_{j-1})}{2}\Big(f(x_{j-1} + f(x_j)\Big).$$

The standard implementation assumes that f is evaluated at uniformly spaced points between a and b, $x_j = a + jh$ for $j = 0, \ldots, n$ and $h = (b-a)/n$, giving the following famous formulation:

Composite Trapezoid rule:

$$\int_a^b f(x)\,\mathrm{d}x \approx \frac{h}{2}\Big(f(a) + 2\sum_{j=1}^{n-1} f(a+jh) + f(b)\Big).$$

(Of course, one can readily adjust this rule by partitioning $[a,b]$ into subintervals of different sizes.) The error in the composite trapezoid rule can be derived by summing up the error in each application of the trapezoid rule:

$$\begin{aligned}\int_a^b f(x)\,\mathrm{d}x - \frac{h}{2}\Big(f(a) + 2\sum_{j=1}^{n-1} f(a+jh) + f(b)\Big) &= \sum_{j=1}^{n}\Big(-\tfrac{1}{12}f''(\eta_j)(x_j - x_{j-1})^3\Big) \\ &= -\frac{h^3}{12}\sum_{j=1}^{n} f''(\eta_j)\end{aligned}$$

for $\eta_j \in [x_{j-1}, x_j]$. We can simplify these f'' terms by noting that $\frac{1}{n}(\sum_{j=1}^{n} f''(\eta_j))$ is the average of n values of f'' evaluated at points in the interval $[a,b]$. Naturally, this average cannot exceed the maximum or minimum value that f'' assumes on $[a,b]$, so there exist points $\xi_1, \xi_2 \in [a,b]$ such that

$$f''(\xi_1) \le \frac{1}{n}\sum_{j=1}^{n} f''(\eta_j) \le f''(\xi_2).$$

Thus the intermediate value theorem guarantees the existence of some $\eta \in [a,b]$ such that

$$f''(\eta) = \frac{1}{n}\sum_{j=1}^{n} f''(\eta_j).$$

We arrive at a bound on the error in the composite trapezoid rule.

Theorem 3.4. Let $f \in C^2[a,b]$. The error in the composite trapezoid rule over n intervals of uniform width $h = (b-a)/n$ is

$$\int_a^b f(x)\,\mathrm{d}x - \frac{h}{2}\Big(f(a) + 2\sum_{j=1}^{n-1} f(a+jh) + f(b)\Big) = -\frac{h^2}{12}(b-a)f''(\eta).$$

for some $\eta \in [a,b]$.

This error analysis has an important consequence: *the error for the composite trapezoid rule is only* $\mathcal{O}(h^2)$, not the $\mathcal{O}(h^3)$ we saw for the usual trapezoid rule (in which case $b-a=h$ since $n=1$).

A similar construction leads to the composite Simpson's rule. We now must ensure that n is even, since each interval on which we apply the standard Simpson's rule has width $2h$. Simple algebra leads to the following formula.

Composite Simpson's rule:

$$\int_a^b f(x)\,\mathrm{d}x \approx \frac{h}{3}\Big(f(a) \;+\; 4\sum_{j=1}^{n/2} f(a+(2j-1)h) \;+\; 2\sum_{j=1}^{n/2-1} f(a+2jh) \;+\; f(b)\Big).$$

Now use Theorem 3.3 to derive an error formula for the composite Simpson's rule, using the same approach as for the composite trapezoid rule.

Theorem 3.5. Let $f \in C^2[a,b]$. The error in the composite Simpson's rule over $n/2$ intervals of uniform width $2h = 2(b-a)/n$ is

$$\int_a^b f(x)\,\mathrm{d}x - \frac{h}{3}\Big(f(a) \;+\; 4\sum_{j=1}^{n/2} f(a+(2j-1)h) + \quad 2\sum_{j=1}^{n/2-1} f(a+2jh) \;+\; f(b)\Big) = -\frac{h^4}{180}(b-a)f^{(4)}(\eta)$$

for some $\eta \in [a,b]$.

The illustrations in Figure 3.7 compare the composite trapezoid and Simpson's rules for the same number of function evaluations. One can see that Simpson's rule, in this typical case, gives considerably better accuracy.

Reflect for a moment. Suppose you are willing to evaluate f a fixed number of times. How can you get the most bang for your buck? If f is smooth, a rule based on a high-order interpolant (such as the Clenshaw–Curtis and Gaussian quadrature rules we will present in a few lectures) are likely to give the best result. If f is not smooth (e.g., with kinks, discontinuous derivatives, etc.), then a robust composite rule would be a good option. (A famous special case: If the function f is sufficiently smooth and is periodic with period $b-a$, then the trapezoid rule converges *exponentially*.)

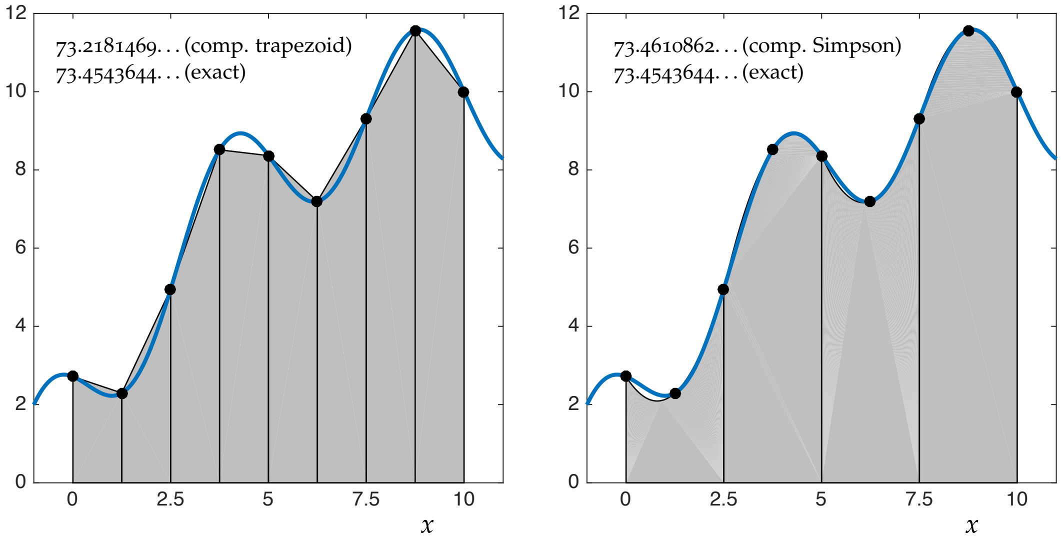

Figure 3.7: Composite trapezoid rule (left) and composite Simpson's rule (right).

3.2.4 *Adaptive Quadrature*

If f is continuous, we can attain arbitrarily high accuracy with composite rules by taking the spacing between function evaluations, h, to be sufficiently small. This might be necessary to resolve regions of rapid growth or oscillation in f. If such regions only make up a small proportion of the domain $[a, b]$, then uniformly reducing h over the entire interval will be unnecessarily expensive. One wants to concentrate function evaluations in the region where the function is the most ornery. Robust quadrature software adjusts the value of h locally to handle such regions. To learn more about such techniques, which are not foolproof, see W. Gander and W. Gautschi, "Adaptive quadrature—revisited," *BIT* 40 (2000) 84–101.

This paper criticizes the routines `quad` and `quad8` that were included in MATLAB version 5. In light of this analysis MATLAB improved its software, essentially incorporating the two routines suggested in this paper starting in version 6 as the routines `quad` (adaptive Simpson's rule) and `quadl` (an adaptive Gauss–Lobatto rule).