18.330 Lecture Notes: Numerical Integration, Part 1

Homer Reid

February 11, 2014

Contents

1 Numerical quadrature

Numerical integration is the art of approximating definite integrals by finite sums,

$$\int_a^b f(x)\,dx \approx \sum_{n=1}^{N} w_n f(x_n) \tag{1}$$

where the sample points x_n are some set of N points lying in the interval $[a, b]$, and w_n are an appropriately chosen set of weight coefficients. Numerical integration is also known as *numerical quadrature*, and the sets $\{x_n\}$ and $\{w_n\}$ are known as the *quadrature points* and the *quadrature weights*. An algorithm for choosing $\{x_n\}$ and $\{w_n\}$ is known as a *quadrature rule*.

The name of the game in numerical quadrature is to obtain accurate estimates of the integral in (1) with the smallest possible number of function samples. Think of $f(x)$ as an experimentally measured quantity that may take minutes or hours to evaluate. If your quadrature rule requires you to sample $f(x)$ at 10^6 points to get a decent estimate of the integral in (1), your project will be hopeless. We will find that unsophisticated quadrature rules may indeed require millions of points to yield decent accuracy, but with a little theoretical sophistication it is possible to do much better, obtaining 6 or more digits of accuracy with a few dozens or hundreds of samples.

Our study of numerical quadrature in 18.330 will unfold in multiple installments:

- We begin in these notes with the simplest approaches to constructing quadrature rules. We show that these rules work, but in most cases do not deliver particularly outstanding performance in the cost-vs-accuracy department (and, for this reason, are not commonly used in practice). We offer a simple error analysis to suggest why this might be. (We also note for future reference a couple of interesting cases in which the simple rules *do* yield excellent performance.)

- Then, later in the course (well into Unit 2), after we have introduced some more of the necessary theoretical background, we will discuss more sophisticated approaches to constructing quadrature rules. These rules do generally yield good performance and are commonly used in numerical practice.

- Meanwhile, there are several general points to be made about numerical quadrature that do not depend of the choice of quadrature rule. These points are addressed in these notes and remain equally valid later, after we have introduced more sophisticated quadrature schemes.

Thus, the content of Section 2 of these notes should *not* be taken as a statement of "this is how we recommend you do numerical quadrature in practice." Instead, think of this section as the first step in a journey that will culminate in a discussion of the right way to do numerical quadrature. On the other hand,

the more general points made in Section 3 of these notes will not be superseded by subsequent developments.

2 Newton-Cotes rules

One way to approximate $\int_a^b f(x)\,dx$ is to find a polynomial $P(x)$ that approximates f on the interval $[a, b]$ and integrate that instead, this being easy to do since we know how to integrate polynomials. This approach leads to the quadrature rules known as *Newton-Cotes rules*. There is a hierarchy of Newton-Cotes rules indexed by the degree of the polynomial; the pth-order Newton-Cotes rule uses a pth degree polynomial.

Of course, if f is not a polynomial itself then it won't be easy to find a single polynomial that approximates f over the whole interval. Instead, we consider subdividing the interval into N subintervals; on each subinterval we approximate f by a different polynomial chosen appropriately to mimic the behavior of f on that subinterval. The smaller we make the subintervals, the more accurately we will be able to approximate f by a polynomial, and thus the more accurate will be our approximation of the integral.[1]

2.1 The rectangular rule

In elementary calculus we learn to understand integration in terms of *Riemann sums*: we approximate the shape under the curve $f(x)$ as a union of rectangles; the area under the curve is approximately the sum of the areas of the rectangles. Then we consider a limiting process in which the width of the rectangles goes to zero, and the number of rectangles becomes infinite.

The simplest approach to numerical quadrature is to *arrest* this limiting process at some finite number of rectangles N. The combined area of those N rectangles then gives us our approximation to the integral. This may be thought of as the zeroth-degree Newton-Cotes rule, in which we approximate $f(x)$ on each interval by a zeroth-order polynomial (that is, a constant). The resulting quadrature rule is known as the *rectangular rule*, and it pictured graphically into Figure 1 for the case $N = 4$.

[1] Technically, the quadrature rule obtained by subdividing an interval into subintervals and applying a p-th order Newton-Cotes rule to each subinterval is known as a *composite* p-th order Newton-Cotes rule, as distinct from the *basic* Newton-Cotes rule which does not subdivide. The basic pth-order Newton-Cotes rule for an interval uses precisely $p + 1$ function samples in that interval, while the composite rule with M subintervals uses approximately Mp total function samples. However, in practice the phrase "Newton-Cotes quadrature" almost always refers to the use of composite Newton-Cotes rules, and the adjective "composite" is generally omitted.

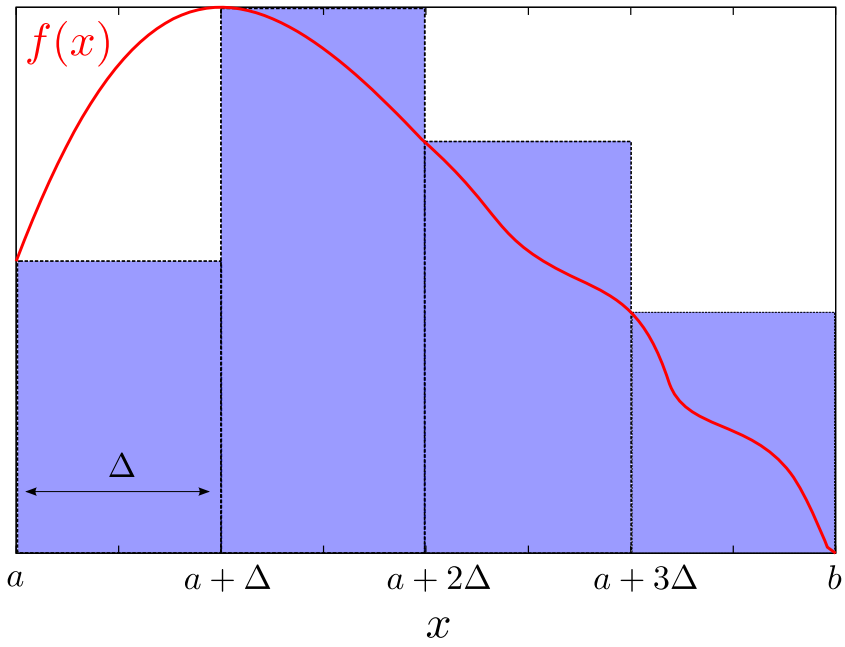

Figure 1: The area of the figure comprised of the shaded rectangles defines the $N = 4$ rectangular-rule approximation to $\int_a^b f(x)\,dx$.

If we subdivide the interval $[a, b]$ into N equal-length subintervals, each subinterval has width $\Delta = \frac{b-a}{N}$. The width of each rectangle in (1) is Δ. The height of the leftmost rectangle in (1) is $f(a)$; thus this rectangle has area $A = f(a) \cdot \Delta$. The height of the second-to-leftmost rectangle is $f(a + \Delta)$, so this rectangle has area $A = f(a + \Delta) \cdot \Delta$. Proceeding in this way and summing the areas of all the rectangles, the $N = 4$ rectangular-rule approximation to our integral is

$$\mathcal{I}^{\text{rect}}_{N=4} = f(a)\Delta \;+\; f(a+\Delta)\Delta \;+\; f(a+2\Delta)\Delta \;+\; f(a+3\Delta)\Delta.$$

More generally, the N-point rectangular-rule approximation to $\int_a^b f(x)\,dx$ is

$$\boxed{\mathcal{I}^{\text{rect}}_N = \sum_{n=0}^{N-1} f(a+n\Delta)\Delta.} \tag{2}$$

Note that this is a quadrature rule of the general form (1): The quadrature weights are $w_n = \Delta$ (the same weight for all n), and the quadrature points are $x_n = a + n\Delta$ for $n = 0, 1, \cdots, N-1$.[2] (Note that $f(b)$ is not referenced by this rule.)

[2] Alternatively, we could define the quadrature points to be $x_n = a + (n-1)\Delta$ for $n = 1, \cdots, N$. I find this slightly more cumbersome, but it agrees better with the convention of equation (1).

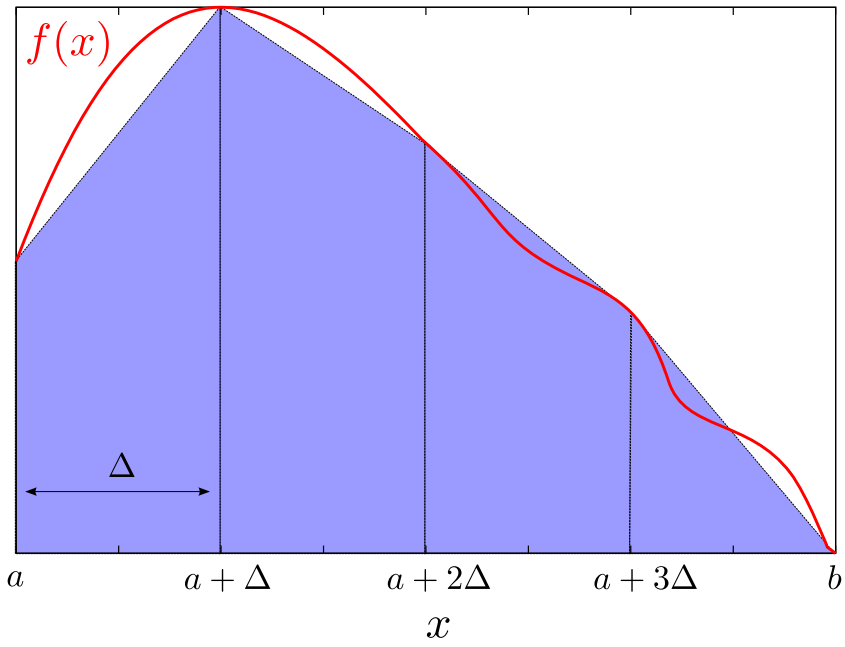

Figure 2: The area of the figure comprised of the shaded trapezoids defines the $N = 4$ trapezoidal-rule approximation to $\int_a^b f(x)\,dx$.

2.2 The trapezoidal rule

A quick glance at Figure 1 shows that the rectangles are a crude approximation to the area under the curve. We can obtain a better approximation by considering *trapezoids* instead of rectangles. A trapezoid is basically just a rectangle, except that its upper edge is slanted into a straight line connecting the values of $f(x)$ at the left and right endpoints of the interval. This is illustrated in Figure 2. One way to interpret Figure 2 is to say that in each interval we are approximating f by a first-order polynomial (a line), so this is a first-order Newton-Cotes rule. It is known as the *trapezoidal rule*.

To write down the formula for the trapezoidal rule, note that the areas of the trapezoids in Figure 2) are

$$\text{area of leftmost trapezoid} = \frac{1}{2}\Big[f(a) + f(a+\Delta)\Big]\Delta$$
$$\text{area of second-to-leftmost trapezoid} = \frac{1}{2}\Big[f(a+\Delta) + f(a+2\Delta)\Big]\Delta$$

etc. Continuing in this way and summing up the area of all the trapezoids, the

$N = 4$ trapezoidal-rule approximation to our integral is

$$\begin{aligned}
\mathcal{I}_{N=4}^{\text{trap}} &= \frac{1}{2}\Big[f(a) + f(a+\Delta)\Big]\Delta \\
&+ \frac{1}{2}\Big[f(a+\Delta) + f(a+2\Delta)\Big]\Delta \\
&+ \frac{1}{2}\Big[f(a+2\Delta) + f(a+3\Delta)\Big]\Delta \\
&+ \frac{1}{2}\Big[f(a+3\Delta) + f(a+b)\Big]\Delta
\end{aligned}$$

which we may write in collapsed form as

$$\mathcal{I}_{N=4}^{\text{trap}} = \left[\frac{1}{2}f(a) + f(a+\Delta) + f(a+2\Delta) + f(a+3\Delta) + \frac{1}{2}f(b)\right]\Delta.$$

More generally, the N-point trapezoidal-rule approximation to $\int_a^b f(x)\,dx$ is

$$\boxed{\mathcal{I}_N^{\text{trap}} = \frac{\Delta}{2}f(a) \;+\; \Delta\sum_{n=1}^{N-1} f(a+n\Delta) \;+\; \frac{\Delta}{2}f(b).} \tag{3}$$

Again, this is a quadrature rule of the general form (1). The quadrature points are $x_n = a + n\Delta$ for $n = 0, \cdots, N$. (Note that there is one more quadrature point than in the rectangular rule, namely, the point $x = b$.) The weights are

$$w_n = \begin{cases} \frac{1}{2}\Delta, & n = 0, N \\ \Delta, & n = 1, 2, \cdots, N-1. \end{cases}$$

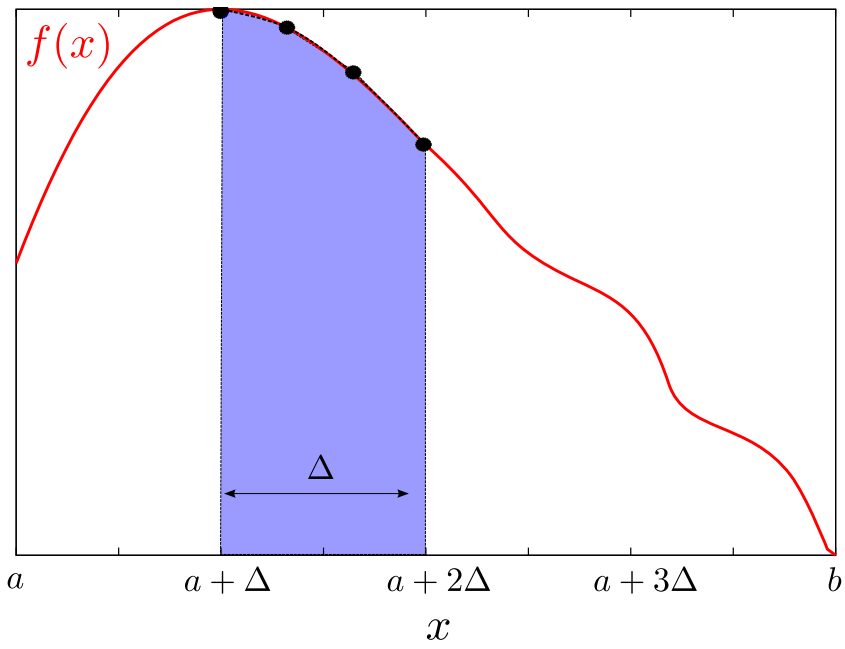

Figure 3: For a third-order Newton-Cotes rule, we write down the unique cubic polynomial that agrees with $f(x)$ at 4 evenly spaced points throughout one subinterval (dashed curve). The area under this curve (blue shape) is the third-order Newton-Cotes approximation to the integral of f over this subinterval.

2.3 Higher-order Newton-Cotes rules

The rectangular and trapezoidal rules correspond, respectively, to the zeroth-degree and first-degree Newton-Cotes rules. It is possible to continue this process. To derive the general p-th order Newton-Cotes rule, we proceed as follows:

1. Consider $p+1$ evenly spaced points in the subinterval $[x_n, x_{n+1}]$, including the endpoints. For example, for $p = 2$ we consider

$$x_n, \qquad \frac{1}{2}\Big[x_n + x_{n+1}\Big], \qquad x_{n+1}$$

   for $p = 3$ we consider

$$x_n, \qquad \frac{1}{3}\Big[x_n + x_{n+1}\Big], \qquad \frac{2}{3}\Big[x_n + x_{n+1}\Big], \qquad x_{n+1}$$

   etc.

2. Find the unique pth order polynomial $P(x)$ that agrees with $f(x)$ at the above points. For $p = 0$ (rectangular rule) this is just the constant $P(x) =$

$f(x_n)$. For $p = 1$ (trapezoidal rule) $P(x)$ is the line connecting $f(x_n)$ to $f(x_{n+1})$. For $p = 2$, $P(x)$ is the unique parabola running through the three points

$$\Big(x_n, f(x_n)\Big), \quad \Big(x_m, f(x_m)\Big), \quad \Big(x_{n+1}, f(x_{n+1})\Big) \quad \left(\text{where } x_m \equiv \frac{x_n + x_{n+1}}{2}\right).$$

In every case, $P(x)$ is a polynomial whose coefficients are linear combinations of function samples $f(x)$.

3. Integrate $P(x)$ from x_n to x_{n+1}. This yields an approximate expression relating the integral of $f(x)$ over the subinterval to a linear combination of values of f at the sample points.

4. Finally, combine the quadrature rules for all subintervals to obtain a composite quadrature rule for the overall interval.

The second-order Newton-Cotes rule is called *Simpson's rule,* for reasons discussed in the Appendix.

Newton-Cotes rules for high orders are not useful because of the *Runge phenomenon*: when we try to fit a high-order polynomial through evenly-spaced data points, the polynomial tends to blow up in the regions between the points, killing the accuracy of our integral estimate. We previewed the Runge phenomenon in our invitation lecture on the first day of class, and we will discuss it again in our unit on interpolation.

2.4 Heuristic error analysis

A heuristic, hand-wavy error analysis of the Newton-Cotes rules goes something like the following. This "analysis" is in fact more of a mnemonic device designed to remind us of how to arrive at the correct error estimate; it should not be mistaken for a rigorous demonstration.

- Consider a single subinterval of width Δ. Within this interval, we approximate $f(x)$ by a pth-degree polynomial P, where

$$P(x) = C_0 + C_1 x + \cdots + C_p x^p.$$

- There exists[3] a point x_0 within the interval such that the first $p+1$ terms in the Taylor-series expansion of f about x_0 agree with $P(x)$. In other words, we can write f in the form

$$f(x) = C_0 + C_1 x + \cdots + C_p x^p + C_{p+1} x^{p+1} + \cdots$$

[3]For example, if $p = 0$ (rectangular rule) we just take x_0 to be the left endpoint of the interval. For $p = 1$, the existence of the point x_0 is guaranteed by the mean value theorem: There is a point between x_n and x_{n+1} at which the derivative of f equals the slope of the straight line connecting $f(x_n)$ to $f(x_{n+1})$. Similar arguments hold for higher p values.

- This means that the difference between $f(x)$ and $P(x)$ is a polynomial that starts at order $p+1$:

  $$f(x) - P(x) = C_{p+1}x^{p+1} + C_{p+2}x^{p+2} + \cdots$$

  and thus the error in our approximate evaluation of the integral over this subinterval is

  $$\begin{aligned} \text{error in this subinterval} &= \int \big[f(x) - p(x)\big]\,dx \\ &= \int \Big[C_{p+1}x^{p+1} + \cdots\Big] \\ &\propto \Delta^{p+2} + \text{ higher order terms} \end{aligned}$$

  since integrating x^{p+1} over an interval of width Δ gives us something proportional to Δ^{p+2}.

Thus we conclude that the error in each subinterval is proportional to Δ^{p+2}. Of course, Δ is related to N (the number of subintervals) according to $\Delta \sim \frac{1}{N}$, and thus the result of the above analysis is

$$\text{error per subinterval } \propto \frac{1}{N^{p+2}}.$$

On the other hand, we have a total of N subintervals, and we must add together all the errors in all the subintervals to get the total error; this gives us an extra factor of N upstairs, so we find

$$\begin{aligned} \text{total error} &\propto \frac{1}{N^{p+2}} \cdot N \\ &= \frac{1}{N^{p+1}}. \end{aligned}$$

In particular, for the rectangular ($p = 0$) and trapezoidal ($p = 1$) rules, we find the expected convergence rates

$$\begin{aligned} \text{error in rectangular rule} &\propto \frac{1}{N} \\ \text{error in trapezoidal rule} &\propto \frac{1}{N^2}. \end{aligned}$$

Our analysis does not furnish the constant of proportionality, but that's OK because here we are only considered with the dependence on N.

2.5 Results

Figure 4 plots the convergence of the rectangular and trapezoidal rules for various values of N in an approximation of the integral

$$\mathcal{I} = \int_1^2 \log^2 x\,dx.$$

The quantity plotted in the figure is

$$\epsilon^{\text{rel}} \equiv \frac{|\mathcal{I}_N^{\text{approx}} - \mathcal{I}^{\text{exact}}|}{\mathcal{I}^{\text{exact}}}.$$

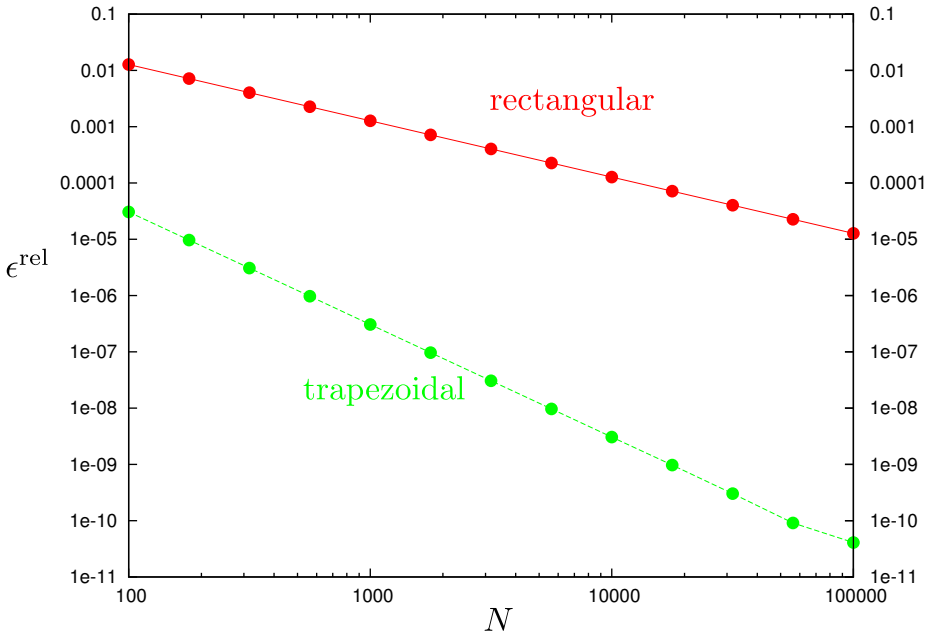

Figure 4: Convergence of the trapezoidal and rectangular rules vs. the number of intervals for evaluating the integral $\int_1^2 \log^2 x \, dx$.

This figure clearly demonstrates the expected convergence of both methods: As N increases over three orders of magnitude, the error in the rectangular rule decreases by three orders of magnitude, while the error in the trapezoidal rule decreases by six orders of magnitude.

This figure also demonstrates the weaknesses of these approaches: Even for this extremely smooth, well-behaved function, the trapezoidal rule requires almost 1000 function samples to achieve 6-digit accuracy, while the rectangular rule has still not achieved 6-digit accuracy even at $N = 10^6$ samples. The more sophisticated quadrature rules we introduce later will achieve 6-digit accuracy with many fewer function evaluations.

3 Miscellaneous points about numerical integration

As noted at the beginning of these notes, the Newton-Cotes approach is generally *not* the best method for deriving quadrature rules, and we will eventually replace it with better strategies that achieve superior accuracy-vs-cost performance. On the other hand, already at this point there are certain general points we can make about numerical quadrature that will remain valid even after we have moved on to more sophisticated quadrature rules.

3.1 Integration over infinite intervals

To handle improper integrals like

$$\int_0^\infty f(x)\,dx$$

we simply make a change of variables that maps the interval $[0,\infty]$ to a finite interval. There are many possible choices of map. One popular example is

$$x = \frac{u}{1-u}, \qquad dx = \frac{du}{(1-u)^2}$$

under which the integral $x \in [0,\infty]$ is mapped into $u \in [0,1]$. Our improper integral is then transformed into a proper integral:

$$\int_0^\infty f(x)\,dx = \int_0^1 f\Big(\frac{u}{1-u}\Big)\cdot\frac{du}{(1-u)^2}.$$

Note that the integral on the RHS appears to have a quadratic singularity as $u \to 1$. However, as $u \to 1$ the argument of f is approaching ∞, and f must vanish at ∞ (otherwise the integral $\int_0^\infty f(x)\,dx$ would not be convergent), so the singularity cancels.

3.2 Integrable singularities

Consider the following integral:

$$\mathcal{I} = \int_0^1 \frac{\exp(-x)}{\sqrt{x}}dx \tag{4}$$

Although the integrand blows up at the origin, the integral is perfectly well-defined. We say that we have an *integrable singularity* at the origin.

Note the important distinction between *integrable* and *non-integrable* singularities. The function $f(x) = \exp(x)/\sqrt{x}$ has an integrable singularity at the origin. In contrast, $f(x) = \exp(x)/x$ has a *non-integrable* singularity at the origin. There is no point in attempting to devise a strategy for estimating $\int_0^1 \exp(x)dx/x$, because the integral does not exist.

Numerical integration in the presence of integrable singularities is something of an art form. There are many approaches, some of which are better suited to particular problems than others; it's impossible to give a general prescription. Instead, we'll survey some common techniques.

Singularity subtraction

It may be possible to isolate the portion of the integrand that causes the difficulty and integrate it analytically. For example, in (4) we can write

$$\mathcal{I} = \underbrace{\int_0^1 \frac{1}{\sqrt{x}}\,dx}_{I_1} + \underbrace{\int_0^1 \frac{\exp(x)-1}{\sqrt{x}}\,dx}_{I_2}$$

The first integral may be evaluated analytically:

$$\mathcal{I}_1 = \int_0^1 \frac{dx}{\sqrt{x}} = \left|2\sqrt{x}\right|_0^1 = 2.$$

The second integral can't be evaluated analytically, but its integrand is now *nonsingular* at the origin, as we can see by Taylor-expanding the numerator:

$$\begin{aligned}\frac{\exp(x)-1}{\sqrt{x}} &= \frac{x - \frac{1}{2}x^2 + \frac{1}{6}x^3 + \cdots}{\sqrt{x}} \\ &= x^{1/2} - \frac{1}{2}x^{3/2} + \frac{1}{6}x^{5/2} + \cdots\end{aligned}$$

This function goes to zero politely as $x \to 0$, so we can evaluate $\mathcal{I}_2$ using standard (nonsingular) numerical quadrature techniques.

Singularity cancellation

Another strategy is to introduce new integration variables in such a way that the Jacobian of the transformation cancels troublesome factors in the denominator. For example, for (4) we can put

$$u = \sqrt{x}, \qquad du = \frac{dx}{2\sqrt{x}}$$

whereupon our integral becomes

$$\int_0^1 \frac{e^x}{\sqrt{x}}\,dx = 2\int_0^1 e^{u^2}\,du.$$

This integral has no singularities and may again be evaluated via straightforward numerical quadrature.

Epsilon expansion

If neither of the above methods work, you may try to introduce a "smoothing parameter" ϵ which removes the singularity for finite ϵ, then try extrapolating to the limit $\epsilon \to 0$. For example, in the above case, we might define

$$\mathcal{I}_\epsilon = \int_0^1 \frac{\exp(-x)dx}{\sqrt{\sqrt{x^2+\epsilon^2}}}$$

which is nonsingular for finite ϵ, but tends to $\mathcal{I}$ as $\epsilon \to 0$.

3.3 Adaptive quadrature

In general the functions we will be integrating behave differently in different portions of their domain of definition. For example, consider the function plotted in Figure 5.

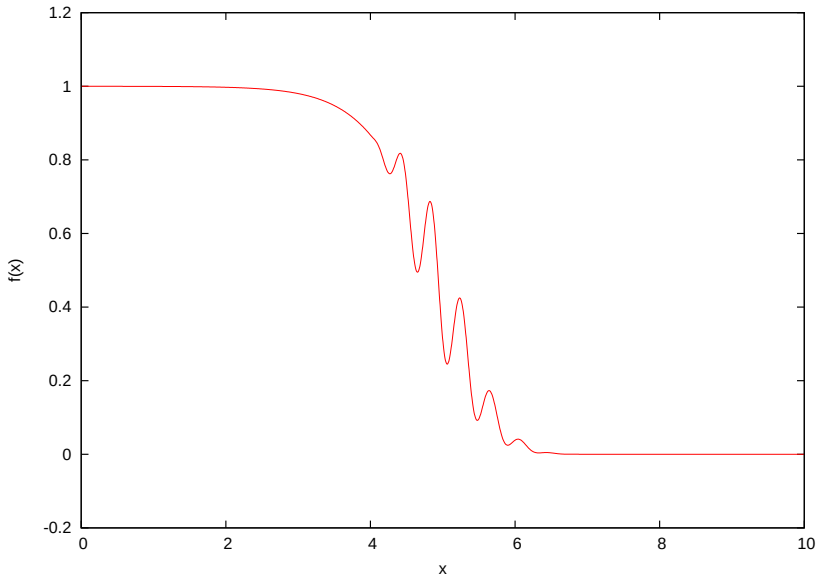

Figure 5: An example of a function whose rate of variation is different in different regions. If we use a trapezoidal rule to integrate this function from 0 to 10, we will clearly need to use rectangles of narrow widths Δ throughout the region $4 < x < 6$, but it would be wasteful to use the same small Δ value for $x < 4$ or $x > 6$.

Suppose we try to integrate this function from 0 to 10 using a trapezoidal rule. Clearly we are going to need a lot of very narrow trapezoids to capture the behavior of the function in the region $4 < x < 6$; we will need to take Δ quite small in that region to get accurate results. But then it would be wasteful to use such a small value of Δ in the regions $x < 4$ or $x > 6$; there we can obtain accurate results with much coarser resolution.

This motivates the notion of *adaptive* quadrature, in which we use quadrature rules of different accuracy for different regions of our integration domain. The most sophisticated integration codes implement a form of *automatic adaptivity:* They divide the range of integration into subintervals and estimate their own error on each subinterval. (To estimate the error in a quadrature scheme, you compare the difference between a coarse-grained and a fine-grained quadrature scheme—for example, trapezoidal rules with $N = 100$ and $N = 200$. If the results aren't very different, that means that refining the accuracy of the coarse-grained rule didn't change things much, and thus that the coarse-grained rule was already somewhat accurate.) When they deem the error to be too large, they recompute the integral using a more accurate quadrature rule (for example, a trapezoidal rule with a smaller value of Δ.)

The subject of adaptivity is particularly important in the context of ODE integrators, and we will revisit it during our discussion of that subject.

A Nomenclature for Newton-Cotes rules of various orders