Chapter 12

Numerical Integration and Differentiation

In the previous chapter, we developed tools for filling in reasonable values of a function $f(\vec{x})$ given a sampling of values $(\vec{x}_i, f(\vec{x}_i))$ in the domain of f. Obviously this interpolation problem is useful in itself for completing functions that are known to be continuous or differentiable but whose values only are known at a set of isolated points, but in some cases we then wish to study properties of these functions. In particular, if we wish to apply tools from calculus to f, we must be able to approximate its integrals and derivatives.

In fact, there are many applications in which numerical integration and differentiation play key roles in computation. In the most straightforward instance, some well-known functions are *defined* as integrals. For instance, the "error function" used as the cumulative distribution of a Gaussian or bell curve is written:

$$\mathrm{erf}(x) \equiv \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2}\, dt$$

Approximations of $\mathrm{erf}(x)$ are needed in many statistical contexts, and one reasonable approach to finding these values is to carry out the integral above numerically.

Other times, numerical approximations of derivatives and integrals are part of a larger system. For example, methods we develop in future chapters for approximating solutions to differential equations will depend strongly on these approximations. Similarly, in computational electrodynamics, *integral equations* solving for an unknown function ϕ given a kernel K and output f appear in the relationship:

$$f(\vec{x}) = \int_{\mathbb{R}^n} K(\vec{x}, \vec{y}) \phi(\vec{y})\, d\vec{y}.$$

These types of equations must be solved to estimate electric and magnetic fields, but unless the ϕ and K are very special we cannot hope to find such an integral in closed form, yet alone solve this equation for the unknown function ϕ.

In this chapter, we will develop assorted methods for numerical integration and differentiation given a sampling of function values. These algorithms are usually fairly straightforward approximations, so to compare them we will also develop some strategies that evaluate how well we expect different methods to perform.

12.1 Motivation

It is not hard to formulate simple applications of numerical integration and differentiation given how often the tools of calculus appear in the basic formulae and techniques of physics, statistics, and other fields. Here we suggest a few less obvious places where integration and differentiation appear.

Example 12.1 (Sampling from a distribution). *Suppose we are given a probability distribution $p(t)$ on the interval $[0,1]$; that is, if we randomly sample values according to this distribution, we expect $p(t)$ to be proportional to the number of times we draw a value near t. A common task is to generate random numbers distributed like $p(t)$.*

Rather than develop a specialized method to do so every time we receive a new $p(t)$, it is possible to make a useful observation. We define the cumulative distribution function *of p to be*

$$F(t) = \int_0^t p(x)\,dx.$$

Then, if X is a random number distributed evenly in $[0,1]$, one can show that $F^{-1}(X)$ is distributed like p, where F^{-1} is the inverse of F. Thus, if we can approximate F or F^{-1}, we can generate random numbers according to an arbitrary distribution p; this approximation amounts to integrating p, which may have to be done numerically when the integrals are not known in closed form.

Example 12.2 (Optimization). *Recall that most of our methods for minimizing and finding roots of a function f depended on having not only values $f(\vec{x})$ but also its gradient $\nabla f(\vec{x})$ and even Hessian H_f. We have seen that algorithms like BFGS and Broyden's method build up rough approximations of the derivatives of f during the process of optimization. When f has high frequencies, however, it may be better to approximate ∇f near the current iterate $\vec{x}_k$ rather than using values from potentially far-away points $\vec{x}_\ell$ for $\ell < k$.*

Example 12.3 (Rendering). *The* rendering equation *from ray tracing and other algorithms for high-quality rendering is an integral stating that the light leaving a surface is equal to the integral of the light coming into the surface over all possible incoming directions after it is reflected and diffused; essentially it states that light energy must be conserved before and after light interacts with an object. Algorithms for rendering must approximate this integral to compute the amount of light emitted from a surface reflecting light in a scene.*

Example 12.4 (Image processing). *Suppose we think of an image as a function of two variables $I(x,y)$. Many filters, including Gaussian blurs, can be thought of as* convolutions, *given by*

$$(I * g)(x,y) = \iint I(u,v)g(x-u,y-v)\,du\,dv.$$

For example, to blur an image we could take g to be a Gaussian; in this case $(I * g)(x,y)$ can be thought of as a weighted average of the colors of I near the point (x,y). In practice images are discrete grids of pixels, so this integral must be approximated.

Example 12.5 (Bayes' Rule). *Suppose X and Y are continuously-valued random variables; we can use $P(X)$ and $P(Y)$ to express the probabilities that X and Y take particular values. Sometimes, knowing X may affect our knowledge of Y. For instance, if X is a patient's blood pressure and Y is a patient's weight,* *then knowing a patient has high weight may suggest that they also have high blood pressure. We thus can also write conditional probability distributions $P(X|Y)$ (read "the probability of X given Y") expressing such relationships.*

A foundation of modern probability theory states that $P(X|Y)$ and $P(Y|X)$ are related as follows:

$$P(X|Y) = \frac{P(Y|X)P(X)}{\int P(Y|X)P(X)\,dY}$$

Estimating the integral in the denominator can be a serious problem in machine learning algorithms where the probability distributions take complex forms. Thus, approximate and often randomized integration schemes are needed for algorithms in parameter selection that use this value as part of a larger optimization technique.

12.2 Quadrature

We will begin by considering the problem of numerical integration, or *quadrature*. This problem–in a single variable– can be expressed as, "Given a sampling of n points from some function $f(x)$, find an approximation of $\int_a^b f(x)\,dx$." In the previous section, we presented several situations that boil down to exactly this technique.

There are a few variations of the problem that require slightly different treatment or adaptation:

- The endpoints a and b may be fixed, or we may wish to find a quadrature scheme that efficiently can approximate integrals for many (a, b) pairs.

- We may be able to query $f(x)$ at any x but wish to approximate the integral using relatively few samples, or we may be given a list of precomputed pairs $(x_i, f(x_i))$ and are constrained to using these data points in our approximation.

These considerations should be kept in mind as we design assorted algorithms for the quadrature problem.

12.2.1 Interpolatory Quadrature

Many of the interpolation strategies developed in the previous chapter can be extended to methods for quadrature using a very simple observation. Suppose we write a function $f(x)$ in terms of a set of basis functions $\phi_i(x)$:

$$f(x) = \sum_i a_i \phi_i(x).$$

Then, we can find the integral of f as follows:

$$\begin{aligned}\int_a^b f(x)\,dx &= \int_a^b \left[\sum_i a_i \phi_i(x)\right] dx \text{ by definition of } f\\ &= \sum_i a_i \left[\int_a^b \phi_i(x)\,dx\right]\\ &= \sum_i c_i a_i \text{ if we make the definition } c_i \equiv \int_a^b \phi_i(x)\,dx\end{aligned}$$

In other words, integrating f simply involves linearly combining the integrals of the basis functions that make up f.

Example 12.6 (Monomials)**.** *Suppose we write $f(x) = \sum_k a_k x^k$. We know*

$$\int_0^1 x^k \, dx = \frac{1}{k+1},$$

so applying the derivation above we know

$$\int_0^1 f(x) \, dx = \sum_k \frac{a_k}{k+1}.$$

In other words, in our notation above we have defined $c_k = \frac{1}{k+1}$.

Schemes where we integrate a function by interpolating samples and integrating the interpolated function are known as *interpolatory quadrature* rules; nearly all the methods we will present below can be written this way. Of course, we can be presented with a chicken-and-egg problem, if the integral $\int \phi_i(x) \, dx$ itself is not known in closed form. Certain methods in higher-order finite elements deal with this problem by putting extra computational time into making a high-quality numerical approximation of the integral of a single ϕ_i, and then since all the ϕ's have similar form apply change-of-coordinates formulas to write integrals for the remaining basis functions. This canonical integral can be approximated offline using a high-accuracy scheme and then reused.

12.2.2 Quadrature Rules

If we are given a set of $(x_i, f(x_i))$ pairs, our discussion above suggests the following form for a *quadrature rule* for approximating the integral of f on some interval:

$$Q[f] \equiv \sum_i w_i f(x_i).$$

Different weights w_i yield different approximations of the integral, which we hope become increasingly similar as we sample the x_i's more densely.

In fact, even the classical theory of integration suggests that this formula is a reasonable starting point. For example, the *Riemann integral* presented in many introductory calculus classes takes the form:

$$\int_a^b f(x) = \lim_{\Delta x_k \to 0} \sum_k f(\tilde{x}_k)(x_{k+1} - x_k)$$

Here, the interval $[a, b]$ is partitioned into pieces $a = x_1 < x_2 < \cdots < x_n = b$, where $\Delta x_k = x_{k+1} - x_k$ and $\tilde{x}_k$ is any point in $[x_k, x_{k+1}]$. For a fixed set of x_k's before taking the limit, this integral clearly can be written in the $Q[f]$ form above.

From this perspective, the choices of $\{x_i\}$ and $\{w_i\}$ completely determine a strategy for quadrature. There are many ways to determine these values, as we will see in the coming section and as we already have seen for interpolatory quadrature.

Example 12.7 (Method of undetermined coefficients)**.** *Suppose we fix $x_1, \ldots, x_n$ and wish to find a reasonable set of accompanying weights w_i so that $\sum_i w_i f(x_i)$ is a suitable approximation of the integral* *of f. An alternative to the basis function strategy listed above is to use the* method of undetermined *coefficients. In this strategy, we choose n functions $f_1(x), \dots, f_n(x)$ whose integrals are known, and ask that our quadrature rule recover the integrals of these functions exactly:*

$$\begin{aligned}
\int_a^b f_1(x)\,dx &= w_1 f_1(x_1) + w_2 f_1(x_2) + \cdots + w_n f_1(x_n)\\
\int_a^b f_2(x)\,dx &= w_1 f_2(x_1) + w_2 f_2(x_2) + \cdots + w_n f_2(x_n)\\
\vdots \quad & \quad \vdots\\
\int_a^b f_n(x)\,dx &= w_1 f_n(x_1) + w_2 f_n(x_2) + \cdots + w_n f_n(x_n)
\end{aligned}$$

This creates an $n \times n$ linear system of equations for the w_i's.

One common choice is to take $f_k(x) = x^{k-1}$, that is, to make sure that the quadrature scheme recovers the integrals of low-order polynomials. We know

$$\int_a^b x^k\,dx = \frac{b^{k+1} - a^{k+1}}{k+1}.$$

Thus, we get the following linear system of equations for the w_i's:

$$\begin{aligned}
w_1 + w_2 + \cdots + w_n &= b - a\\
x_1 w_1 + x_2 w_2 + \cdots + x_n w_n &= \frac{b^2 - a^2}{2}\\
x_1^2 w_1 + x_2^2 w_2 + \cdots + x_n^2 w_n &= \frac{b^3 - a^3}{2}\\
\vdots \quad & \quad \vdots\\
x_1^{n-1} w_1 + x_2^{n-1} w_2 + \cdots + x_n^{n-1} w_n &= \frac{b^2 - a^2}{2}
\end{aligned}$$

This system is exactly the Vandermonde system discussed in §11.1.1.

12.2.3 Newton-Cotes Quadrature

Quadrature rules when the $x_i^{\prime}s$ are evenly spaced in $[a, b]$ are known as *Newton-Cotes* quadrature rules. As illustrated in Figure NUMBER, there are two reasonable choices of evenly-spaced samples:

- *Closed* Newton-Cotes quadrature places x_i's at a and b. In particular, for $k \in \{1, \dots, n\}$ we take
$$x_k \equiv a + \frac{(k-1)(b-a)}{n-1}.$$

- *Open* Newton-Cotes quadrature does not place an x_i at a or b:

$$x_k \equiv a + \frac{k(b-a)}{n+1}.$$

After making this choice, the Newton-Cotes formulae simply apply polynomial interpolation to approximate the integral from a to b; the degree of the polynomial obviously must be $n-1$ to keep the quadrature rule well-defined.

In general, we will keep n relatively small. This way we avoid oscillatory and noise phenomena that occur when fitting high-degree polynomials to a set of data points. As in piecewise polynomial interpolation, we will then chain together small pieces into *composite* rules when integrating over a large interval $[a,b]$.

Closed rules. Closed Newton-Cotes quadrature strategies require $n \geq 2$ to avoid dividing by zero. Two strategies appear often in practice:

- The *trapezoidal* rule is obtained for $n = 2$ (so $x_1 = a$ and $x_2 = b$) by linearly interpolating from $f(a)$ to $f(b)$. It states that

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

- *Simpson's rule* comes from taking $n = 3$, so we now have

$$\begin{aligned} x_1 &= a \\ x_2 &= \frac{a+b}{2} \\ x_3 &= b \end{aligned}$$

  Integrating the parabola that goes through these three points yields

$$\int_a^b f(x)\,dx \approx \frac{b-a}{6}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right).$$

Open rules. Open rules for quadrature allow the possibility of $n = 1$, giving the simplistic midpoint rule:

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

Larger values of n yield rules similar to Simpson's rule and the trapezoidal rule.

Composite integration. Generally we might wish to integrate $f(x)$ with more than one, two, or three values x_i. It is obvious how to construct a composite rule out of the midpoint or trapezoidal rules above, as illustrated in Figure NUMBER; simply sum up the values along each interval. For example, if we subdivide $[a,b]$ into k intervals, then we can take $\Delta x \equiv \frac{b-a}{k}$ and $x_i \equiv a + i\Delta x$. Then, the composite midpoint rule is:

$$\int_a^b f(x)\,dx \approx \sum_{i=1}^{k} f\left(\frac{x_{i+1}+x_i}{2}\right)\Delta x$$

Similarly, the composite trapezoid rule is:

$$\begin{aligned}\int_a^b f(x)\,dx &\approx \sum_{i=1}^{k}\left(\frac{f(x_i)+f(x_{i+1})}{2}\right)\Delta x\\ &= \Delta x\left(\frac{1}{2}f(a)+f(x_1)+f(x_2)+\cdots+f(x_{k-1})+\frac{1}{2}f(b)\right)\\ &\qquad\text{by separating the two averaged values of } f \text{ in the first line and re-indexing}\end{aligned}$$

An alternative treatment of the composite midpoint rule is to apply the interpolatory quadrature formula from §12.2.1 to piecewise linear interpolation; similarly, the composite version of the trapezoidal rule comes from piecewise linear interpolation.

The composite version of Simpson's rule, illustrated in Figure NUMBER, chains together three points at a time to make parabolic approximations. Adjacent parabolas meet at even-indexed x_i's and may not share tangents. This summation, which only exists when n is even, becomes:

$$\begin{aligned}\int_a^b f(x)\,dx &\approx \frac{\Delta x}{3}\left[f(a)+2\sum_{i=1}^{n-2-1} f(x_{2i})+4\sum_{i=1}^{n/2} f(x_{2i-1})+f(b)\right]\\ &= \frac{\Delta x}{3}\left[f(a)+4f(x_1)+2f(x_2)+4f(x_3)+2f(x_4)+\cdots+4f(x_{n-1})+f(b)\right]\end{aligned}$$

Accuracy. So far, we have developed a number of quadrature rules that effectively combine the same set of $f(x_i)$'s in different ways to obtain different approximations of the integral of f. Each approximation is based on a different engineering assumption, so it is unclear that any of these rules is better than any other. Thus, we need to develop error estimates characterizing their respective behavior. We will use our Newton-Cotes integrators above to show how such comparisons might be carried out, as presented in CITE.

First, consider the midpoint quadrature rule on a single interval $[a,b]$. Define $c \equiv \frac{1}{2}(a+b)$. The Taylor series of f about c is:

$$f(x) = f(c)+f'(c)(x-c)+\frac{1}{2}f''(c)(x-c)^2+\frac{1}{6}f'''(c)(x-c)^3+\frac{1}{24}f''''(c)(x-c)^4+\cdots$$

Thus, by symmetry about c the odd terms drop out:

$$\int_a^b f(x)\,dx = (b-a)f(c)+\frac{1}{24}f''(c)(b-a)^3+\frac{1}{1920}f''''(c)(b-a)^5+\cdots$$

Notice that the first term of this sum exactly the estimate of $\int_a^b f(x)\,dx$ provided by the midpoint rule, so this rule is accurate up to $O(\Delta x^3)$.

Now, plugging a and b into our Taylor series for f about c shows:

$$\begin{aligned}f(a) &= f(c)+f'(c)(a-c)+\frac{1}{2}f''(c)(a-c)^2+\frac{1}{6}f'''(c)(a-c)^3+\cdots\\ f(b) &= f(c)+f'(c)(b-c)+\frac{1}{2}f''(c)(b-c)^2+\frac{1}{6}f'''(c)(b-c)^3+\cdots\end{aligned}$$

Adding these together and multiplying both sides by ${}^{b-a}/_{2}$ shows:

$$(b-a)\frac{f(a)+f(b)}{2} = f(c)(b-a)+\frac{1}{4}f''(c)(b-a)((a-c)^2+(b-c)^2)+\cdots$$

The $f'(c)$ term vanishes by definition of c. Notice that the left hand side is the trapezoidal rule integral estimate, and the right hand side agrees with our Taylor series for $\int_a^b f(x)\,dx$ up to the cubic term. In other words, the trapezoidal rule is also $O(\Delta x^3)$ accurate in a single interval.

We pause here to note an initially surprising result: The trapezoidal and midpoint rules have the same order of accuracy! In fact, examining the third-order term shows that the midpoint rule is approximately two times more accurate than the trapezoidal rule. This result seems counterintuitive, since the trapezoidal rule uses a linear approximation while the midpoint rule is constant. As illustrated in Figure NUMBER, however, the midpoint rule actually recovers the integral of linear functions, explaining its extra degree of accuracy.

A similar argument applies to finding an error estimate for Simpson's rule. [WRITE EXPLANATION HERE; OMIT FROM 205A]. In the end we find that Simpson's rule has error like $O(\Delta x^5)$.

An important caveat applies to this sort of analysis. In general, Taylor's theorem only applies when Δx is sufficiently *small*. If samples are far apart, then the drawbacks of polynomial interpolation apply, and oscillatory phenomena as discussed in Section NUMBER can cause unstable results for high-order integration schemes.

Thus, returning to the case when a and b are far apart, we now divide $[a, b]$ into intervals of width Δx and apply any of our quadrature rules inside these intervals. Notice that our total number of intervals is ${}^{b-a}/_{\Delta x}$, so we must multiply our error estimates by ${}^{1}/_{\Delta x}$ in this case. In particular, the following orders of accuracy hold:

- Composite midpoint: $O(\Delta x^2)$
- Composite trapezoid: $O(\Delta x^2)$
- Composite Simpson: $O(\Delta x^4)$

12.2.4 Gaussian Quadrature

In some applications, we can choose the locations x_i at which f is sampled. In this case, we can optimize not only the weights for the quadrature rule but also the locations x_i to get the highest quality. This observation leads to challenging but theoretically appealing quadrature rules.

The details of this technique are outside the scope of our discussion, but we provide one simple path to its derivation. In particular, as in Example 12.7, suppose that we wish to optimize $x_1, \ldots, x_n$ and $w_1, \ldots, w_n$ simultaneously to increase the order of an integration scheme. Now we have $2n$ instead of n knowns, so we can enforce equality for $2n$ examples:

$$\begin{aligned}
\int_a^b f_1(x)\,dx &= w_1 f_1(x_1) + w_2 f_1(x_2) + \cdots + w_n f_1(x_n) \\
\int_a^b f_2(x)\,dx &= w_1 f_2(x_1) + w_2 f_2(x_2) + \cdots + w_n f_2(x_n) \\
\vdots \quad & \quad \vdots \\
\int_a^b f_{2n}(x)\,dx &= w_1 f_n(x_1) + w_2 f_n(x_2) + \cdots + w_n f_n(x_n)
\end{aligned}$$

Now both the x_i's and the w_i's are unknown, so this system of equations is no longer linear. For example, if we wish to optimize these values for polynomials on the interval $[-1, 1]$ we would have to solve the following system of polynomials (CITE):

$$
\begin{aligned}
w_1 + w_2 &= \int_{-1}^{1} 1\, dx = 2 \\
w_1 x_1 + w_2 x_2 &= \int_{-1}^{1} x\, dx = 0 \\
w_1 x_1^2 + w_2 x_2^2 &= \int_{-1}^{1} x^2\, dx = \frac{2}{3} \\
w_1 x_1^3 + w_2 x_2^3 &= \int_{-1}^{1} x^3\, dx = 0
\end{aligned}
$$

It can be the case that systems like this have multiple roots and other degeneracies that depend not only on the choice of f_i's (typically polynomials) but also the interval over which we are approximating an integral. Furthermore, these rules are not *progressive*, in the sense that the set of x_i's for n data points has nothing in common with those for k data points when $k \neq n$, so it is difficult to reuse data to achieve a better estimate. On the other hand, when they are applicable Gaussian quadrature has the highest possible degree for fixed n. The *Kronrod* quadrature rules attempt to avoid this issue by optimizing quadrature with $2n + 1$ points while reusing the Gaussian points.

12.2.5 Adaptive Quadrature

As we already have shown, there are certain functions f whose integrals are better approximated with a given quadrature rule than others; for example, the midpoint and trapezoidal rules integrate linear functions with full accuracy while sampling issues and other problems can occur if f oscillates rapidly.

Recall that the Gaussian quadrature rule suggests that the placement of the x_i's can have an effect on the quality of a quadrature scheme. There still is one piece of information we have not used, however: the values $f(x_i)$. After all, these determine the quality of our quadrature scheme.

With this in mind, *adaptive* quadrature strategies examine the current estimate and generate new x_i where the integrand is more complicated. Strategies for adaptive integration often compare the output of multiple quadrature techniques, e.g. trapezoid and midpoint, with the assumption that they agree where sampling of f is sufficient (see Figure NUMBER). If they do not agree with some tolerance on a given interval, an additional sample point is generated and the integral estimates are updated.

ADD MORE DETAIL OR AN EXAMPLE; DISCUSS RECURSIVE ALGORITHM; GANDER AND GAUTSCHI

12.2.6 Multiple Variables

Many times we wish to integrate functions $f(\vec{x})$ where $\vec{x} \in \mathbb{R}^n$. For example, when $n = 2$ we might integrate over a rectangle by computing

$$\int_a^b \int_c^d f(x, y)\, dx\, dy.$$

More generally, as illustrated in Figure NUMBER¡ we might wish to find an integral $\int_\Omega f(\vec{x})\, d\vec{x}$, where Ω is some subset of $\mathbb{R}^n$.

A "curse of dimensionality" makes integration exponentially more difficult as the dimension increases. In particular, the number of samples of f needed to achieve comparable quadrature accuracy for an integral in $\mathbb{R}^k$ increases like $O(n^k)$. This observation may be disheartening but is somewhat reasonable: the more input dimensions for f, the more samples are needed to understand its behavior in all dimensions.

The simplest strategy for integration in $\mathbb{R}^k$ is the integrated integral. For example, if f is a function of two variables, suppose we wish to find $\int_a^b \int_c^d f(x,y)\,dx\,dy$. For fixed y, we can approximate the inner integral over x using a one-dimensional quadrature rule; then, we integrate these values over y using another quadrature rule. Obviously both integration schemes induce some error, so we may need to sample $\vec{x}_i$'s more densely than in one dimension to achieve desired output quality.

Alternatively, just as we subdivided $[a,b]$ into intervals, we can subdivide Ω into triangles and rectangles in 2D, polyhedra or boxes in 3D, and so on and use simple interpolatory quadrature rules in each piece. For instance, one popular option is to integrate the output of barycentric interpolation inside polyhedra, since this integral is known in closed form.

When n is high, however, it is not practical to divide the domain as suggested. In this case, we can use the randomized *Monte Carlo method*. In this case, we simply generate k random points $\vec{x}_i \in \Omega$ with, for example, uniform probability. Averaging the values $f(\vec{x}_i)$ yields an approximation of $\int_\Omega f(\vec{x})\,d\vec{x}$ that converges like $1/\sqrt{k}$ – independent of the dimension of Ω! So, in large dimensions the Monte Carlo estimate is preferable to the deterministic quadrature methods above.

MORE DETAIL ON MONTE CARLO CONVERGENCE AND CHOICE OF DISTRIBUTIONS OVER Ω

12.2.7 Conditioning

So far we have considered the quality of a quadrature method using accuracy values $O(\Delta x^k)$; obviously by this metric a set of quadrature weights with large k is preferable.

Another measure, however, balances out the accuracy measurements obtained using Taylor arguments. In particular, recall that we wrote our quadrature rule as $Q[f] \equiv \sum_i w_i f(x_i)$. Suppose we perturb f to some other $\hat{f}$. Define $\|f - \hat{f}\|_\infty \equiv \max_{x\in[a,b]} |f(x) - \hat{f}(x)|$. Then,

$$\begin{aligned}
\frac{|Q[f] - Q[\hat{f}]|}{\|f - \hat{f}\|_\infty} &= \frac{|\sum_i w_i(f(x_i) - \hat{f}(x_i))|}{\|f - \hat{f}\|_\infty} \\
&\leq \frac{\sum_i |w_i||f(x_i) - \hat{f}(x_i)|}{\|f - \hat{f}\|_\infty} \text{ by the triangle inequality} \\
&\leq \|\vec{w}\|_\infty \text{ since } |f(x_i) - \hat{f}(x_i)| \leq \|f - \hat{f}\|_\infty \text{ by definition.}
\end{aligned}$$

Thus, the stability or conditioning of a quadrature rule depends on the norm of the set of weights $\vec{w}$.

In general, it is easy to verify that as we increase the order of quadrature accuracy, the conditioning $\|\vec{w}\|$ gets worse because the w_i's take large negative values; this contrasts with the all-positive case, where conditioning is bounded by $b - a$ because $\sum_i w_i = b - a$ for polynomial interpolatory schemes and most low-order methods have only positive coefficients (CHECK). This fact is a reflection of the same intuition that we should not interpolate functions using high-order polynomials. Thus, in practice we usually prefer composite quadrature to high-order methods, that may provide better estimates but can be unstable under numerical perturbation.

12.3 Differentiation

Numerical integration is a relatively stable problem. in that the influence of any single value $f(x)$ on $\int_a^b f(x)\,dx$ shrinks to zero as a and b become far apart. Approximating the derivative of a function $f'(x)$, on the other hand, has no such stability property. From the Fourier analysis perspective, one can show that the integral $\int f(x)$ generally has lower frequencies than f, while differentiating to produce f' amplifies the high frequencies of f, making sampling constraints, conditioning, and stability particularly challenging for approximating f'.

Despite the challenging circumstances, approximations of derivatives usually are relatively easy to compute and can be stable depending on the function at hand. In fact, while developing the secant rule, Broyden's method, and so on we used simple approximations of derivatives and gradients to help guide optimization routines.

Here we will focus on approximating f' for $f : \mathbb{R} \to \mathbb{R}$. Finding gradients and Jacobians often is accomplished by differentiating in one dimension at a time, effectively reducing to the one-dimensional problem we consider here.

12.3.1 Differentiating Basis Functions

The simplest case for differentiation comes for functions that are constructed using interpolation routines. Just as in §12.2.1, if we can write $f(x) = \sum_i a_i\phi_i(x)$ then by linearity we know

$$f'(x) = \sum_i a_i\phi_i'(x).$$

In other words, we can think of the functions ϕ_i' as a *basis* for derivatives of functions written in the ϕ_i basis!

An example of this procedure is shown in Figure NUMBER. This phenomenon often connects different interpolatory schemes. For example, piecewise linear functions have piecewise constant derivatives, polynomial functions have polynomial derivatives of lower degree, and so on; we will return to this structure when we consider discretizations of partial differential equations. In the meantime, it is valuable to know in this case that f' is known with full certainty, although as in Figure NUMBER its derivatives may exhibit undesirable discontinuities.

12.3.2 Finite Differences

A more common case is that we have a function $f(x)$ that we can query but whose derivatives are unknown. This often happens when f takes on a complex form or when a user provides $f(x)$ as a subroutine without analytical information about its structure.

The definition of the derivative suggests a reasonable approach:

$$f'(x) \equiv \lim_{h\to 0} \frac{f(x+h) - f(x)}{h}$$

As we might expect, for a finite $h > 0$ with small $|h|$ the expression in the limit provides a possible value approximating $f'(x)$.

To substantiate this intuition, we can use Taylor series to write:

$$f(x+h) = f(x) + f'(x)h + \frac{1}{2}f''(x)h^2 + \cdots$$

Rearranging this expression shows:

$$f'(x) = \frac{f(x+h) - f(x)}{h} + O(h)$$

Thus, the following *forward difference approximation* of f' has linear convergence:

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

Similarly, flipping the sign of h shows that *backward differences* also have linear convergence:

$$f'(x) \approx \frac{f(x) - f(x-h)}{h}$$

We actually can improve the convergence of our approximation using a trick. By Taylor's theorem we can write:

$$\begin{aligned}
f(x+h) &= f(x) + f'(x)h + \frac{1}{2}f''(x)h^2 + \frac{1}{6}f'''(x)h^3 + \cdots \\
f(x-h) &= f(x) - f'(x)h + \frac{1}{2}f''(x)h^2 - \frac{1}{6}f'''(x)h^3 + \cdots \\
\Longrightarrow f(x+h) - f(x-h) &= 2f'(x)h + \frac{1}{3}f'''(x)h^3 + \cdots \\
\Longrightarrow \frac{f(x+h) - f(x-h)}{2h} &= f'(x) + O(h^2)
\end{aligned}$$

Thus, this *centered difference* gives an approximation of $f'(x)$ with quadratic convergence; this is the highest order of convergence we can expect to achieve with a divided difference. We can, however, achieve more accuracy by evaluating f at other points, e.g. $x + 2h$, although this approximation is not used much in practice in favor of simply decreasing h.

Constructing estimates of higher-order derivatives can take place by similar constructions. For example, if we add together the Taylor expansions of $f(x+h)$ and $f(x-h)$ we see

$$\begin{aligned}
f(x+h) + f(x-h) &= 2f(x) + f''(x)h^2 + O(h^3) \\
\Longrightarrow \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} &= f''(x) + O(h^2)
\end{aligned}$$

To predict similar combinations for higher derivatives, one trick is to notice that our second derivative formula can be factored differently:

$$\frac{f(x+h) - 2f(x) + f(x-h)}{h^2} = \frac{\frac{f(x+h)-f(x)}{h} - \frac{f(x)-f(x-h)}{h}}{h}$$

That is, our approximation of the second derivative is a "finite difference of finite differences." One way to interpret this formula is shown in Figure NUMBER. When we compute the forward difference approximation of f' between x and $x + h$, we can think of this slope as living at $x + {}^h\!/_2$; we similarly can use backward differences to place a slope at $x - {}^h\!/_2$. Finding the slope between these values puts the approximation back on x.

One strategy that can improve convergence of the approximations above is *Richardson extrapolation*. As an example of a more general pattern, suppose we wish to use forward differences to approximate f'. Define

$$D(h) \equiv \frac{f(x+h) - f(x)}{h}.$$

Obviously $D(h)$ approaches $f'(x)$ as $h \to 0$. More specifically, however, from our discussion in §12.3.2 we know that $D(h)$ takes the form:

$$D(h) = f'(x) + \frac{1}{2}f''(x)h + O(h^2)$$

Suppose we know $D(h)$ and $D(\alpha h)$ for some $0 < \alpha < 1$. We know:

$$D(\alpha h) = f'(x) + \frac{1}{2}f''(x)\alpha h + O(h^2)$$

We can write these two relationships in a matrix:

$$\left(\begin{array}{cc} 1 & \frac{1}{2}h \\ 1 & \frac{1}{2}\alpha h \end{array}\right) \left(\begin{array}{c} f'(x) \\ f''(x) \end{array}\right) = \left(\begin{array}{c} D(h) \\ D(\alpha h) \end{array}\right) + O(h^2)$$

Or equivalently,

$$\left(\begin{array}{c} f'(x) \\ f''(x) \end{array}\right) = \left(\begin{array}{cc} 1 & \frac{1}{2}h \\ 1 & \frac{1}{2}\alpha h \end{array}\right)^{-1} \left(\begin{array}{c} D(h) \\ D(\alpha h) \end{array}\right) + O(h^2)$$

That is, we took an $O(h)$ approximation of $f'(x)$ using $D(h)$ and made it into an $O(h^2)$ approximation! This clever technique is a method for *sequence acceleration*, since it improves the order of convergence of the approximation $D(h)$. The same trick is applicable more generally to many other problems by writing an approximation $D(h) = a + bh^n + O(h^m)$ where $m > n$, where a is the quantity we hope to estimate and b is the next term in the Taylor expansion. In fact, Richardson extrapolation even can be applied recursively to make higher and higher order approximations.

12.3.3 Choosing the Step Size

Unlike quadrature, numerical differentiation has a curious property. It appears that any method we choose can be arbitrarily accurate simply by choosing a sufficiently small h. This observation is appealing from the perspective that we can achieve higher-quality approximations without additional computation time. The catch, however, is that we must divide by h and compare more and more similar values $f(x)$ and $f(x+h)$; in finite-precision arithmetic, adding and/or dividing by near-zero values induces numerical issues and instabilities. Thus, there is a range of h values that are not large enough to induce significant discretization error and not small enough to make for numerical problems; Figure NUMBER shows an example for differentiating a simple function in IEEE floating point arithmetic.

12.3.4 Integrated Quantities

Not covered in CS 205A, fall 2013.

12.4 Problems

- Gaussian quadrature – always contains midpoints, strategy using orthogonal polynomials
- Adaptive quadrature
- Applications of Richardson extrapolation elsewhere