The general idea of numerical integration (or *quadrature*) is to approximate a given function $f(x)$ by something we know how to integrate (i.e. polynomials) and then to integrate the approximation. All of the well known integration rules (trapezoid, midpoint, left hand, right hand, simpson, etc.) that are taught in calculus courses are examples of this idea.

Recall the polynomial interpolation (using Lagrange interpolation) with error term

$$f(x) = \sum_{k=0}^{n} f(x_k) L_{n,k}(x) + \frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x - x_0)\dots(x - x_n).$$

Let's integrate this from $a = x_0$ to $b = x_n$ and see what we get

$$\begin{aligned}
\int_a^b f(x)dx &= \int_a^b \sum_{k=0}^{n} f(x_k) L_{n,k}(x)dx + \int_a^b \frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x - x_0)\dots(x - x_n)dx \\
&= \sum_{k=0}^{n} f(x_k) \int_a^b L_{n,k}(x)dx + \int_a^b \frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x - x_0)\dots(x - x_n)dx \\
&\approx \sum_{k=0}^{n} w_k f(x_k)
\end{aligned}$$

where $w_k = \int_a^b L_{n,k}(x)$, which is completely independent of $f(x)$. We will spend some time analyzing the error term as well as the form of the approximation here.

1 Analyzing the error

Let's start with the error term

$$\int_a^b \frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x - x_0)\dots(x - x_n)dx.$$

Unfortunately, due to the presence of $\xi(x)$, this formula is somewhat difficult to reason about. It may feel unnerving that the derivative appears in the error, but this makes sense since the more quickly a function changes, the less accurate a (low degree) polynomial will be. If it happens that $(x - x_0)\dots(x - x_n)$ is nonnegative (or nonpositive), then we can apply the (weighted) mean value theorem for integrals

$$\int_a^b \frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x - x_0)\dots(x - x_n)dx = \frac{f^{(n+1)}(\varphi)}{(n+1)!}\int_a^b (x - x_0)\dots(x - x_n)dx$$

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

Question 1.1. Which interpolating polynomial degrees does this apply to?

This is mainly useful for constant (left/right/midpoint) and linear (trapezoidal) approximations, and for certain Taylor series computations.

Question 1.2. Use this estimate to bound the error of constant and linear approximations for numerical integration

Let's consider the constant case. From the interpolation perspective, we obtain the same error bound for any choice of constant.

$$(\max_x f'(x)) \int_a^b (x - x_0) dx = f'(\varphi) \frac{(b-a)^2}{2}.$$

It turns out we can do better in the midpoint (i.e. $x_0 = \frac{a+b}{2}$) case by using Taylor approximation of order 1 (with error term)

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(\xi(x))}{2}(x - x_0)^2$$

to approximate the error instead of integrating the interpolation error. When we integrate the Taylor approximation, we obtain

$$\int_a^b f(x)dx = \int_a^b f(x_0)dx + \int_a^b f'(x_0)(x - x_0) + \frac{f''(\xi(x))}{2}(x - x_0)^2 dx$$

The first term of the second integral is zero due to the symmetry of $x - x_0$ on the interval $[a, b]$. Then applying the weighted mean value theorem to the remaining error term, we get

$$\frac{f''(\varphi)}{2} \int_a^b (x - x_0)^2 dx = \frac{(b-a)^3 f''(\varphi)}{24}$$

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

For linear approximation, we have the interpolating polynomial

$$f(x) = f(x_0)\frac{x - x_1}{x_0 - x_1} + f(x_1)\frac{x - x_0}{x_1 - x_0} + \frac{f^{(2)}(\xi(x))}{2}(x - x_0)(x - x_1).$$

Integrating both sides and applying the weighted mean value theorem gives

$$\int_a^b f(x)dx = \int_a^b f(x_0)\frac{x - x_1}{x_0 - x_1} + f(x_1)\frac{x - x_0}{x_1 - x_0}dx + \frac{f''(\varphi)}{2}\int_a^b (x - x_0)(x - x_1)dx$$

for some $\varphi \in [a, b]$. The first term is the approximation and the second is the error

$$\begin{aligned}
\frac{f''(\varphi)}{2}\int_a^b (x - x_0)(x - x_1)dx &= \frac{f''(\varphi)}{2}\int_a^b x^2 - (x_0 + x_1)x + x_0x_1 dx \\
&= \frac{f''(\varphi)}{2}\left[\frac{x^3}{3} - (x_0 + x_1)\frac{x^2}{2} + x_0x_1x\right]_a^b \\
&= -\frac{f''(\varphi)}{2}\frac{(b-a)^3}{6}.
\end{aligned}$$

You'll see better ways to analyze the linear approximation (trapezoid rule) on the homework.

Outside of these two cases, the conditions of the weighted mean value theorem don't apply directly to the interpolation error, so we need to use the (slightly) weaker bound

$$\int_a^b \frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x - x_0)\ldots(x - x_n)dx \leq \max_x \frac{|f^{(n+1)}(x)|}{(n+1)!}\int_a^b |(x - x_0)\ldots(x - x_n)|dx.$$

We can apply this bound to quadratic approximation at the points x_0, x_1, x_2, also known as Simpson's rule. The error term is bounded above by

$$\max_x \frac{|f^{(3)}(x)|}{6} \int_a^b |(x-a)(x-\frac{a+b}{2})(x-b)|dx = \max_x \frac{|f^{(3)}(x)|}{6}\frac{(b-a)^4}{32}.$$

However, this error bound is not the best we can do. It turns out we can again make clever use of Taylor approximation to get a better bound by expanding $f(x)$ around $x_1 = \frac{a+b}{2}$

$$f(x) = f(x_1) + f'(x_1)(x-x_1) + \frac{f''(x_1)}{2}(x-x_1)^2 + \frac{f'''(x_1)}{6}(x-x_1)^3 + \frac{f^{(4)}(\xi(x))}{24}(x-x_1)^4.$$

Integrating gives

$$\int_a^b f(x)dx = \int_a^b f(x_1) + f'(x_1)(x-x_1) + \frac{f''(x_1)}{2}(x-x_1)^2 + \frac{f'''(x_1)}{6}(x-x_1)^3 + \frac{f^{(4)}(\xi(x))}{24}(x-x_1)^4 dx$$

Note that the odd powers of $x - x_1$ do not appear below because of the symmetry of the integration interval. To simplify notation, let $h = \frac{a+b}{2}$ denote the length of the intervals used in Simpson's rule.

$$\begin{aligned}
&= \left[f(x_1)(x-x_1) + \frac{f''(x_1)}{6}(x-x_1)^3\right]_a^b + \int_a^b \frac{f^{(4)}(\xi(x))}{24}(x-x_1)^4 dx \\
&= 2hf(x_1) + \frac{h^3}{3}f''(x_1) + \int_a^b \frac{f^{(4)}(\xi(x))}{24}(x-x_1)^4 dx
\end{aligned}$$

It takes a bit of effort, but it's not too difficult to show that the second derivative $f''(x_1)$ can be approximated (using the numerical differentiation techniques that you will see on the homework) with error by

$$f''(x_1) = \frac{1}{h^2}(f(x_0) - 2f(x_1) + f(x_2)) - \frac{h^2}{12}f^{(4)}(\xi)$$

for some $\xi \in [a, b]$. Combining this with the previous results gives

$$\begin{aligned}
&2hf(x_1) + \frac{h^3}{3}(\frac{1}{h^2}(f(x_0) - 2f(x_1) + f(x_2)) - \frac{h^2}{12}f^{(4)}(\xi)) + \int_a^b \frac{f^{(4)}(\xi(x))}{24}(x-x_1)^4 dx \\
&= 2hf(x_1) + \frac{h}{3}f(x_0) - \frac{2h}{3}f(x_1) + \frac{h}{3}f(x_2) - \frac{h^5}{36}f^{(4)}(\xi)) + \int_a^b \frac{f^{(4)}(\xi(x))}{24}(x-x_1)^4 dx \\
&= \frac{h}{3}(f(x_0) + 4f(x_1) + f(x_2)) - \frac{h^5}{36}f^{(4)}(\xi)) + \int_a^b \frac{f^{(4)}(\xi(x))}{24}(x-x_1)^4 dx.
\end{aligned}$$

Since $(x - x_1)^4 \geq 0$, the remaining error term can be bounded by applying the weighted mean value theorem to get the error bound

$$\int_a^b \frac{f^{(4)}(\xi(x))}{24}(x-x_1)^4 dx = \frac{f^{(4)}(\varphi)}{120}(x-x_1)^5|_a^b = \frac{f^{(4)}(\varphi)}{60}h^5,$$

giving an overall error of

$$-\frac{h^5}{36}f^{(4)}(\xi) + \frac{h^5}{60}f^{(4)}(\varphi) = -\frac{h^5}{90}f^{(4)}(\xi_1),$$

where $h = \frac{b-a}{2}$ for some $\xi_1 \in [a, b]$. (Be careful: it takes some effort to replace φ and ξ by a uniform ξ_1.) The conclusion of all these computations is that we can often do better than the obvious error bound using some clever calculations.

Observe that these error bounds all involve derivatives of $f(x)$. If it happens that the relevant derivative is equal to zero, then these results say that the integration scheme can compute the integrals exactly. For this reason, we are interested in "higher order" integration schemes which are able to compute more functions exactly.

One issue here is that we also have various powers of h appearing in the error terms. This means that as we integrate over larger intervals, the error grows very quickly. We'll see one way to handle this below.

1.1 Composite rules

Recall that interpolation using e.g. evenly spaced interpolation points encounters the *Runge phenomenon* where the error can grow very quickly near the endpoints. This is especially noticeable if the interval is large. The way we fixed this for polynomials was to pick better interpolation points or to split up the interval into smaller pieces.

Sampling different points can not always be used, since we are often given data and asked to integrate it numerically as opposed to being able to specify the points freely. In these situations, one way to decrease the error is to integrate piecewise over the larger intervals to avoid the Runge phenomenon. The idea of the approximation is the same, but now we split up the interval $[a, b]$ into n pieces and apply the approximation to each interval separately.

For example, for the midpoint (left, right) and trapezoid rules, we can directly apply the previous results to each interval. If the interval is calculated using evenly spaced points $x_i = a + ih$, then the total error becomes

$$\sum_{i=1}^{n} \frac{h^3 f''(\varphi_i)}{24} = \frac{h^3}{24}\sum_i f''(\varphi_i)$$

If we want to give a rough bound, we can replace each $f''(\varphi_i)$ with its the maximum value $\max_x f''(x)$. If we want to be a bit more clever, observe first that $f''(x)$ must attain its maximum and minimum values on the compact interval $[a, b]$. Then for $x \in [a, b]$, we have

$$\min_x f''(x) \leq \frac{1}{n}\sum_i f''(\varphi_i) \leq \max_x f''(x)$$

so that by the intermediate value theorem, there is some $\varphi \in [a, b]$ such that

$$f''(\varphi) = \frac{1}{n}\sum_i f''(\varphi_i).$$

Plugging this in to our formula above, we obtain

$$\frac{h^3 n}{24} f''(\varphi).$$

Since we divided the interval $[a, b]$ into n intervals each of length h, we have the relation $hn = b - a$. Using this, we can rewrite the error above

$$\frac{(b-a)^3}{24n^2} f''(\varphi)$$

Question 1.3. Repeat the above for the composite trapezoid rule.

We can repeat the same analysis we just did for midpoint to find

$$f''(\varphi) \frac{(b-a)^3}{12n^2}$$

Simpson's rule is a bit trickier to make composite, due to the fact that it requires three points for interpolation.

Question 1.4. How can we make Simpson's rule composite?

Divide $[a, b]$ into $2n$ subintervals, and use adjacent pairs for Simpson's rule. Thus, we do Simpson's rule on n intervals, so the overall error will be bounded by

$$f^{(4)}(\varphi) \frac{nh^5}{90} = f^{(4)}(\varphi) \frac{(b-a)^5}{180(2n)^4}.$$

(Note in this case that $2nh = b - a$ instead of $nh = b - a$.) We can proceed similarly, making a composite version of any type of polynomial interpolation. We could even use splines, Hermite, etc. Things are looking great at this point: if we just split the interval into smaller and smaller pieces, we can significantly reduce the error.

Question 1.5. Are there any issues with blindly increasing the number of subintervals for composite integration?

We've seen before that adding too many terms leads to bad things because of floating point errors that accumulate over time. Miraculously, this does not happen for numerical integration. To exhibit this, let's investigate what happens with Simpson's rule.

$$\frac{h}{3}(f(a) + 4f(\frac{a+b}{2}) + f(b))$$

and for simplicity let $e_i = f(x_i) - fl(f(x_i))$. Then the accumulated error of Composite Simpson's rule is

$$\frac{h}{3}\left[e_0 + 2\sum_{j=1}^{n/2-1} e_{2j} + 4\sum_{j=1}^{n/2} e_{2j-1} + e_n\right] \leq \frac{h}{3}\left[|e_0| + 2\sum_{j=1}^{n/2-1} |e_{2j}| + 4\sum_{j=1}^{n/2} |e_{2j-1}| + |e_n|\right]$$

Let c denote a uniform bound on the roundoff errors (alternatively, we could have worked with relative error instead), so we obtain

$$\leq \frac{h}{3}\left[c + 2(\frac{n}{2} - 1)c + 4\frac{n}{2}c + c\right] = \frac{h}{3}3nc = hnc = \leq \frac{b-a}{2}c.$$

Note that this error is independent of h or n, so only depends on the length of the interval. In other words, no matter how many subintervals we use, the floating point roundoff errors only contribute a constant error to our overall result.

1.2 Trapezoid rule revisited

The error estimates for numerical integration that we obtain directly from interpolation are good to have, but we can often do better, as we now demonstrate for the composite trapezoid rule. We'll need to start with a better way to write the error of the trapezoid rule that does not use the interpolation error. The trick for doing this will be the repeated application of integration by parts.

$$\begin{aligned}
\int_i^{i+1} f(x)dx &= f(x)\left(x + \frac{2i+1}{2}\right)\Big|_i^{i+1} - \int_i^{i+1}\left(x + \frac{2i+1}{2}\right)f'(x)dx \\
&= \frac{1}{2}f(i+1) + \frac{1}{2}f(i) - \left(f'(x)\frac{1}{2}\left(x + \frac{2i+1}{2}\right)^2\Big|_i^{i+1} - \int_i^{i+1}\frac{1}{2}\left(x + \frac{2i+1}{2}\right)^2 f''(x)dx\right) \\
&= \frac{1}{2}(f(i+1) + f(i)) - \frac{1}{2}\frac{1}{2^2}(f'(i+1) - f'(i)) + \int_i^{i+1}\frac{1}{2}\left(x + \frac{2i+1}{2}\right)^2 f''(x)dx \\
&= \frac{1}{1!}\frac{1}{2}(f(i+1) + f(i)) - \frac{1}{2!}\frac{1}{2^2}(f'(i+1) - f'(i)) \\
&\qquad + \frac{1}{3!}f''(x)\left(x + \frac{2i+2}{2}\right)^3\Big|_i^{i+1} - \int_i^{i+1}\frac{1}{3!}\left(x + \frac{2i+1}{2}\right)^3 f'''(x)dx \\
&= \frac{1}{1!}\frac{1}{2}(f(i+1) + f(i)) - \frac{1}{2!}\frac{1}{2^2}(f'(i+1) - f'(i)) + \frac{1}{3!}\frac{1}{2^3}(f''(i+1) + f''(i)) \\
&\qquad - \int_i^{i+1}\frac{1}{3!}\left(x + \frac{2i+1}{2}\right)^3 f'''(x)dx.
\end{aligned}$$

Repeating this process infinitely, we establish the following series representation of the integral.

$$\begin{aligned}
\int_i^{i+1} f(x)dx &= \frac{1}{1!}\frac{1}{2}(f(i+1) + f(i)) - \frac{1}{2!}\frac{1}{2^2}(f'(i+1) - f'(i)) + \frac{1}{3!}\frac{1}{2^3}(f''(i+1) + f''(i)) - \dots \\
&= \sum_{j=0}^{\infty}\frac{1}{(2j+1)!}\frac{1}{2^{2j+1}}(f^{(2j)}(i+1) + f^{(2j)}(i)) \\
&\qquad - \sum_{j=1}^{\infty}\frac{1}{(2j)!}\frac{1}{2^{2j}}(f^{(2j-1)}(i+1) - f^{(2j-1)}(i)).
\end{aligned}$$

Now, if we rearrange the formula we just found, we obtain

$$\frac{1}{2}(f(i+1)+f(i)) = \int_i^{i+1} f(x)dx + \sum_{j=1}^{\infty} \frac{1}{(2j)!}\frac{1}{2^{2j}}(f^{(2j-1)}(i+1) - f^{(2j-1)}(i)) - \sum_{j=1}^{\infty} \frac{1}{(2j+1)!}\frac{1}{2^{2j+1}}(f^{(2j)}(i+1) + f^{(2j)}(i)).$$

Notice that the left hand side is exactly the trapezoid rule, while the right hand side is the integral plus some error terms. The error terms we found in this problem have the issue that the "even" terms (those in the first sum with even derivatives) are added together rather than subtracted. Because we want most terms to cancel out when we look at the composite trapezoid rule, we need to manipulate the equation somehow to eliminate these terms. To do this, we'll apply the repeated integration by parts we performed above to $f^{(2k)}(x)$ in place of $f(x)$ (for any $k \geq 1$)

$$\int_i^{i+1} f^{(2k)}(x)dx = \sum_{j=0}^{\infty} \frac{1}{(2j+1)!}\frac{1}{2^{2j+1}}(f^{(2j+2k)}(i+1) + f^{(2j+2k)}(i)) - \sum_{j=1}^{\infty} \frac{1}{(2j)!}\frac{1}{2^{2j}}(f^{(2j-1+2k)}(i+1) - f^{(2j-1+2k)}(i)).$$

Rearranging the formula in the same way as before, we get for each $k \geq 1$ the equation

$$\begin{aligned}\frac{1}{2}(f^{(2k)}(i+1)+f^{(2k)}(i)) &= \int_i^{i+1} f^{(2k)}(x)dx + \sum_{j=1}^{\infty} \frac{1}{(2j)!}\frac{1}{2^{2j}}(f^{(2j-1+2k)}(i+1) - f^{(2j-1+2k)}(i)) \\ &\quad - \sum_{j=1}^{\infty} \frac{1}{(2j+1)!}\frac{1}{2^{2j+1}}(f^{(2j+2k)}(i+1) + f^{(2j+2k)}(i)) \\ &= \sum_{j=0}^{\infty} \frac{1}{(2j)!}\frac{1}{2^{2j}}(f^{(2j-1+2k)}(i+1) - f^{(2j-1+2k)}(i)) \\ &\quad - \sum_{j=1}^{\infty} \frac{1}{(2j+1)!}\frac{1}{2^{2j+1}}(f^{(2j+2k)}(i+1) + f^{(2j+2k)}(i))\end{aligned}$$

Although it doesn't seem like we've gotten much out of this work, we observe that the equations we just found express the sum $f^{(2k)}(i+1) + f^{(2k)}(i)$ in terms of differences of odd derivatives (which we like) and sums of even derivatives of higher order (which we don't like, but can now handle). Thus, when we plug in these equations (successively, for each $k \geq 1$) into the original equation, we finally obtain

$$\frac{1}{2}(f(i+1)+f(i)) = \int_i^{i+1} f(x)dx + \sum_{j=1}^{\infty} b_j(f^{2j-1}(i+1) - f^{2j-1}(i)) \tag{1}$$

for some unknown constants b_j. (Equation 1 is known as the *Euler-Maclaurin formula* and the constants b_j are closely related to *Bernoulli numbers*.) Since we are making infinitely many substitutions, we need to justify why the coefficients b_j are well-defined. To see an example, consider the following. We first substitute the following formula for $k = 1$

$$\frac{1}{2}(f^{(2)}(i+1) + f^{(2)}(i)) = \sum_{j=0}^{\infty} \frac{1}{(2j)!}\frac{1}{2^{2j}}(f^{(2j+1)}(i+1) - f^{(2j+1)}(i)) - \sum_{j=1}^{\infty} \frac{1}{(2j+1)!}\frac{1}{2^{2j+1}}(f^{(2j+2)}(i+1) + f^{(2j+2)}(i))$$

into the equation we found above. Note that this substitution only affects terms of the formula that have derivative order higher than 1. This substitution gives

$$\frac{1}{2}(f(i+1) + f(i)) = \int_i^{i+1} f(x)dx + \sum_{j=1}^{\infty} c_{2j-1}(f^{(2j-1)}(i+1) - f^{(2j-1)}(i)) - \sum_{j=2}^{\infty} c_{2j}(f^{(2j)}(i+1) + f^{(2j)}(i))$$

for some well-defined constants c_j. Note that the c_j are well-defined since each one is just the sum of (at most) two contributions – one from the original formula and one from the substitution. Now making the substitution for $k = 2$,

$$\frac{1}{2}(f^{(4)}(i+1) + f^{(4)}(i)) = \sum_{j=0}^{\infty} \frac{1}{(2j)!}\frac{1}{2^{2j}}(f^{(2j+3)}(i+1) - f^{(2j+3)}(i)) - \sum_{j=1}^{\infty} \frac{1}{(2j+1)!}\frac{1}{2^{2j+1}}(f^{(2j+4)}(i+1) + f^{(2j+4)}(i))$$

we obtain

$$\frac{1}{2}(f(i+1) + f(i)) = \int_i^{i+1} f(x)dx + \sum_{j=1}^{\infty} d_{2j-1}(f^{(2j-1)}(i+1) - f^{(2j-1)}(i)) - \sum_{j=3}^{\infty} d_{2j}(f^{(2j)}(i+1) + f^{(2j)}(i))$$

for some constants d_j that depend on at most three contributions – one from the original formula, one from the substitution for $k = 1$, and one from the substitution for $k = 2$. Note however that the $k = 2$ substitution only affects terms with derivative of order 3 or higher. This means that the terms of orders 1 no longer depend on future substitutions. This will be the general trend, and we can see that in general the coefficient b_j is only affected by the substitutions for $k = 1, \ldots, j$ so each b_j is well-defined. (Thus, no particular coefficient b_j is affected by infinitely many substitutions.)

Now we add all the left-hand sides to get

$$\frac{1}{2}(f(0)+f(1))+\frac{1}{2}(f(1)+f(2))+\cdots+\frac{1}{2}(f(n-1)+f(n)) = \frac{1}{2}(f(0)+2f(1)+\cdots+2f(n-1)+f(n))$$

which is exactly the composite trapezoid rule for integration. Next adding all the right-hand sides, we get

$$
\begin{aligned}
&\int_0^1 f(x)dx + \sum_{j=1}^{\infty} b_j(f^{2j-1}(1) - f^{2j-1}(0)) \\
&+ \int_1^2 f(x)dx + \sum_{j=1}^{\infty} b_j(f^{2j-1}(2) - f^{2j-1}(1)) \\
&+ \dots \\
&+ \int_{n-1}^n f(x)dx + \sum_{j=1}^{\infty} b_j(f^{2j-1}(n) - f^{2j-1}(n-1)) \\
&= \int_0^n f(x)dx + \sum_{j=1}^{\infty} b_j(f^{2j-1}(n) - f^{2j-1}(0))
\end{aligned}
$$

Thus the error of the trapezoid rule depends only on the odd derivatives of f at the endpoints of the integration interval..

Although the Euler-Maclauren formula looks like a bit of a mess, it leads to many practical applications.

1. The error of (composite) trapezoid rule depends *only* on the endpoints. This naturally leads to the idea of endpoint corrected trapezoid rule (ECTR), which modifies the composite trapezoid rule near the endpoints of the integration interval $[0, n]$ to decrease the error.

2. If it happens that $f^{(2j+1)}(0) = f^{(2j+1)}(n)$ for all $j = 0, 1, \dots,$ then the equation we found says that the trapezoid rule will exactly compute the desired integral. For example, any periodic function with period p such that $\frac{n}{p}$ is an integer will satisfy this condition.

3. The form of this error is suitable for a technique known as *Richardson extrapolation*. In this context, this application is known as *Romberg integration*.

   We begin with the sequence $R_{1,1}, R_{2,1}, \dots$ where $R_{k,1}$ denotes the composite trapezoid rule for $\int_a^b f(x)dx$ with $n = 2^k$ subintervals. We saw before that the error of this sequence behaves like $\approx h^2$, where $h = \frac{b-a}{n}$. This means that we can decrease the error by increasing the number of subintervals (i.e. letting k grow large in $R_{k,1}$. However, we saw in class that this can cost an incredibly amount of computation time.

   As an alternative, we can instead take our computations for $R_{k,1}$ and use them to define further sequences

   $$R_{k,j} = R_{k,j-1} + \frac{1}{4^{j-1} - 1}(R_{k,j-1} - R_{k-1,j-1})$$

   for $k = j, j+1, \dots$. Miraculously, these new sequences $R_{j,j}, R_{j+1,j}, \dots$ have an error behaving like $\approx h^{2j}$ in approximating $\int_a^b f(x)dx$, so we can get great gain in performance without performing additional integrations.

4. If f is not infinitely differentiable, but only $2m$ times differentiable, we can truncate the Euler-Maclaurin formula 1 to obtain an error of order $2m$, and everything we just said applies with this new error estimate instead.

As a final remark, it's not important to use interpolation points at $0, 1, \ldots, n$. Instead, the only thing required by this method is that we used equispaced points to obtain reasonably interpretable coefficients b_j.

1.3 Adaptive quadrature

While composite integration schemes are very effective in approximating an integral, they suffer from one practical concern.

Question 1.6. What shortcoming(s) does composite integration face?

The difficulty in composite integration is that we may be required to use a very small step size to get the desired accuracy, leading to a very large number of computations to be performed. Sometimes, if the function is sufficiently complicated, this is unavoidable. However, in some situations, we may be able to save a lot of computations if we use a very fine subdivision on only the complicated parts of the function. Otherwise, for example if the function is constant on a large interval, then we don't need complicated machinery to approximate the integral on that piece. Roughly speaking, the goal of adaptive integration is to "spend" the desired approximation error evenly over an interval.

Question 1.7. How can we detect where to use finer subdivisions?

Let's suppose that we have an oracle that can tell us what the error is on a given interval. Then the adaptive integration procedure is as follows. As part of the input, we will need a desired error; we use ε as a default here. We first compute an approximation of the integral using the entire interval. If this value is less than ε, we output this value. Otherwise, split the interval into two subintervals. If the approximation is less than $\frac{\varepsilon}{2}$ for both subintervals, add them together and output the result. Otherwise, split up the offending regions (could be one or both) into smaller regions and determine whether the approximation is accurate to within $\frac{\varepsilon}{4}$. Continue this halving procedure until we reach the required error.

Question 1.8. What problems could arise with the procedure as stated?

Unfortunately, this process is not guaranteed to terminate. Nevertheless, it's quite effective in practice. More importantly, we need a way to estimate the error at each step, or else this procedure is useless. There's no silver bullet here, and we'll exhibit the particular case of Simpson's rule. Suppose that we want to approximate $\int_a^b f(x)dx$ to a specified tolerance $\varepsilon > 0$. For ease of notation, let $I(a, b)$ denote the integral $\int_a^b f(x)dx$ and $S(a, b)$ denote the Simpson's rule approximation to $I(a, b)$

$$S(a, b) = \frac{h}{3}(f(a) + 4f(a + h) + f(b))$$

with an error of $\frac{h^5}{90} f^{(4)}(\xi)$ for some $\xi \in [a, b]$, where $h = \frac{b-a}{2}$. We start with $S(a, b)$ as an approximation to $I(a, b)$. Unfortunately, it's usually not reasonable to try to figure out

$f^{(4)}(\xi)$ in the error term directly. In order to get an error approximation not involving $f^{(4)}(\xi)$, we compute the next steps $S(a, \frac{a+b}{2})$ and $S(\frac{a+b}{2}, b)$ (defined analogously to above), which is just composite Simpson's rule with $n = 2$ intervals. From last time, this gives

$$\begin{aligned} I(a,b) &= S(a, \frac{a+b}{2}) + S(\frac{a+b}{2}, b) - \frac{2(h/2)^5}{90} f^{(4)}(\widetilde{\xi}) \\ &= S(a, \frac{a+b}{2}) + S(\frac{a+b}{2}, b) - \frac{1}{16}\frac{h^5}{90} f^{(4)}(\widetilde{\xi}) \end{aligned}$$

for some $\widetilde{\xi} \in [a,b]$. If we assume that $f^{(4)}(\xi) \approx f^{(4)}(\widetilde{\xi})$, then setting our two approximations equal to each other gives

$$S(a,b) - \frac{h^5}{90} f^{(4)}(\xi) \approx S(a, \frac{a+b}{2}) + S(b, \frac{a+b}{2}) - \frac{1}{16}\frac{h^5}{90} f^{(4)}(\xi).$$

Simplifying this gives us

$$\frac{h^5}{90} f^{(4)}(\xi) \approx \frac{16}{15}(S(a,b) - S(a, \frac{a+b}{2}) - S(\frac{a+b}{2}, b))$$

and plugging this back in to our two step composite Simpson calculation gives

$$\int_a^b f(x)dx - S(a, \frac{a+b}{2}) - S(\frac{a+b}{2}, b) = -\frac{1}{16}\frac{h^5}{90} f^{(4)}(\xi) \approx -\frac{1}{15}(S(a,b) - S(a, \frac{a+b}{2}) - S(\frac{a+b}{2}, b)).$$

In other words, the error of our approximation can be approximated itself by the approximations! In particular, the two step composite Simpson calculation agrees with $I(a,b)$ approximately 15 times better than it agrees with $S(a,b)$. Thus, if

$$S(a,b) - S(a, \frac{a+b}{2}) - S(\frac{a+b}{2}, b) < 15\varepsilon,$$

then we will have the desired accuracy in the approximation

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

If the difference is larger than 15ε, then we can repeat this process on each subinterval $[a, \frac{a+b}{2}]$ and $[\frac{a+b}{2}, b]$, this time asking for the error to be smaller than $\frac{\varepsilon}{2}$. If this approximation fails, then the offending subintervals will be further divided.

Question 1.9. Why is it reasonable to aim for a smaller error each iteration?

In practice, we may want to decrease the bound from 15ε to 10ε or even smaller to account for the fact that we used the approximation $f^{(4)}(\xi) \approx f^{(4)}(\widetilde{\xi})$ for the fourth derivative. If the function is known to have large variations, then we should be more conservative in our choice. The rough reason why this works is that roughly speaking, we get a 15 times improvement each time so even if the allowable error is decreasing by a factor of 2 each time, $15^n > 2^n$ by a lot so we hope to obtain a suitable bound eventually.

1.3.1 Implementing adaptive quadrature

This is the first procedure we've encountered in this class that requires some thought about coding up. Since Simpson's rule only uses evaluations at the specified interpolation points, we will recompute the same function values many times over the course of adaptive quadrature if we're not careful.

This problem lends itself well to a recursive approach. Suppose that functions are implemented efficiently so that asking for $f(x)$ multiple times will store the result for future computations each time a new request is made. Suppose that we have a program to compute $S(a, b)$ for any a, b. Then we can use the following to perform one round of the adaptive quadrature. Suppose the inputs are a, b, e, f where we want to compute $\int_a^b f(x)dx$ to the tolerance e.

1. Compute $S(a, b), S(a, \frac{a+b}{2}), S(\frac{a+b}{2}, b)$.
2. If $|S(a, b) - S(a, \frac{a+b}{2}) - S(\frac{a+b}{2}, b)| < 15e$, then output $S(a, \frac{a+b}{2}) + S(\frac{a+b}{2}, b)$.
3. If not, recursively invoke this procedure with the new inputs $a, \frac{a+b}{2}, \frac{e}{2}, f$ to compute the left subinterval and $\frac{a+b}{2}, b, \frac{e}{2}, f$ to compute the right subinterval.

Alternatively, we could implement this using a *stack*. An example of this might be as follows

1. Initialize the stack to contain (a, b, e, f).
2. Initialize a running total to keep track of the approximation.
3. While the stack is not empty,
   (a) Remove the last element (a, b, e, f) of the stack.
   (b) Compute $|S(a, b) - S(a, \frac{a+b}{2}) - S(\frac{a+b}{2}, b)|$. If this is smaller than $15e$, then add $S(a, \frac{a+b}{2}) + S(\frac{a+b}{2}, b)$ to the running total.
   (c) If it's larger, then add $(a, \frac{a+b}{2}, \frac{e}{2}, f)$ and $(\frac{a+b}{2}, b, \frac{e}{2}, f)$ to the stack.
4. Output the running total.

While we discussed the details using Simpson's method, this could be replaced with any desired quadrature method (though the error analysis would need to be redone to see what the new criterion for dividing an interval into subintervals should be). One way to avoid infinite loops is to default to standard composite integration if the procedure goes too deep into recursion. However, in practice, while examples can be crafted that will never yield an acceptable error, adaptive quadrature performs quite well.

2 Gaussian Quadrature

Recall that the form of the numerical approximation to the integral was given by

$$\int_a^b f(x)dx \approx \sum_{k=0}^{n} w_k f(x_k),$$

where $w_k = \int_a^b L_{n,k}(x)dx$ was the "weight" obtained by integrating the corresponding Lagrange polynomial. We saw from the error term that this correctly integrates any polynomial of degree at most n. Furthermore, we've also seen ways to decrease the error by performing a more careful analysis or using composite rules.

However, all of our previous analyses were using equally spaced points (for example, if the data is given to us this way, we have to work with what we're given). On the other hand, this can significantly decrease the accuracy of the approximation. If we're given the freedom to choose the interpolation points, an alternative approach would be to find the "best" formula

$$\sum_{k=0}^{n} w_k f(x_k)$$

to approximate a given integral by carefully choosing the weights c_k and the interpolation points x_k.

Question 2.1. We saw this idea with interpolation in the form of Chebyshev points. Do you think we should use the Chebyshev points here?

While the Chebyshev points give better polynomial interpolations, they're not exactly what we're after here. It's always a good idea to get a better approximation of the function we're interested in integrating, but remember here that it's more important to us to get the value of the integral close even if the function we choose to approximate it with is not perfectly accurate. Because we're allowing ourselves to choose the weights w_k and the points x_k, this gives us $2n+2$ degrees of freedom to play around with, meaning that we should expect to be able to correctly integrate any polynomial of degree up to $2n+1$.

Question 2.2. Try this problem for $n+1=2$ on the interval $[a,b]=[-1,1]$. That is, try to find w_0, w_1 and x_0, x_1 such that any polynomial of degree up to $2+2-1=3$ is correctly integrated on the interval $[-1,1]$ by the formula

$$w_0 f(x_0) + w_1 f(x_1).$$

To do this, since integrals are linear, it's enough to make sure that we correctly integrate $f(x)$ when $f(x)$ is equal to $1, x, x^2, x^3$. Therefore, we need to find w_0, w_1, x_0, x_1 such that

$$w_0 + w_1 = \int_{-1}^{1} 1dx = 2 \qquad w_0x_0 + w_0x_1 = \int_{-1}^{1} xdx = 0$$

$$w_0x_0^2 + w_1x_1^2 = \int_{-1}^{1} x^2dx = \frac{2}{3} \qquad w_0x_0^3 + w_1x_1^3 = \int_{-1}^{1} x^3dx = 0$$

With some work, we can find the solution $w_0 = w_1 = 1$ and $x_0 = \frac{-1}{\sqrt{3}} = -x_1$, from which we obtain the approximation

$$\int_{-1}^{1} f(x)dx \approx f(\frac{-1}{\sqrt{3}}) + f(\frac{1}{\sqrt{3}}).$$

In general, we'll obtain a very non-linear system of equations to solve (albeit with a lot of structure like the one we saw here). While we could try to solve this system in general, it turns out that we can use some other clever ideas to get at the method more easily.

Question 2.3. Do we know any method that exactly approximates polynomials up to degree $2n+1$?

While we don't want to use information about the derivatives of f, it turns out to be very useful to think about Hermite interpolation in this case. Remember that Hermite interpolation asks to match both function values and derivative values at $n+1$ points, and had the format

$$f(x) \approx \sum_{k=0}^{n} A_{n,k}(x) f(x_k) + B_{n,k}(x) f'(x_k)$$

with error term

$$\frac{f^{(2n+2)}(\xi(x))}{(2n+2)!}(x-x_0)^2 \dots (x-x_n)^2$$

where

$$A_{n,k}(x) = (1 - 2L'_{n,k}(x_k)(x-x_k))L_{n,k}(x)^2$$
$$B_{n,k}(x) = (x-x_k)L_{n,k}(x)^2.$$

When we integrate this approximation, we obtain

$$\begin{aligned}\int_{-1}^{1} f(x)dx &\approx \sum_{k=0}^{n} f(x_k) \int_{-1}^{1} A_{n,k}(x)dx + f'(x_k) \int_{-1}^{1} B_{n,k}(x)dx \\ &\approx \sum_{k=0}^{n} f(x_k)a_k + f'(x_k)b_k.\end{aligned}$$

If we can select the interpolation points so that $b_k = 0$ (Gauß's idea), then we can obtain the desired approximation. Recall that

$$L_{n,k}(x) = \frac{(x-x_0)\dots(x-x_{k-1})(x-x_{k+1})\dots(x-x_n)}{(x_k-x_0)\dots(x_k-x_{k-1})(x_k-x_{k+1})\dots(x_k-x_n)}$$

so we have that

$$\begin{aligned}\int_{-1}^{1} B_{n,k}(x)dx &= \int_{-1}^{1}(x-x_k)L_{n,k}(x)^2 dx \\ &= \frac{1}{(x_k-x_0)\dots(x_k-x_{k-1})(x_k-x_{k+1})\dots(x_k-x_n)} \int_{-1}^{1}(x-x_0)\dots(x-x_n)L_{n,k}(x)dx.\end{aligned}$$

This integral to be zero for all k if and only if

$$\int_{-1}^{1}(x-x_0)\dots(x-x_n)p(x)dx = 0$$

for all polynomials $p(x)$ of degree at most n, since the Lagrange polynomials $L_{n,k}(x)$ form a basis for the space of the polynomials of degree at most n.

Remark 2.4. If we consider the vector space of integrable functions, the integral of the product defines an inner product on this space. We are very often interested in having orthonormal bases for (subspaces of) the space of integrable functions. For example, this is the entire idea behind Fourier theory.

In any case, our goal is to choose $x_0, \dots, x_n$ so that the above integral vanishes for any polynomial of degree at most n. It is equivalent to ask for this for the particular polynomials $1, x, x^2, \dots, x^n$ by linearity.

Question 2.5. Use this procedure to find the points x_0, x_1 for the $n+1=2$ case.

For $n+1=2$, we need to satisfy

$$\int_{-1}^{1}(x-x_0)(x-x_1)dx = \int_{-1}^{1}(x-x_0)(x-x_1)xdx = 0.$$

Evaluating these, we get

$$\frac{1}{3}+x_0x_1 = 0 \qquad x_0+x_1=0$$

and solving this system gives the two points $x_0 = -\frac{1}{\sqrt{3}}, x_1 = \frac{1}{\sqrt{3}}$ just as we found before. Next, we further compute

$$L_{1,0}(x) = \frac{x-x_1}{x_0-x_1} \qquad L_{1,1}(x) = \frac{x-x_0}{x_1-x_0}$$

$$L'_{1,0}(x) = \frac{1}{x_0-x_1} \qquad L'_{1,1}(x) = \frac{1}{x_1-x_0}$$

so plugging these in to the coefficients a_j, we find

$$a_0 = \int_{-1}^{1} A_0(x)dx = \int_{-1}^{1}(1+2\sqrt{3}(x+\frac{1}{\sqrt{3}}))\frac{3}{4}(x-\frac{1}{\sqrt{3}})^2dx = 1$$

and similarly for a_1. This recovers our previous result in a generalizable way. However, we now need a way to evaluate these integrals in a reasonable way.

Question 2.6. How can we simplify this integral?

To simplify finding this coefficient, observe that

$$A_{n,k}(x) = (1-2L'_{n,k}(x_k)(x-x_k))L_{n,k}(x)^2 = L_{n,k}(x)^2 - 2L'_{n,k}(x_k)B_{n,k}(x).$$

Because we chose the integration points so that $\int_{-1}^{1} B_{n,k}(x)dx = 0$, we thus obtain

$$\int_{-1}^{1} A_{n,k}(x)dx = \int_{-1}^{1} L_{n,k}(x)^2dx - 2L'_{n,k}(x_k)\int_{-1}^{1} B_{n,k}(x)dx = \int_{-1}^{1} L_{n,k}(x)^2dx.$$

It turns out we can even drop the square and just integrate directly, but it's useful in this form to know that the weights are nonnegative. Integrating the (square of) the Lagrange interpolating polynomials is not too difficult, but we need a better method for determining the integration points, since we ended up having to solve a (nonlinear) system of equations anyway.

To solve this issue, recall that we've already defined the integration points as the roots of a polynomial. In particular, the $n+1$ Gaussian quadrature points are exactly the roots of the unique (monic) polynomial $P_{n+1}(x)$ of degree $n+1$ satisfying

$$\int_{-1}^{1} P_{n+1}(x)p(x)dx = 0$$

for any polynomial $p(x)$ of degree at most n.

Question 2.7. Why is this polynomial unique? Find the degree 2 polynomial $P_2(x)$.

Let's denote the coefficients by $P_2(x) = x^2 + ax + b$. Then by linearity, we can again test orthogonality against the simple polynomials $1, x$.

$$0 = \int_{-1}^{1} (x^2 + ax + b)1dx = \frac{2}{3} + 2b \implies b = -\frac{1}{3}$$
$$0 = \int_{-1}^{1} (x^2 + ax + b)xdx = \frac{2a}{3} \implies a = 0$$

so we obtain $P_2(x) = x^2 - \frac{1}{3}$ which has the two roots $\pm\frac{1}{\sqrt{3}}$. This description only involves solving a *linear* system of equations, which we can do without much difficulty, and provides a satisfactory method, because we can then find the weights using any root-finding procedure that we've discussed before (e.g. bisection, fixed point, Newton's method).

While we've worked over the interval $[-1.1]$ so far, Gaussian quadrature can equally well be applied to arbitrary intervals. To do this, we transform a given integral over $[a, b]$ into an integral over $[-1, 1]$ by making the substitution $u = \frac{2x-a-b}{b-a}$ (and $x = \frac{1}{2}((b-a)u + a + b)$, so $dx = \frac{b-a}{2}$).

$$\int_{a}^{b} f(x)dx = \int_{-1}^{1} f(\frac{(b-a)t + b + a}{2})\frac{b-a}{2}dt.$$