Numerical integration for AMSC 460
Lecture 1

Prof. Jacob Bedrossian
University of Maryland, College Park

These notes will supplement the lectures on numerical integration. Doron Levy's notes on the topic are also good and closely approximating the lectures.

Consider the problem of attempting to numerically integrate a given function f:

$$\int_0^1 f(x)dx = ? \tag{1}$$

One of the easiest ways to think about most numerical integration methods is to imagine that we are going to build an approximation to f, say $f_n(x)$ for some large n, and then exactly integrate f_n. Let $h = 1/n$ and $x_j = hj$ (in numerical integration these are often called "quadrature points" and numerical integration formulas are sometimes called "quadrature"). Consider the left piecewise linear approximation:

$$f_n(x) = f(x_j) \quad x \in [x_j, x_{j+1}]. \tag{2}$$

One can (and should) check that $\|f_n - f\|_{L^\infty} = O(h)$. The exact integral of this approximation is the *left* Riemann sum:

$$\int_0^1 f_n(x)dx = \sum_{j=0}^{n-1} \int_{x_j}^{x_{j+1}} f(x_j)dx = \frac{1}{n}\sum_{j=0}^{n-1} f(x_j). \tag{3}$$

Next, we want to evaulate the difference between our approximation and our the true integral. Write

$$\left|\int_0^1 f_n(x)dx - \int_0^1 f(x)dx\right| = \left|\sum_{j=0}^{n-1} \int_{x_j}^{x_{j+1}} f(x_j) - f(x)dx\right|. \tag{4}$$

By the mean-value theorem, for all x, x_j, there exists c (between x and x_j) such that

$$f(x) = f(x_j) + f'(c)(x - x_j). \tag{5}$$

Let

$$\max_{z\in[0,1]} \left|f'(z)\right| = M. \tag{6}$$

Then, we have (recall that $\left|\int_a^b g(x)dx\right| \leq \int_a^b |g(x)|\, dx$):

$$\begin{aligned}
\left|\int_0^1 f_n(x)dx - \int_0^1 f(x)dx\right| &= \left|\sum_{j=0}^{n-1} \int_{x_j}^{x_{j+1}} f'(c)(x - x_j)dx\right| && (7)\\
&\leq \sum_{j=0}^{n-1} M \int_{x_j}^{x_{j+1}} |x - x_j|\, dx && (8)\\
&\leq \sum_{j=0}^{n-1} Mh^2 && (9)\\
&= O(h). && (10)
\end{aligned}$$

We say that the numerical integration scheme is *first order* (since $O(h) = O(h^1)$). It is very natural to expect that a more accurate approximation f_n will result in a better integration formula, and this is true. However, the situation is in fact even a little more sutble than that. Consider the *midpoint* piecewise linear approximation:

$$f_n(x) = f(x_j + \frac{h}{2}) \quad x \in [x_j, x_{j+1}]. \tag{11}$$

This approximation still satisfies $\|f_n - f\|_{L^\infty} = O(h)$. The exact integral of this approximation is the *midpoint* Riemann sum:

$$\int_0^1 f_n(x)dx = \sum_{j=0}^{n-1} \int_{x_j}^{x_{j+1}} f(x_j + \frac{h}{2})dx = \frac{1}{n} \sum_{j=0}^{n-1} f(x_j + \frac{h}{2}). \tag{12}$$

Next,

$$\left| \int_0^1 f_n(x)dx - \int_0^1 f(x)dx \right| = \left| \sum_{j=0}^{n-1} \int_{x_j}^{x_{j+1}} f(x_j + \frac{h}{2}) - f(x)dx \right|. \tag{13}$$

By Taylor's theorem, for all x, x_j, there exists c (between x and x_j) such that

$$f(x) = f(x_j + \frac{h}{2}) + f'(x_j + \frac{h}{2})(x - x_j - \frac{h}{2}) + \frac{1}{2} f''(c)(x - x_j - \frac{h}{2})^2, \tag{14}$$

and hence

$$\left| \int_0^1 f_n(x)dx - \int_0^1 f(x)dx \right| = \left| \sum_{j=0}^{n-1} \int_{x_j}^{x_{j+1}} f'(x_j + \frac{h}{2})(x - x_j - \frac{h}{2}) + \frac{1}{2} f''(c)(x - x_j - \frac{h}{2})^2 dx \right|. \tag{15}$$

However, notice the special cancellation:

$$\int_{x_j}^{x_j+h} f'(x_j + \frac{h}{2})(x - x_j - \frac{h}{2})dx = f'(x_j + \frac{h}{2}) \int_{-h/2}^{h/2} xdx = 0. \tag{16}$$

Therefore the error is in fact:

$$\left| \int_0^1 f_n(x)dx - \int_0^1 f(x)dx \right| = \left| \sum_{j=0}^{n-1} \int_{x_j}^{x_{j+1}} \frac{1}{2} f''(c)(x - x_j - \frac{h}{2})^2 dx \right|. \tag{17}$$

Let M be such that

$$\max_{z \in [0,1]} |f''(z)| = M. \tag{18}$$

Then,

$$\begin{aligned} \left| \int_0^1 f_n(x)dx - \int_0^1 f(x)dx \right| &\le \frac{M}{2} \sum_{j=0}^{n-1} \int_{x_j}^{x_{j+1}} (x - x_j - \frac{h}{2})^2 dx & (19) \\ &= O(h^2). & (20) \end{aligned}$$

Hence, the midpoint integration rule is *second-order accurate* which is much better. This cancellation is similar to that which occurs when you approximate a derivative by a central difference:

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + O(h^2) \tag{21}$$

vs a forward difference

$$f'(x) = \frac{f(x+h) - f(x)}{h} + O(h). \tag{22}$$

As a general rule *centered* approximations are better than uncentered approximations.

Second order accuracy is pretty good, but for relatively simple problem such as integration of a smooth function, we can do a lot better with only a moderate amount of suffering. The next thing one would try is to approximate f as a piecewise linear function:

$$f_n(x) = f(x_j) + \frac{f(x_{j+1}) - f(x_j)}{h}(x - x_j) \quad x \in [x_j, x_{j+1}]. \tag{23}$$

This yields the integration scheme known as the *trapezoidal rule* (as you will derive on your homework):

$$\int_0^1 f(x)dx \approx \int_0^1 f_n(x)dx = \sum_{j=0}^{n-1} \frac{h}{2}\left(f(x_j) + f(x_{j+1})\right) = \frac{h}{2}f(x_0) + \frac{h}{2}f(x_n) + h\sum_{j=1}^{n-2} f(x_j). \tag{24}$$

This seems almost the same as one of the left or right Riemann sum rules, but actually as you will check on your homework,

$$\left|\int_0^1 f(x)dx - \int_0^1 f_n(x)dx\right| = O(h^2). \tag{25}$$

One can also derive this result from the fact that $\|f_n - f\|_{L^\infty} = O(h^2)$.

We still have not succeeded in obtaining higher order accuracy, so maybe it is time to break out the big guns. On each sub-interval $[x_j, x_{j+1}]$ let us try to use a higher order polynomial interpolation to approximate f_n. Hence, let us choose ν points $x_{j,i}$: $x_{j,0} = x_j < x_{j,1} < ... < x_{j,\nu} = x_{j+1}$ and build the Lagrange interpolation polynomial

$$f_n(x) = \sum_{i=0}^{\nu} f(x_{j,i})\ell_{j,i}(x), \tag{26}$$

where $\ell_{j,i}(x)$ is the Lagrange polynomial: for $p \in \{0, 1, .., \nu\}$,

$$\ell_{j,i}(x_{j,p}) = \begin{cases} 1 & p = i \\ 0 & p \neq i \end{cases}. \tag{27}$$

Then we can integrate this exactly as:

$$\begin{aligned}\int_0^1 f_n(x)dx &= \sum_{j=0}^{n-1} \int_{x_j}^{x_{j+1}} \sum_{i=0}^{\nu} f(x_{j,i})\ell_{j,i}(x)dx \\ &= \sum_{j=0}^{n-1}\sum_{i=0}^{\nu} f(x_{j,i}) \int_{x_j}^{x_{j+1}} \ell_{j,i}(x)dx.\end{aligned}$$

As the Lagrange polynomials are just polynomials, we can pre-compute those integrals by hand and get another quadrature formula. As a general scheme, this looks a little daunting. We generally only need to consider a few special cases. The first case is when we choose the sub-points to be evenly spaced: $x_{j,i} = x_j + \frac{i}{\nu}h$. The resulting formulas are called *Newton-Cotes* formulas. They rapidly get pretty messy to write down the exact formulas (but still not above our pay-grades!), however, let us just consider the case of one intermediate point, which would be the center point $x_{j,1} = x_j + h/2$. In this case, the Lagrange polynomial is the quadratic which goes through the three points $(x_j, f(x_j)), (x_j + \frac{h}{2}, f(x_j + \frac{h}{2}), (x_{j+1}, f(x_{j+1}))$. This gives (after recalling the formulas for the Lagrange polynomial):

$$\begin{aligned}\sum_{i=0}^{2} f(x_{j,i})\ell_{j,i}(x) = {} & f(x_j)\frac{(x_j + \frac{h}{2} - x)(x_{j+1} - x)}{\frac{h}{2}h} + f(x_j + \frac{h}{2})\frac{(x - x_j)(x_{j+1} - x)}{\frac{h^2}{4}} \\ & + f(x_{j+1})\frac{(x - x_j)(x - x_j - \frac{h}{2})}{\frac{h}{2}h}.\end{aligned}$$

The integral over the subinterval is then given by a slightly annoying calculation:

$$\begin{aligned}\int_{x_j}^{x_j+h} \sum_{i=0}^{2} f(x_{j,i})\ell_{j,i}(x)dx &= \int_{x_j}^{x_j+h} f(x_j)\frac{(x_j + \frac{h}{2} - x)(x_{j+1} - x)}{\frac{h}{2}h} + f(x_j + \frac{h}{2})\frac{(x_j + \frac{h}{2} - x)(x_{j+1} - x)}{\frac{h^2}{4}} \\ &\quad + f(x_{j+1})\frac{(x - x_j)(x - x_j - \frac{h}{2})}{\frac{h}{2}h}dx \\ &= f(x_j)\frac{2}{h^2}\int_0^h (\frac{h}{2} - x)(h - x)dx + f(x_j + \frac{h}{2})\frac{4}{h^2}\int_0^h x(h - x)dx \\ &\quad + f(x_{j+1})\frac{2}{h^2}\int_0^h x(x - \frac{h}{2})dx \\ &= f(x_j)\frac{2}{h^2}\int_0^h x^2 - \frac{3}{2}hx + \frac{h^2}{2}dx + f(x_j + \frac{h}{2})\frac{4}{h^2}\int_0^h hx - x^2dx \\ &\quad + f(x_{j+1})\frac{2}{h^2}\int_0^h x^2 - \frac{xh}{2}dx \\ &= f(x_j)\frac{2}{h^2}\left(\frac{h^3}{3} - \frac{3h^3}{4} + \frac{2h^3}{4}\right) + f(x_j + \frac{h}{2})\frac{4}{h^2}\left(\frac{h^3}{2} - \frac{h^3}{3}\right) \\ &\quad + f(x_{j+1})\frac{2}{h^2}\left(\frac{h^3}{3} - \frac{h^3}{4}\right) \\ &= f(x_j)\frac{h}{6} + f(x_j + \frac{h}{2})\frac{2h}{3} + f(x_{j+1})\frac{h}{6}.\end{aligned}$$

This gives the numerical integration scheme known as *Simpson's rule*:

$$\int_0^1 f(x)dx \approx \sum_{j=0}^{n-1}\int_{x_j}^{x_j+h}\sum_{i=0}^{2} f(x_{j,i})\ell_{j,i}(x)dx \tag{28}$$

$$= \sum_{j=0}^{n-1}\frac{h}{6}\left(f(x_j) + 4f(x_j + \frac{h}{2}) + f(x_{j+1})\right). \tag{29}$$

Notice that this scheme is roughly twice the work as the midpoint rule and the trapezoidal rules: we need to evaluate f at double the number of quadrature points. We of course hope that this scheme should be at least twice as accurate as trapezoidal or midpoint for this reason, but it was so annoying to compute that we are really hoping for it be even more accurate. Since we made an approximation which is one order more accurate than the approximation made when we did trapezoidal, it might be reasonable to expect that Simpson's rule is $O(h^3)$. However, (similar to what happened with the midpoint rule, this scheme is actually *fourth order*, which is MUCH faster than trapezoidal and midpoint. This certainly makes it attractive for efficiently doing integration.

The midpoint and Simpson's rule suggests that it is not totally obvious how to relate the number of quadrature points to the accuracy of the integration. Just like we had in interpolation and approximation, it makes sense to wonder how much more we could increase the efficiency by intelligently choosing the quadrature points (and setting intelligent values for the weights)...