Week 11: Wednesday, Apr 6

Logistics

- Prelim 2 is tomorrow, 7:30-9:30 in Upson B17
- Makeup prelim is 8:00-10:00 am in Upson 315
- Practice prelim 2 solutions are posted

Degree of an integration rule

Suppose we write

$$I_h[f] = \int_0^h f(x)\,dx$$
$$Q_h[f] = h \sum_{j=1}^n w_j f(hx_j)$$

We have in mind that the quadrature rule $Q_h[f]$ is supposed to approximate $I_h[f]$. What we want to show now is that we can analyze the quality of that approximation just based on whether or not $Q_h[x^m] = I_h[x^m]$ for small values of m.

Suppose $Q_h[f]$ has degree d; that means that $Q_h[f]$ integrates polynomials of degree $\leq d$ exactly. Using Taylor's theorem with remainder, we can write

$$f(x) = p(x) + \frac{f^{(d+1)}(\xi)}{(d+1)!} x^{d+1},$$

where p is a degree d polynomial (the degree d Taylor approximation). Suppose $|f^{(d+1)}| < M$; then we have

$$|I_h[f-p]| \leq \frac{M_d}{(d+2)!} h^{d+2} = O(h^{d+2})$$

and

$$|Q_h[f-p]| \leq \sum_{j=1}^n |w_j| \frac{M_d}{(d+1)!} h^{d+2} = O(h^{d+2}).$$

Therefore

$$\begin{aligned} |I_h[f] - Q_h[f]| &= |I_h[f-p] - Q_h[p-f]| \\ &\leq |I_h[f-p]| + |Q_h[f-p]| = O(h^{d+2}). \end{aligned}$$

This tells us that the local truncation error (the error per panel) of a degree d integration rule is $O(h^{d+2})$; in a composite rule where there are $O(h^{-1})$ panels, we have a total error of $O(h^{d+1})$.

Raising the degree

An interpolatory quadrature rule through n points has degree $n-1$, and so yields (total) error that decreases at least like $O(h^n)$, assuming that the function in question is sufficiently smooth. In some cases, though, we know that we get lucky and do even better. For example, the midpoint rule ($n = 1$) has degree 2, and Simpson's rule ($n = 3$) has degree 4. Why is this the true?

For convenience, let us consider a quadrature rule on $[-1, 1]$. A quadrature rule with n points has degree $n+s$ for $s \geq 0$, that means it computes any polynomial of degree up to $n + s$ exactly. In particular, if $x_1, \ldots, x_n$ are the nodes, we can define the degree n polynomial $q(x) = (x-x_1)\ldots(x-x_n)$, and our rule should be able to integrate $q(x)x^j$ exactly for $0 \leq j \leq s$. But notice that $q(x)x^j$ is exactly zero at each of the quadrature nodes, so the quadrature rule returns exactly zero at each of these points. Therefore, the quadrature rule can have degree $n + s$ for $s \geq 0$ only if it satisfies the conditions

$$\int_{-1}^{1} q(x)x^j \, dx = 0, \quad 0 \leq j \leq s.$$

This says that with respect to the standard inner product for functions on $[-1, 1]$, the polynomial q should be *orthogonal* to x^j for $0 \leq j \leq s$. Note that we must have $s < n$, since otherwise we would have that $\int_{-1}^{1} q(x)^2 \, dx$ was zero.

As it happens, the *Legendre polynomials* $P_k(x)$ satisfy the property that $P_0(x), \ldots, P_d(x)$ forms an orthonormal basis (with respect to the standard inner product on $[-1, 1]$) for the degree d polynomials. The first few Legendre polynomials are

$$\begin{aligned} P_0(x) &= 1 \\ P_1(x) &= x \\ P_2(x) &= (3x^2 - 1)/2, \end{aligned}$$

and we can compute higher-order Legendre polynomials by a recurrence:

$$(k+1)P_{k+1}(x) = (2k+1)xP_k(x) - kP_{k-1}(x).$$

Interpolatory quadrature rules based on interpolation through the zeros of Legendre polynomials are *Gauss-Legendre* quadrature rules. The midpoint rule is the lowest-order such rule; the second rule is

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

In general, n-point Gauss-Legendre quadrature rules have degree $2n-1$; the two-point Gauss-Legendre rule has degree 3, for example.

There are a few variants on the Gaussian integration theme. One involves constraining the nodes for computational convenience. For example, if we insist that the interval endpoints must be quadrature nodes, we arrive at the Gauss-Lobatto rules (degree $2n-3$). The Gauss-Kronrod rules involve a pair consisting of an n-point Gauss quadrature rule together with a $2n+1$ point rule that re-uses the Gauss quadrature nodes; these rules are popular for adaptive quadrature, since the Gauss rule and the Kronrod rule can be compared in order to get an error estimate.

Special behaviors

As is often the case, MATLAB has a reasonable library of quadrature rules, and they work terrifically well for nice, smooth integrands on bounded domains. But for some problems, you have to think a little about how to set up the integral if you want the computation to be efficient and accurate. Examples of problematic features include:

- Non-smooth integrands (often piecewise smooth)
- Singular integrands

- Unbounded domains
- Oscillatory integrands

Techniques for dealing with these features include:

- Breaking up the domain of integration
- Subtracting off singularities
- Change of variables
- Integration by parts
- Weighted quadrature rules

We will discuss some examples in the next lecture.