Week 10: Monday, Apr 2

Hermite interpolation

For standard polynomial interpolation problems, we seek to satisfy conditions of the form

$$p(x_j) = y_j,$$

where y_j is frequently a sampled function value $f(x_j)$. If all we know is function values, this is a reasonable approach. But sometimes we have more information. *Hermite* interpolation constructs an interpolant based not only on equations for the function values, but also for the derivatives.

For example, consider the important special case of finding a cubic polynomial that satisfies proscribed conditions on the values and derivatives at the endpoints of the interval $[-1, 1]$. That is, we require

$$\begin{aligned} p(1) &= f(1) & \qquad p(-1) &= f(-1) \\ p'(1) &= f'(1) & \qquad p'(-1) &= f'(-1). \end{aligned}$$

As with polynomial interpolation based just on function values, we can express the cubic that satisfies these conditions with respect to several different bases: monomial, Lagrange, or Newton.

For the monomial basis, we have

$$p(x) = c_0 + c_1 x + c_2 x^2 + c_3 x^3,$$

which yields the linear system

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 0 & 1 & 2 & 3 \\ 0 & 1 & -2 & 3 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} f(1) \\ f(-1) \\ f'(1) \\ f'(-1) \end{bmatrix}.$$

For the Newton basis, we have expressions like

$$p(x) = f[-1]+f[-1,-1](x+1)+f[1,-1,-1](x+1)^2+f[1,1,-1,-1](x+1)^2(x-1),$$

where the divided differences for f are

$$\begin{aligned}
f[1] &= f(1) \\
f[-1] &= f(-1) \\
f[1,1] &= f'(1) \\
f[-1,-1] &= f'(-1) \\
f[1,-1] &= \left(f(1) - f(-1)\right)/2 \\
f[1,-1,-1] &= \left(f[1,-1] - f[-1,-1]\right)/2 \\
f[1,1,-1] &= \left(f[1,1] - f[1,-1]\right)/2 \\
f[1,1,-1,-1] &= \left(f[1,1,-1] - f[1,-1,-1]\right)/2.
\end{aligned}$$

We leave the Lagrange basis as a problem to ponder (or look up).

Piecewise polynomial approximations

Polynomials are convenient for interpolation for a few reasons: we know how to manipulate them symbolically, we can evaluate them quickly, and there is a theorem of analysis (the Weierstrass approximation theorem) that says that any continuous function on some interval $[a,b]$ can be uniformly approximated by polynomials. In practice, though, high-degree polynomial interpolation does not always provide fantastic function approximation. An alternative approach that retains the advantages of working with polynomials is to work with *piecewise* polynomial functions.

Piecewise linear interpolation

Perhaps the simplest example is piecewise linear interpolation; if function values $f(x_j)$ are given at points $x_1 < x_2 < x_3 < \ldots < x_n$, then we write the approximating function $\hat{f}(x)$ as

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

Alternately, we can write

$$\hat{f}(x) = \sum_{j=1}^{n} \phi_j(x) f(x_j)$$

where $\phi_j(x)$ is a "hat function":

$$\phi_j(x) = \begin{cases} (x - x_{j+1})/(x_j - x_{j+1}), & x \in [x_j, x_{j+1}], \\ (x - x_{j-1})/(x_j - x_{j-1}), & x \in [x_{j-1}, x_j], \\ 0 & \text{otherwise.} \end{cases}$$

This last you may recognize as similar in spirit to using a basis of Lagrange polynomials for polynomial interpolation.

Using piecewise linear interpolation to approximate a function f yields $O(h^2)$ error (where h is the distance between interpolation points), assuming f has two continuous derivatives. This level of accuracy is adequate for many purposes. Beyond the basic error behavior, though, piecewise linear interpolation has several virtues when *structural* properties are important. For example, the maximum and minimum values of a piecewise linear interpolant are equal to the maximum and minimum values of the data. And if f is positive or monotone (like a probability density or cumulative density function), then any piecewise linear interpolant inherits these properties.

Piecewise cubic interpolation

If f is reasonably smooth and the data points are widely spaced, it may make sense to use higher-order polynomials. For example, we might decide to use a *cubic spline* $\hat{f}(x)$ characterized by the properties:

- Interpolation: $\hat{f}(x_i) = f(x_i)$
- Twice differentiability: $\hat{f}'$ and $\hat{f}''$ are continuous at $\{x_2, \ldots, x_{n-1}\}$

The interpolation and differentiability constraints give us $4n - 2$ constraints on the $4n$-dimensional space of piecewise polynomial functions that are defined by general cubics on each interval $[x_j, x_{j+1}]$. In order to uniquely determine the spline, we need some additional constraint; common choices are

- Specified values of f' at x_1 and x_n (clamped conditions)
- A natural spline: $f''(x_1) = f''(x_n) = 0$
- Not-a-knot conditions: f''' is continuous at x_2 and x_{n-1}
- Periodicity: $f'(x_1) = f'(x_n), f''(x_1) = f''(x_n)$

For the clamped conditions and not-a-knot conditions, one has the error bound

$$\|p - f\|_\infty \leq c\|f'''\|_\infty h^4$$

where $\|\cdot\|_\infty$ is the L^∞ norm (max norm) on some interval of interest and h is the maximum space between interpolation nodes.

In addition to spline conditions, one can choose piecewise cubic polynomials that satisfy Hermite interpolation conditions (sometimes referred to by the acronym PCHIP or Piecewise Cubic Hermite Interpolating Polynomials). That is, the function values and derivatives are specified at each nodal point. If we don't actually have derivative values prescribed at the nodal points, then we can assign these values to satisfy additional constraints. We gain this flexibility at the cost of some differentiability; piecewise cubic Hermite interpolants are in general not twice continuously differentiable.

As in the case of polynomial interpolation, there are several different bases for the space of piecewise cubic functions. Any choice of locally supported basis functions (basis functions that are only nonzero on only a fixed number of intervals $[x_j, x_{j+1}]$) leads to a banded linear system which can be solved in $O(n)$ time to find either cubic splines or piecewise Hermite cubic interpolants. One common choice of basis is the B-spline basis, which you can find described in the book.