Let's be rational about this

Our go-to form of interpolating function is a polynomial. When used globally for interpolation in a stable algorithm such as the barycentric formula, polynomials converge to smooth functions if the nodes are distributed in the manner of Chebyshev points. However, smoothness is a relative property. The function $\tanh(10^6 x)$, for example, is infinitely differentiable everywhere, but it has a derivative at the origin so large that it looks like a jump unless you sample very finely there.

The barycentric formula for p is

$$p(x) = \frac{\displaystyle\sum_{j=0}^{n} \frac{w_j y_j}{x - t_j}}{\displaystyle\sum_{j=0}^{n} \frac{w_j}{x - t_j}}, \tag{1}$$

where the (t_j, y_j) are interpolation points and the w_j are the barycentric weights. We saw that for a certain choice of weights based on the node locations, p is in fact a polynomial. But the formula guarantees the interpolation conditions for any selection of the weights, and in general, p is a **rational function**—that is, the ratio of two polynomials of degree n or less.

Because they can have real or complex **poles**, which are zeros in the denominator, rational functions may be superior to polynomials for approximating functions that have steep gradients. Moreover, they are not subject to the Runge phenomenon that dictates the stable distribution of nodes for polynomials.

Suppose we want to construct a rational interpolant to $f(x)$ at nodes t_j given in an interval $[a, b]$. In order to help us find a sensible choice for the barycentric weights, we choose some test points $s_1, \dots, s_m$ in the interval. We would like to impose that $f(s_i) \approx p(s_i)$ at all i. Applying (1), this rearranges into

$$f(s_i) \cdot \sum_{j=0}^{n} \frac{w_j}{s_i - t_j} \approx \sum_{j=0}^{n} \frac{w_j y_j}{s_i - t_j}, \qquad i = 1, \dots, m, \tag{2}$$

where the $y_j = f(t_j)$ are known. We now define the $m \times n$ **Cauchy matrix**

$$C_{ij} = \frac{1}{s_i - t_j}, \qquad i = 1, \dots, m, \quad j = 0, \dots, n. \tag{3}$$

Then (3) becomes

$$\boldsymbol{F}\boldsymbol{C}\boldsymbol{w} \approx \boldsymbol{C}\boldsymbol{Y}\boldsymbol{w} \quad \Rightarrow \quad (\boldsymbol{F}\boldsymbol{C} - \boldsymbol{C}\boldsymbol{Y})\boldsymbol{w} \approx \boldsymbol{0}, \tag{4}$$

where $\boldsymbol{F} = \operatorname{diag}(f(s_1), \dots, f(s_m))$ and $\boldsymbol{Y} = \operatorname{diag}(y_0, \dots, y_n)$.

We could interpret equation (4) as trying to minimize a norm of the vector $(\boldsymbol{F}\boldsymbol{C} - \boldsymbol{C}\boldsymbol{Y})\boldsymbol{w}$. That does not seem like a well-posed problem, because we can just multiply any choice of $\boldsymbol{w}$ by a tiny scalar to make this norm get smaller. But a close look at (1) shows that a uniform scaling of the weights actually has no effect on p. Therefore, we may add a normalization constraint and get the problem

$$\boldsymbol{w} = \operatorname{argmin}_{\|\boldsymbol{v}\|_2 = 1} \|\boldsymbol{A}\boldsymbol{v}\|_2, \qquad \boldsymbol{A} = \boldsymbol{F}\boldsymbol{C} - \boldsymbol{C}\boldsymbol{Y}. \tag{5}$$

We have reached the standard problem of finding the least-significant (i.e., associated with the smallest singular value) right singular vector of $\boldsymbol{A}$. From there we get everything needed to evaluate p in (1) at any value of x.

Goals

You will write a procedure to create rational interpolants and explore it on a few functions.

Procedure

1. Write a function `ratinterp(f, t)` that returns a rational interpolant function given $f(x)$ and the nodes t_j. You can assume that the interval of approximation is $[t_0, t_n]$ (the first and last elements of `t`). You should select $m = 5n + 2$ evenly spaced test points in the interval.

   Use `polyinterp` from the book as a guide. Due to the definition (3), you don't want the test points to coincide with any nodes; just delete any such test points and it should be OK.

2. Define the function $f(x) = \tanh(10x) + 2x^2$ and plot it at 1000 points over the interval $[-1, 1]$.

3. Add to your plot the polynomial interpolant using equally spaced nodes in $[1, 1]$ for $n = 18$, i.e., there are 19 nodes including the endpoints. (It will not be a great result.)

4. Find the rational interpolant r with the same nodes as in the last step. Start a new plot for the error $f(x) - r(x)$ over $[-1, 1]$. It should be very small over the whole interval.

5. Repeat the previous step for $g(x) = \tanh(100x) + 2x^2$.

6. Find an arrangement of 19 nodes that produces a rational interpolant for g whose error is less than 10^{-6} in magnitude everywhere in the whole interval.