7. Piecewise cubic polynomial interpolation. Present results on piecewise cubic interpolation. Use either `pchiptx` or `splinetx` to discuss implementation of the algorithms.

Numerical Analysis E2021

Institute of Mathematics
Aalborg University

AALBORG UNIVERSITY
DENMARK

Lagrange Interpolation Theorem
Let $\mathbb{F}$ be a field, and let $m \in \mathbb{N}$. If $\alpha_1, \dots, \alpha_m \in \mathbb{F}$ are all distinct, then the evaluation map $\mathcal{E}\colon \mathbb{F}[X]_{\leq m-1} \to \mathbb{F}^m$, such that $f(X) \mapsto (f(\alpha_1), \dots, f(\alpha_m))$, is an isomorphism.

Proof.
$\mathcal{E}$ is a linear map, and the two vector spaces has the same dimension. Thus, it suffices to show that it is injective. This fact immediately follows by the fundamental theorem of algebra. $\square$

However, Lagrange interpolation often has issues due to high-degree interpolation often yielding oscillatory polynomials. See MATLAB demo of Runge. How do we circumvent this?

Cubic Polynomial Interpolation

Many of the most effective interpolation techniques are based on piecewise cubic polynomials. For this we use Hermite polynomials, speficially the following result

Proposition

There exists a polynomial p of at most degree 3 satisfying

$$p(x_j) = y_j \quad \text{and} \quad p'(x_j) = z_j, \quad j = 0, 1 \tag{1}$$

and p is uniquely determined by these equations.

However, we have a problem if we do not have the first derivative in our data set. Two methods for generating these derivatives are with `pchip` or `spline`.

Piecewise cubic Hermite interpolating polynomial

The idea is to determine the slopes such that the function does not overshoot the data locally. The implementation is done in the following sense

we define $\delta_k := \dfrac{y_{k+1} - y_k}{x_{k+1} - x_k}$, and denote our derivative as $d_k := p'(x_k)$.

- **δ_k and δ_{k-1} have opposite signs, or one of them is zero:**
  x_k is a discrete local maximum or minimum, so set $d_k := 0$.
- **δ_k and δ_{k-1} have the same sign, and the two intervals have the same length:**
  We set d_k as the harmonic mean between the two discrete slopes, so

$$\frac{1}{d_k} := \frac{1}{2}\left(\frac{1}{\delta_{k-1}} + \frac{1}{\delta_k}\right) \tag{2}$$

- **δ_k and δ_{k-1} have the same sign, but the two intervalls have differing lengths:**
  We set d_k as a weighted harmonic mean, where the weights depend on the lengths of the intervals, so

$$\frac{w_1 + w_2}{d_k} := \frac{w_1}{\delta_{k-1}} + \frac{w_2}{\delta_k}, \quad w_1 + 2h_k + h_{k-1}, w_2 = h_k + 2h_{k-1} \tag{3}$$

Summary

We have several methods for polynomial interpolation:

- Piecewise linear interpolation
  - Hardly any smoothness
  - Preserves local monotonicity
- Lagrange interpolation
  - Infinitely differentiable
  - Often fails to preserve shape, particularly near ends, as seen with Runge
- pchip
  - Guaranteed to preserve shape
  - Discontinuous curvature due to non-continuous second-order derivaties
- spline
  - Has continuous second-order derivatives
  - Not guaranteed to preserve shape

MATLAB demo of `interpgui`