Overview

- Polynomial interpolant forms
  - Monomial basis
  - Newton basis
- Interpolation accuracy
- Inexact data

- Recall: $f\colon \mathbb{R} \to \mathbb{R}$, "difficult" to compute, and known through a sample

$$\mathcal{D} = \{(x_i, y_i), i = 0, 1, 2, \ldots, m\}, y_i = f(x_i), i \neq j \Rightarrow x_i \neq x_j.$$

- Approximation built from linear combination of $\{g_0(t), g_1(t), \ldots, g_n(t)\}$

$$g(t) = c_0\, g_0(t) + c_1\, g_1(t) + \cdots + c_n\, g_n(t)$$

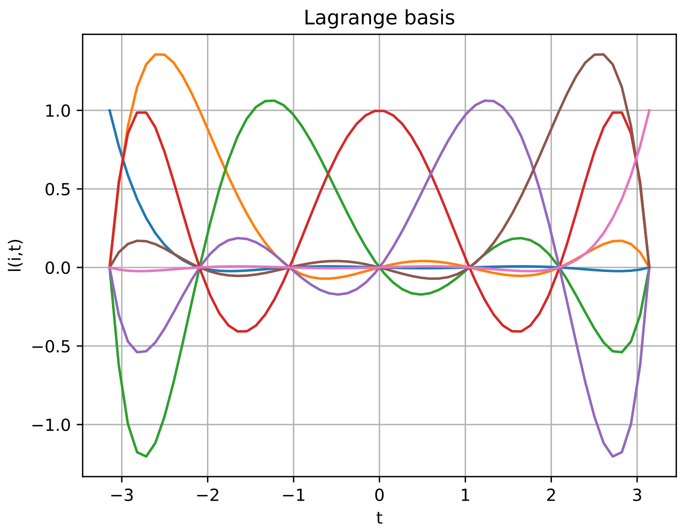
- Lagrange basis: $l_i(t) = \prod_{j=0}^{m}{}' (t - x_j)$ or $\ell_i(t) = \frac{\prod_{j=0}^{m}{}' (t - x_j)}{\prod_{j=0}^{m}{}' (x_i - x_j)}$
- Monomial basis $\{1, t, t^2, \ldots\}$

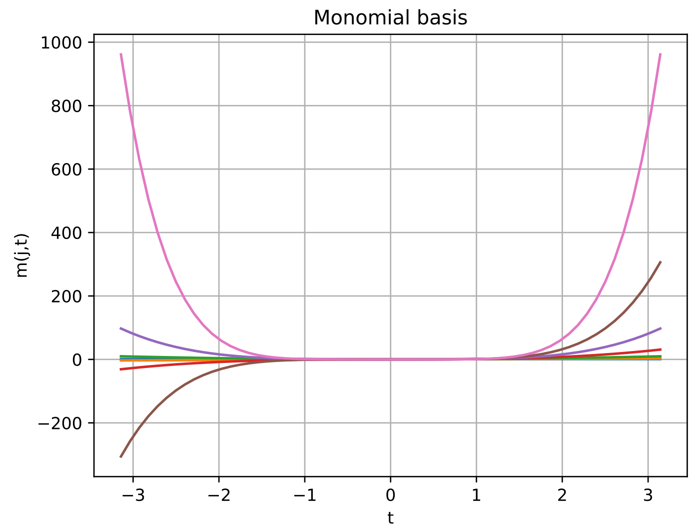
$$p(t) = c_0 \cdot 1 + c_1\, t + \cdots + c_n\, t^n$$

- Newton basis $\{n_0(t), \ldots, n_n(t)\} = \{1, t - x_0, (t - x_0)(t - x_1), \ldots\}$

$$p(t) = c_0 \cdot 1 + c_1\,(t - x_0) + \cdots + c_n\,(t - x_0)(t - x_1)\ldots(t - x_{n-1})$$

Monomial basis compared to Lagrange

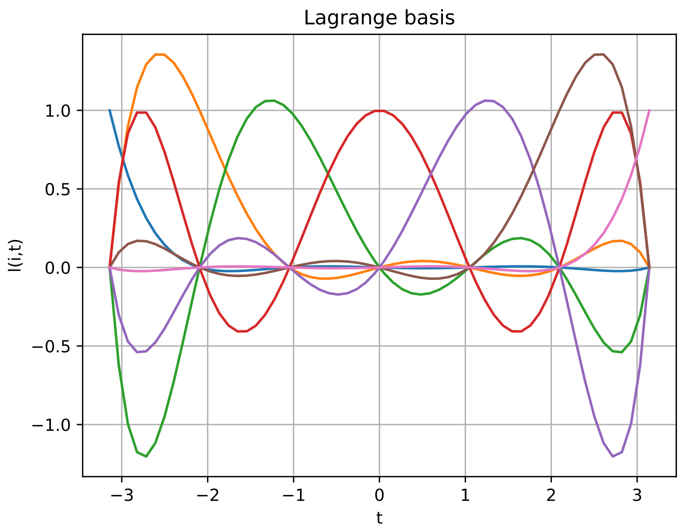


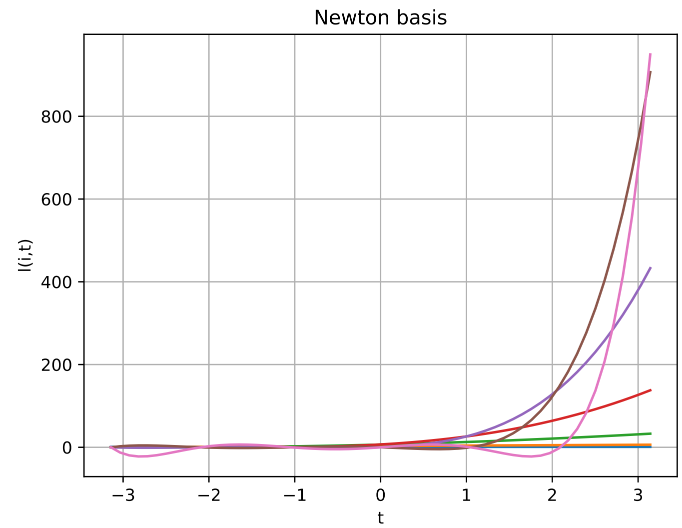


- Monomial basis functions are almost indistinguishable over portions of the function domain, closer to linearly dependent than the Lagrange basis functions
- Intuitive analogy from linear algebra: $\boldsymbol{b} \in \mathbb{R}^m$ can be expressed either in an orthogonal basis $\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_m\}$ or a non-orthogonal basis $\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_m\}$

$$\boldsymbol{b} = b_1\boldsymbol{e}_1 + \cdots + b_m\boldsymbol{e}_m = x_1\boldsymbol{a}_1 + \cdots + x_m\boldsymbol{a}_m \Leftrightarrow \boldsymbol{b} = \boldsymbol{A}\boldsymbol{x}$$

When $\boldsymbol{A}$ is close to singular, small errors in $\boldsymbol{b}$ lead to large errors in $\boldsymbol{x}$. Orthogonal bases are preferable.





- Behavior similar to monomial basis: almost indistinguishable over portions of the domain.
- Ideally the basis functions $\{g_0(t), \ldots, g_n(t)\}$ would be *orthonormal* with respect to a scalar product with weight $w(t)$

$$(g_i, g_j) = \int_a^b w(t)\, g_i(t)\, g_j(t)\, \mathrm{d}t = \delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

Compare with vector scalar product $\boldsymbol{u}^T \boldsymbol{v} = u_1 v_1 + \cdots + u_m v_m$.

- Interpolation conditions $p(x_i) = c_0 + c_1 x_i + \cdots + c_n x_i^n = f(x_i) = y_i$ lead to

$$p(t) = \left[\begin{array}{cccc} 1 & t & \ldots & t^n \end{array}\right] \left[\begin{array}{c} c_0 \\ c_1 \\ \vdots \\ c_n \end{array}\right], \left[\begin{array}{cccc} 1 & x_0 & \ldots & x_0^n \\ 1 & x_1 & \ldots & x_1^n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \ldots & x_n^n \end{array}\right] \left[\begin{array}{c} c_0 \\ c_1 \\ \vdots \\ c_n \end{array}\right] = \left[\begin{array}{c} y_0 \\ y_1 \\ \vdots \\ y_n \end{array}\right] \Leftrightarrow \boldsymbol{X}\boldsymbol{c} = \boldsymbol{y}.$$

- The matrix $\boldsymbol{X}$ is known as a Vandermonde matrix and though non-singular for distinct sample nodes $(i \neq j \Rightarrow x_i \neq x_j)$, can be become close to singular.
- For given distinct data, the interpolating polynomial is unique
- Coefficients of the monomial basis require solving the linear system, $\boldsymbol{X}\boldsymbol{c} = \boldsymbol{y}$, at $\mathcal{O}(n^3/3)$ FLOPs, more expensive than the $\mathcal{O}(n^2)$ for Lagrange form
- Notes:
  - Though $p(t) = c_0 \cdot 1 + c_1 t + \cdots + c_n t^n$ is the most often encountered form of a polynomial in analytical mathematics, other forms are more useful in numerical approximation
  - Monomial, Lagrange are different forms of the unique interpolating polynomial

- Interpolation conditions lead to a triangular system

$$p(t) = \begin{bmatrix} n_0(t) & n_1(t) & \dots & n_n(t) \end{bmatrix} \begin{bmatrix} d_0 \\ d_1 \\ \vdots \\ d_n \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & x_1 - x_0 & \cdots & 0 \\ 1 & x_2 - x_0 & \cdots & 0 \\ 1 & x_3 - x_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n - x_0 & \cdots & \prod_{j=0}^{n-1} (x_n - x_j) \end{bmatrix} \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

- Solving the above system now requires $\mathcal{O}(n^2/2)$
- The Newton interpolating polynomial arises in finite difference calculus.

- The first few coefficients are

$$d_0 = y_0,\ d_1 = \frac{y_1 - d_0}{x_1 - x_0} = \frac{y_1 - y_0}{x_1 - x_0},$$

$$d_2 = \frac{y_2 - (x_2 - x_0)d_1 - d_0}{(x_2 - x_0)(x_2 - x_1)} = \frac{\dfrac{y_2 - y_1}{x_2 - x_1} - \dfrac{y_1 - y_0}{x_1 - x_0}}{x_2 - x_0}.$$

- Introduce divided differences: $[y_i] = y_i$,

$$[y_{i+1}, y_i] = \frac{[y_{i+1}] - [y_i]}{x_{i+1} - x_i} = \frac{y_{i+1} - y_i}{x_{i+1} - x_i}, [y_{i+2}, y_{i+1}, y_i] = \frac{[y_{i+2}, y_{i+1}] - [y_{i+1}, y_i]}{x_{i+2} - x_i}$$

$$[y_{i+k}, y_{i+k-1}, \ldots, y_i] = \frac{[y_{i+k}, y_{i+k-1}, \ldots, y_{i+1}] - [y_{i+k-1}, y_{i+k-1}, \ldots, y_i]}{x_{i+k} - x_i}$$

- Obtain

$$p(t) = [y_0] \cdot 1 + [y_1, y_0] \cdot (t - x_0) + \cdots + [y_n, \ldots, y_0] \cdot (t - x_0) \cdot \ldots \cdot (t - x_{n-1}).$$

- Polynomial interpolant has no error at sampling nodes, $p(x_i) = f(x_i)$
- What about other points. Introduce error function $e(t) = f(t) - p(t)$
- Assume $f \in C^{\infty}(\mathbb{R})$ (smooth). The error is the reminiscent of Taylor series remainder

$$f(t) - p(t) = \frac{f^{(n+1)}(\xi_t)}{(n+1)!} \prod_{i=0}^{n} (t - x_i) = \frac{f^{(n+1)}(\xi_t)}{(n+1)!} w(t).$$

- Above obtained by repeated application of Rolle's theorem to the function

$$\Phi(u) = f(u) - p(u) - \frac{f(t) - p(t)}{w(t)} w(u),$$

- $\Phi(u)$ $n+2$ has roots at $t, x_0, x_1, \ldots, x_n$, hence its $(n+1)$-order derivative must have a root in the interval (x_0, x_n), denoted by ξ_t

$$\Phi^{(n+1)}(\xi_t) = \frac{\mathrm{d}^{n+1}\Phi}{\mathrm{d}u^{n+1}}(\xi_t) = 0 = f^{(n+1)}(\xi_t) - \frac{f(t) - p(t)}{w(t)} (n+1)!$$

- Idea: choose x_i to minimize error. This leads to Chebyshev basis, Lessons 9,10.

- If $y_i = f(x_i)$ are not known exactly, replace interpolation $p(x_i) = y_i$ by

$$g(x_i) \cong y_i,\, g(x_i) = \hat{y}_i \cong y_i$$

- Define *Lebesgue function* to expresses deviation from known data,

$$\lambda(t) = \sum_{i=0}^{n} |\ell_i(t)|$$

- Define the worst case by the *Lebesgue constant*

$$\Lambda = \max_{a \leqslant t \leqslant b} \lambda(t)$$

- The distance between:
  1. the interpolant $p(t)$, $p(x_i) = y_i$, and
  2. another approximating polynomial $g(t)$, $g(x_i) = \hat{y}_i$,

  is bounded by the errors in the data $|y_i - \hat{y}_i| \leqslant \delta$ and the Lebesgue constant

$$\|p - g\|_\infty = \max_{a \leqslant t \leqslant b} |p(t) - g(t)| \leqslant \Lambda\delta\,.$$

- The Lebesgue constant depends on the chosen sample nodes.