

LINMA2171 Numerical Analysis

SIMON DESMIDT

Academic year 2024-2025 - Q1



Table des matières

1		oduction	2
	1.1	General Framework	2
2	Poly	Polynomials	
	2.1	Lagrange interpolation	3
	2.2	Hermite interpolation	3
		Neville's algorithm	
	2.4	Newton's interpolation formula	4

Introduction

1.1 General Framework

- Data:
 - $\chi \subseteq \mathbb{R}^d$ (here, d = 1 often).
 - $f: \chi \to \mathbb{R}$, with $f \in \mathfrak{f}$.
- Design:
 - $\hat{\mathfrak{f}} \subseteq \mathbb{R}^{\hat{\chi}}$ is the set of admissible function, and is a subset of all function from $\hat{\chi}$ to \mathbb{R} .
 - $\mathcal{L}:\hat{\mathfrak{f}}\times\mathfrak{f}\to\mathbb{R}$ is the loss function.
 - $\mathcal{R}:\hat{\mathfrak{f}}\to\mathbb{R}$ is the regularizer.
- Optimisation problem :

$$\arg_{\hat{f} \in \hat{f}} \min \quad \mathcal{L}(\hat{f}, f) + \lambda \mathcal{R}(\hat{f})$$

— Optimisation algorithm.

Polynomials

 \mathcal{P}_n is the set of all real polynomials of degree at most n.

— The Runge phenomenon is the explosion of the polynomial near the boundary of the domain when the interpolation points are chosen to be equidistant. A solution to that is to put more points near the boundary and less in the middle of the domain, e.g. Chebyshev points.

2.1 Lagrange interpolation

Let $x_0, ..., x_n$ be distinct real numbers. The Lagrange polynomial L_k of degree n is such that it is equal to 0 for all x_i , $i \neq k$ and 1 for x_k . This serves as a base for the next interpolations. The general formula for the Lagrange polynomial is

$$L_k(x) = \prod_{i=0}^n \frac{x - x_i}{x_k - x_i} \qquad k = 0, 1, \dots, n$$
 (2.1)

— N.B.: we usually denote $L_k(x; x_0, ..., x_n)$ or let $\chi = (x_0, ..., x_n)$ and $L_k(x; \chi)$.

2.2 Hermite interpolation

Let $x_0, ..., x_n$ be distinct real numbers. Then, given two sets of real numbers $(y_0, ..., y_n)$ and $(z_0, ..., z_n)$, there is a unique polynomial $p_{2n+1} \in \mathcal{P}_{2n+1}$ such that

$$p_{2n+1}(x_i) = y_i$$
 $p'_{2n+1}(x_i) = z_i$ $i = 0, ..., n$ (2.2)

The polynomial p_{2n+1} is termed the Hermite interpolation polynomial of degree at most 2n + 1 for the data points $(x_0, y_0, z_0), \ldots, (x_n, y_n, z_n)$. The expression is

$$p_{2n+1}(x) = \sum_{k=0}^{n} (H_k(x)y_k + K_k(x)z_k) \qquad \begin{cases} H_k(x) = (L_k(x))^2 (1 - 2L'_k(x_k)(x - x_k)) \\ K_k(x) = (L_k(x))^2 (x - x_k) \end{cases}$$
(2.3)

where $L_k(x)$ is the Lagrange polynomial.

— The $H_k(x)$ are such that their derivative is zero for all x_i , and their value is zero for all x_i except x_k , where it is 1.

$$H_k(x_i) = \delta_{ik}$$
 $H'_k(x_i) = 0$ $\forall i$

— The $K_k(x)$ are such that their derivative is zero for all x_i except x_k where it is one, and their value is zero for all x_i .

$$K_k(x_i) = 0$$
 $K'_k(x_i) = \delta_{ik}$ $\forall i$

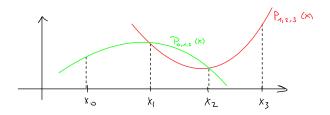
2.3 Neville's algorithm

Let us assume we are given a set of support points (x_i, y_i) , i = 0, 1, ..., n, and p_n is their Lagrange interpolation polynomial. Let us now define the notation $P_{i_0i_1...i_k} \in \mathcal{P}_k$, the polynomial for which $P_{i_0i_1...i_k}(x_{i_j} = y_{i_j} \text{ for all } j = 0, 1, ..., k$. We work by recursion, with the following formula :

$$\begin{cases}
P_i(x) = y_i \\
P_{i_0 i_1 \dots i_k} = \frac{(x - x_{i_0}) P_{i_1 i_2 \dots i_k}(x) - (x - x_{i_k}) P_{i_0 i_1 \dots i_{k-1}}(x)}{x_{i_k} - x_{i_0}}
\end{cases}$$
(2.4)

Example:

Let us have four points $(x_0, y_0), \dots (x_3, y_3)$. We want the polynomial interpolating all of them, using Neville's algorithm.



Here,

$$P_{0123}(x) = \frac{x - x_0}{x_3 - x_0} P_{123}(x) + \frac{x_3 - x}{x_3 - x_0} P_{012}(x)$$
 (2.5)

2.4 Newton's interpolation formula

Newton's interpolation formula is used to evaluate polynomials with a computer, as it only needs to compute each operation $(x - x_i)$ one time. We write it like :

$$p_n(x) = ((\dots(y_{0\dots n}(x-x_n) + y_{0\dots n-1})(x-x_{n-1}) + y_{0\dots n-2})(x-x_{n-2}) + \dots) + y_0$$
(2.6)

And the recursive formula is

$$P_{i_0 i_1 \dots i_k} = P_{i_0 i_1 \dots i_{k-1}}(x) + y_{i_0 i_1 \dots i_k}(x - x_{i_0})(x - x_{i_1}) \dots (x - x_{i_{k-1}})$$
(2.7)