Mohamed Boudiaf University - M'sila
Faculty of Technology
Department of Civil Engineering

Numerical Methods Course

Chapter 2: Polynomial Interpolation

Merini Abdelaziz

February 16, 2026

Contents

1 Introduction **2**
1.1 Position of the Interpolation Problem 2
1.2 Existence and Uniqueness 2
1.3 Some Simple Examples (Direct Method) 3

2 Lagrange Interpolation Polynomial **4**

3 Newton Interpolation **6**
3.1 General Form and Determination of Coefficients 6
3.2 Divided Differences (D.D.) 7
3.2.1 Calculation of Divided Differences Table 7
3.3 Newton's Interpolation in the Equidistant Case 7
3.3.1 Table of Progressive Finite Differences 8

4 Interpolation Error **8**

1 Introduction

Polynomial interpolation is a powerful tool for estimating values between known data points. It fits a polynomial function to given points, creating a continuous approximation of the underlying relationship.

1.1 Position of the Interpolation Problem

Given $(n+1)$ points $\{(x_i, y_i)\}_{i=0}^{n}$ with distinct abscissas $x_i \neq x_j$. We seek a polynomial P such that $P(x_i) = y_i$ for all i.

Graphical representation of Interpolation

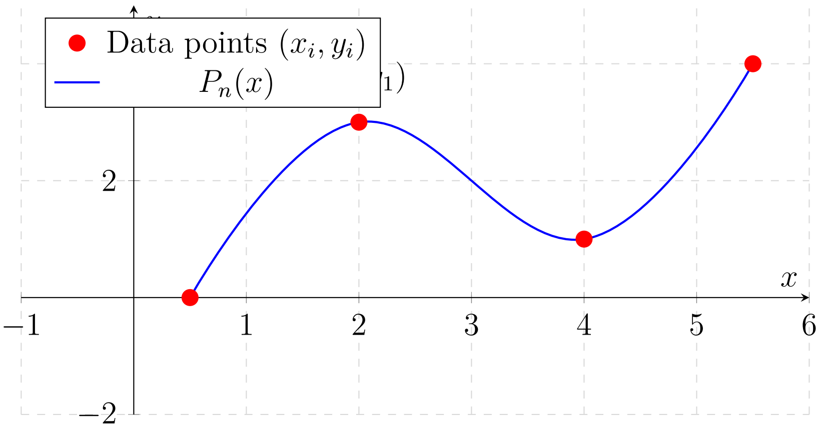

Figure 1: The interpolating polynomial passing exactly through discrete nodes.

Definition 1.1: Interpolation Polynomial and Error

The polynomial P is called the interpolation polynomial of f if:

$$\begin{cases} P(x_i) = f(x_i), & i = 0, \dots, n \\ \deg(P) \leq n. \end{cases}$$

The **approximation error** is defined as: $\epsilon(x) = |f(x) - P(x)|$.

1.2 Existence and Uniqueness

Theorem 1.1

If nodes x_i are distinct, there exists a unique polynomial P_n of degree $\leq n$ satisfying $P_n(x_i) = y_i$.

Proof 1.1

We can write the polynomial P_n as follows

$$P_n(x) = \sum_{k=0}^{n} a_k x^k = a_0 + a_1 x + a_2 x^2 + ... + a_n x^n$$

Since $P_n(x_i) = y_i$, i = 0,..., n, the coefficients $a_k; k = 0, ..., n$ satisfy the system

$$(S) \begin{cases} a_0 + a_1 x_0^1 + a_2 x_0^2 + :::::: + a_n x_0^n = y_0 \\ a_0 + a_1 x_1^1 + a_2 x_1^2 + :::::: + a_n x_1^n = y_1 \\ .... \\ a_0 + a_1 x_n^1 + a_2 x_n^2 + :::::: + a_n x_n^n = y \end{cases} \Leftrightarrow$$

$$\begin{pmatrix} 1 & x_0^1 & x_0^2 & ... & x_0^n \\ 1 & x_1^1 & x_1^2 & ... & x_1^n \\ ... & ... & ... & ... & \\ & & & & \\ 1 & x_n^1 & x_n^2 & ... & x_n^n \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ ... \\ ... \\ a_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ ... \\ .. \\ y_n \end{pmatrix}$$

Thus, the coefficients $(a_i)_i$ of the interpolation polynomial are obtained by solving the system linear

$$\Delta x = b$$

The Δ matrix is called the **Vandermonde** matrix. $det\Delta \neq 0$ because x_i are distinct, hence the existence and uniqueness of the solution of system (S).

1.3 Some Simple Examples (Direct Method)

Example 1.1: Linear Interpolation

Determine the interpolation polynomial for the following tabulated function:

x_i	2	5
y_i	-9	18

Solution: We have $n + 1 = 2 \Rightarrow n = 1$, so $P_1(x) = a_0 + a_1 x$. The coefficients satisfy the system:

$$\begin{pmatrix} 1 & 2 \\ 1 & 5 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} -9 \\ 18 \end{pmatrix}$$

Calculating the determinants:

- $\det \Delta = \begin{vmatrix} 1 & 2 \\ 1 & 5 \end{vmatrix} = 3 \neq 0 \quad \det \Delta_0 = \begin{vmatrix} -9 & 2 \\ 18 & 5 \end{vmatrix} = -81 \quad ,\det \Delta_1 = \begin{vmatrix} 1 & -9 \\ 1 & 18 \end{vmatrix} = 27$

Therefore:

$$a_0 = \frac{\det \Delta_0}{\det \Delta} = \frac{-81}{3} = -27, \quad a_1 = \frac{\det \Delta_1}{\det \Delta} = \frac{27}{3} = 9$$

Result: $P(x) = -27 + 9x$

Example 1.2: Quadratic Interpolation

Determine the interpolation polynomial for the following tabulated function:

x_i	1	2	5
y_i	0	-9	18

Solution: We have $n+1=3 \Rightarrow n=2$, so $P_2(x)=a_0+a_1x+a_2x^2$. The coefficients satisfy:

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 5 & 25 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} 0 \\ -9 \\ 18 \end{pmatrix}$$

Calculating the determinants:

- $\det\Delta = 12 \neq 0 \quad \det\Delta_0 = 216, \quad \det\Delta_1 = -270, \quad \det\Delta_2 = 54$

Therefore:

$$a_0 = \frac{216}{12} = 18, \quad a_1 = \frac{-270}{12} = -22.5, \quad a_2 = \frac{54}{12} = 4.5$$

Result: $P(x) = 18 - 22.5x + 4.5x^2$

Important Remark

If n is a sufficiently large natural number, this method involves solving an $(n+1)\times(n+1)$ linear system, which is computationally expensive and difficult. Therefore, other techniques such as **Lagrange** or **Newton** interpolation must be sought.

2 Lagrange Interpolation Polynomial

Definition 2.1: Elementary Lagrange Polynomials

The elementary Lagrange polynomials, denoted as $\ell_k(x)$, for a set of $n+1$ distinct nodes $\{x_0, x_1, \ldots, x_n\}$ are defined by:

$$\ell_k(x) = \prod_{\substack{j=0 \\ j\neq k}}^{n} \frac{x - x_j}{x_k - x_j} \tag{1}$$

Proposition 2.1: Properties of Elementary Lagrange Polynomials

The polynomials $\ell_k(x)$ for $k \in \{0, \dots, n\}$ satisfy:

1. Each ℓ_k is a polynomial of degree n.
2. Kronecker Property:

$$\ell_k(x_i) = \delta_{ki} = \begin{cases} 1 & \text{if } k = i \\ 0 & \text{if } k \neq i \end{cases} \tag{2}$$

Proof 2.1: Properties of Lagrange Basis

1. **Degree:** For any k, $\ell_k(x)$ is a product of n linear factors of the form $(x - x_j)$, hence $\deg(\ell_k) = n$. For example, if $k = 0$:

$$\ell_0(x) = \frac{x - x_1}{x_0 - x_1} \times \frac{x - x_2}{x_0 - x_2} \times \cdots \times \frac{x - x_n}{x_0 - x_n}$$

2. **Evaluation:**
   - If $i = k$: $\ell_k(x_k) = \prod_{j \neq k} \frac{x_k - x_j}{x_k - x_j} = 1$.
   - If $i \neq k$: The product contains the term $(x_i - x_i)$ in the numerator when $j = i$, thus $\ell_k(x_i) = 0$.

Remark: Roots of Lagrange Basis

Each elementary Lagrange polynomial $\ell_k(x)$ vanishes (i.e., becomes zero) at exactly n distinct nodes:

$$x_0, x_1, \dots, x_{k-1}, x_{k+1}, \dots, x_n$$

In other words, the set of roots of $\ell_k(x)$ is $\{x_i\}_{i \neq k}$. Since $\deg(\ell_k) = n$, these are all its possible roots.

Definition 2.2: Lagrange Interpolation Polynomial

The Lagrange interpolation polynomial P_n passing through the points $(x_i, y_i)_{i=0}^n$ is given by:

$$P_n(x) = \sum_{i=0}^{n} y_i \ell_i(x) = y_0 \ell_0(x) + y_1 \ell_1(x) + \cdots + y_n \ell_n(x) \tag{3}$$

Note that $P_n(x_k) = \sum_{i=0}^n y_i \ell_i(x_k) = y_k \ell_k(x_k) = y_k$.

Example 2.1: Construction of Lagrange Polynomial

Determine the Lagrange interpolation polynomial for the following data:

x_i	1	3	4
y_i	3	0	5

Solution: We have $n + 1 = 3 \Rightarrow n = 2$. The polynomial is:

$$P_2(x) = y_0\ell_0(x) + y_1\ell_1(x) + y_2\ell_2(x) = 3\ell_0(x) + 0 + 5\ell_2(x)$$

Calculating the basis polynomials:

- $\ell_0(x) = \frac{(x-3)(x-4)}{(1-3)(1-4)} = \frac{x^2-7x+12}{6}$
- $\ell_2(x) = \frac{(x-1)(x-3)}{(4-1)(4-3)} = \frac{x^2-4x+3}{3}$

Substituting back:

$$P_2(x) = 3\left(\frac{x^2 - 7x + 12}{6}\right) + 5\left(\frac{x^2 - 4x + 3}{3}\right) = \frac{13}{6}x^2 - \frac{61}{6}x + 11$$

Important Remark on Recurrence

The Lagrange method is **not recursive**. It is impossible to deduce P_{n+1} from P_n easily. Adding a new point x_{n+1} requires re-calculating **all** basis polynomials ℓ_i. This motivates the need for **Newton's Interpolation**, which allows for a recursive construction.

3 Newton Interpolation

3.1 General Form and Determination of Coefficients

Newton's Polynomial uses the divided differences of f at the given points.

Definition 3.1: Newton's Interpolation Form

The general form of Newton's interpolation polynomial is:

$$P_n(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \cdots + a_n \prod_{k=0}^{n-1}(x - x_k) \tag{4}$$

To determine the coefficients a_i, we use the interpolation conditions:

- $P_n(x_0) = f(x_0) \implies a_0 = f(x_0)$
- $P_n(x_1) = f(x_1) \implies a_1 = \frac{f(x_1)-f(x_0)}{x_1-x_0} = f[x_0, x_1]$

Proceeding in the same way, we find that $a_k = f[x_0, x_1, \ldots, x_k]$.

3.2 Divided Differences (D.D.)

Definition 3.2: Divided Differences of Order k

Let f be defined on $[a, b]$ with distinct nodes $x_0, \dots, x_n$:

- **Order 0:** $f[x_i] = f(x_i)$
- **Order 1:** $f[x_i, x_{i+1}] = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}$
- **Order k:** $f[x_i, \dots, x_{i+k}] = \frac{f[x_{i+1}, \dots, x_{i+k}] - f[x_i, \dots, x_{i+k-1}]}{x_{i+k} - x_i}$

3.2.1 Calculation of Divided Differences Table

To compute the coefficients, we form the following table:

x_i	$f(x_i)$	$DD1$	$DD2$	DDn
x_0	$\mathbf{a_0}$			
		$\mathbf{a_1}$		
x_1	$f(x_1)$		$\mathbf{a_2}$	
		$f[x_1, x_2]$		$\mathbf{a_n}$
x_2	$f(x_2)$		$\dots$	

Example 3.1: Newton Form for Square Root Function

Calculate $P_2(x)$ for $f(x) = \sqrt{1+x}$ at $x \in \{0, 3, 8\}$.

x_i	$f(x_i)$	$f[x_i, x_{i+1}]$	$f[x_i, x_{i+1}, x_{i+2}]$
0	**1**		
3	2	$\frac{2-1}{3-0} = \mathbf{1/3}$	
8	3	$\frac{3-2}{8-3} = 1/5$	$\frac{1/5-1/3}{8-0} = \mathbf{-1/60}$

Result: $P_2(x) = 1 + \frac{1}{3}x - \frac{1}{60}x(x-3) = -\frac{1}{60}x^2 + \frac{23}{60}x + 1$.

3.3 Newton's Interpolation in the Equidistant Case

When $x_i = x_0 + ih$, h is the **interpolation step**.

Definition 3.3: Progressive Finite Differences (P.F.D.)

- **Order 1:** $\Delta f(x_i) = f(x_{i+1}) - f(x_i)$
- **Order k:** $\Delta^k f(x_i) = \Delta^{k-1} f(x_{i+1}) - \Delta^{k-1} f(x_i)$

3.3.1 Table of Progressive Finite Differences

x_i	$f(x_i)$	Δf	$\Delta^2 f$	$\Delta^n f$
x_0	$f(x_0)$			
		$\Delta f(x_0)$		
x_1	$f(x_1)$		$\Delta^2 f(x_0)$	
		$\Delta f(x_1)$		$\Delta^n f(x_0)$

Theorem 3.1: Relationship and Finite Difference Form

The relationship is: $\frac{\Delta^k y_i}{k!h^k} = f[x_i, \dots, x_{i+k}]$. The polynomial is:

$$P_n(x) = f(x_0) + \sum_{k=1}^{n} \frac{\Delta^k f(x_0)}{k!h^k} \prod_{j=0}^{k-1} (x - x_j)$$

Example 3.2: Application

Points: (0, 1), (2, 5), (4, 17) with $h = 2$.

x_i	y_i	Δy_i	$\Delta^2 y_i$
0	**1**		
2	5	**4**	
4	17	12	**8**

$P_2(x) = 1 + \frac{4}{1! \cdot 2}(x - 0) + \frac{8}{2! \cdot 2^2}(x - 0)(x - 2) = 1 + 2x + (x^2 - 2x) = x^2 + 1.$

4 Interpolation Error

Theorem 4.1: Error Bound Formula

If $f \in C^{n+1}[a, b]$, then for any $x \in [a, b]$, there exists $\zeta \in (a, b)$ such that:

$$f(x) - P_n(x) = \frac{f^{(n+1)}(\zeta)}{(n+1)!} \prod_{i=0}^{n} (x - x_i) \tag{5}$$

The maximum error bound is: $E_n(x) \leq \frac{M}{(n+1)!} \prod_{i=0}^{n} |x - x_i|$, where $M = \sup |f^{(n+1)}(x)|$.

Proof 4.1: Proof of the Error Bound Formula

Let $x \in [a, b]$ be a fixed point. If $x = x_i$ for any $i \in \{0, \dots, n\}$, then both sides of the equation are zero, and the theorem holds. Assume $x \neq x_i$.
Define an auxiliary function $g(t)$ for $t \in [a, b]$ as follows:

$$g(t) = f(t) - P_n(t) - [f(x) - P_n(x)]\frac{\omega(t)}{\omega(x)}$$

where $\omega(t) = \prod_{i=0}^{n}(t - x_i)$.
Step 1: Counting the roots of $g(t)$

- For $t = x_i$, we have $g(x_i) = f(x_i) - P_n(x_i) - [f(x) - P_n(x)]\frac{0}{\omega(x)} = 0 - 0 = 0$ (since P_n interpolates f at x_i). This gives $n + 1$ roots.
- For $t = x$, we have $g(x) = f(x) - P_n(x) - [f(x) - P_n(x)]\frac{\omega(x)}{\omega(x)} = 0$. This gives 1 additional root.

Thus, $g(t)$ has at least $n + 2$ **distinct roots** in $[a, b]$.
Step 2: Applying Rolle's Theorem By Rolle's Theorem, between any two roots of $g(t)$, there is at least one root of $g'(t)$. Since $g(t)$ has $n + 2$ roots:

- $g'(t)$ has at least $n + 1$ roots.
- $g''(t)$ has at least n roots.
- Repeating this $n + 1$ times, $g^{(n+1)}(t)$ has at least **one root**, let's call it $\zeta \in (a, b)$.

Step 3: Calculating $g^{(n+1)}(\zeta)$ Differentiating $g(t)$ $n + 1$ times with respect to t:

1. $\frac{d^{n+1}}{dt^{n+1}}f(t) = f^{(n+1)}(t)$.
2. $\frac{d^{n+1}}{dt^{n+1}}P_n(t) = 0$ (since P_n is a polynomial of degree $\leq n$).
3. $\frac{d^{n+1}}{dt^{n+1}}\omega(t) = (n + 1)!$ (since $\omega(t) = t^{n+1} + \dots$).

So, $g^{(n+1)}(\zeta) = f^{(n+1)}(\zeta) - 0 - [f(x) - P_n(x)]\frac{(n+1)!}{\omega(x)} = 0$.
Step 4: Conclusion Solving for $f(x) - P_n(x)$:

$$f(x) - P_n(x) = \frac{f^{(n+1)}(\zeta)}{(n + 1)!}\omega(x) = \frac{f^{(n+1)}(\zeta)}{(n + 1)!}\prod_{i=0}^{n}(x - x_i)$$

By taking the absolute value and replacing $f^{(n+1)}(\zeta)$ with its supremum M, we obtain the maximum error bound.

Example 4.1: Accuracy of Square Root Approximation

How accurately can $\sqrt{115}$ be calculated using Lagrange's formula to interpolate $f(x) = \sqrt{x}$ at nodes: $x_0 = 100, x_1 = 121, x_2 = 144$?

Solution: For $n = 2$, we have $n + 1 = 3$. The error bound for $P_2(x)$ is:

$$|f(115) - P_2(115)| \leq \frac{M}{3!}|(115 - 100)(115 - 121)(115 - 144)|$$

1. **Product Calculation:** $|15 \times (-6) \times (-29)| = 2610$.
2. **Derivatives:** $f^{(3)}(x) = \frac{3}{8}x^{-5/2}$.
3. **Max Derivative** M**:** Since $f^{(3)}$ decreases on $[100, 144]$, $M = |f^{(3)}(100)| = \frac{3}{8}10^{-5}$.
4. **Final Bound:**

$$|E_2(115)| \leq \frac{3/8 \cdot 10^{-5}}{6} \times 2610 = 0.16 \times 10^{-4}$$

The estimated theoretical error at $x = 115$ is approximately 0.16×10^{-4}.