Numerical Techniques for Data Estimation:

Interpolation and Extrapolation

Anik Kirtania

June 23, 2025

Abstract

This document provides a comprehensive overview of interpolation and extrapolation techniques used in numerical analysis. These methods allow estimation of hypothetical values for variables based on observed data points. We discuss the fundamental concepts, mathematical formulations, and compare various techniques including piecewise constant, linear, polynomial, and spline interpolation methods. The document explores the advantages, disadvantages, implementation considerations, and appropriate use cases for each approach. The material is presented in a structured format suitable for both educational purposes and practical implementation reference.

Contents

1 Introduction

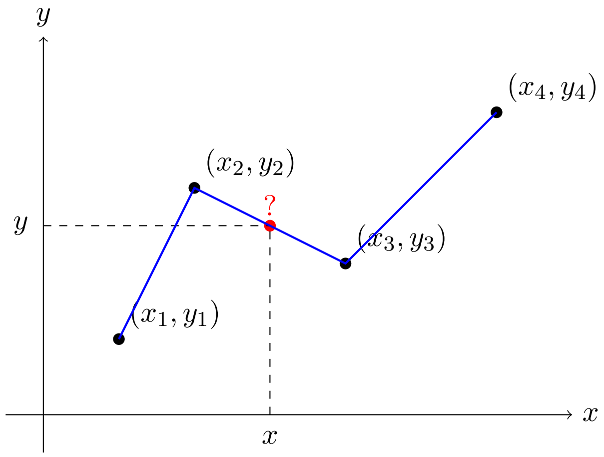

Figure 1: Visualization of the interpolation problem: estimating y for a given x between known data points.

Interpolation and extrapolation are fundamental numerical techniques used to estimate unknown values based on a set of observed data points. Both methods serve the common objective of determining hypothetical values that are not explicitly provided in the original dataset.

1.1 Problem Statement

Consider a dataset consisting of N distinct points in a two-dimensional coordinate plane, denoted as (x_i, y_i) for $i = 1, 2, \ldots, N$. For each point, both the x_i and y_i values are known. The objective is to determine the value of y corresponding to a given value of x that is not present in the original dataset.

Definition

Given a set of N data points $\{(x_i, y_i)\}_{i=1}^{N}$, where each x_i is unique, find a function f such that $f(x_i) = y_i$ for all $i = 1, 2, \ldots, N$. Then use f to estimate $y = f(x)$ for any x not in the original dataset.

1.2 Notation

- $\{(x_i, y_i)\}_{i=1}^{N}$: Set of N known data points
- $x_{\min} = \min\{x_i\}_{i=1}^{N}$: Minimum value of x in the dataset
- $x_{\max} = \max\{x_i\}_{i=1}^{N}$: Maximum value of x in the dataset
- x: Query point for which we seek to estimate y
- y: Unknown value corresponding to x that we seek to estimate

2 Distinction Between Interpolation and Extrapolation

While both interpolation and extrapolation aim to estimate unknown values, they differ in the location of the query point relative to the range of the existing data.

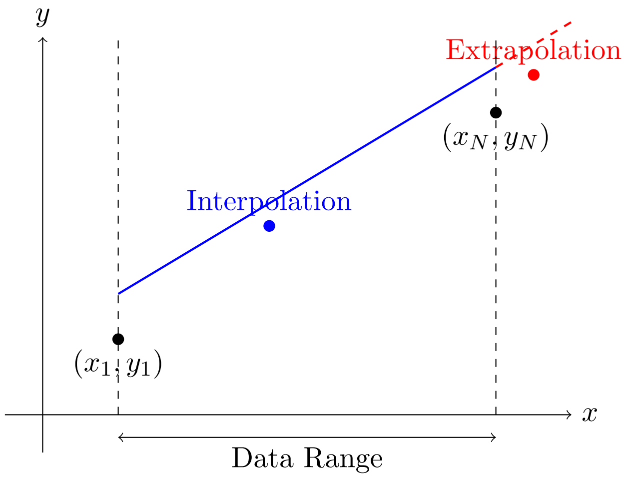

Figure 2: Comparison of interpolation (within data range) and extrapolation (outside data range).

2.1 Interpolation

Interpolation involves estimating y for a value of x that lies within the range of the observed x values:

$$x_{\min} \leq x \leq x_{\max} \tag{1}$$

The key characteristic of interpolation is that it predicts values within the boundaries of the known data. This generally yields more reliable estimates as it does not require extending patterns beyond observed limits.

2.2 Extrapolation

Extrapolation involves estimating y for a value of x that lies outside the range of the observed x values:

$$x < x_{\min} \quad \text{or} \quad x > x_{\max} \tag{2}$$

Extrapolation requires extending patterns observed within the data to regions where no observations exist. This makes extrapolation inherently more uncertain and prone to errors, especially when the underlying relationship changes beyond the observed range.

3 Interpolation Techniques

Several methods exist for performing interpolation, each with distinct characteristics in terms of complexity, accuracy, and smoothness.

3.1 Piecewise Constant Interpolation

Also known as nearest neighbor interpolation, this is the simplest form of interpolation.

3.1.1 Method

For any unknown value x, find the nearest data point x_{nearest} in the dataset and use its corresponding y value:

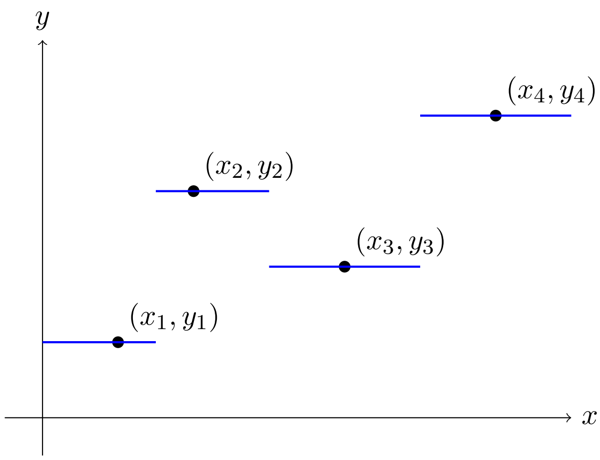

Figure 3: Piecewise constant (nearest neighbor) interpolation.

$$y = y_{\text{nearest}} \quad \text{where} \quad x_{\text{nearest}} =_{x_i} |x_i - x| \tag{3}$$

3.1.2 Algorithm

Algorithm 1 Piecewise Constant Interpolation

procedure NEARESTNEIGHBORINTERPOLATION($x, \{(x_i, y_i)\}_{i=1}^{N}$)
    min_distance $\leftarrow \infty$
    nearest_index $\leftarrow -1$
    for $i \leftarrow 1$ to N **do**
        distance $\leftarrow |x_i - x|$
        if distance $<$ min_distance **then**
            min_distance $\leftarrow$ distance
            nearest_index $\leftarrow i$
        end if
    end for
    return $y_{\text{nearest\_index}}$
end procedure

3.1.3 Advantages and Disadvantages

Advantages	Disadvantages
Simple to understand and implement	Ignores all data points except the nearest neighbor
Computationally efficient	Not smooth at transition points
Works with irregularly spaced data	Poor accuracy compared to other methods
Minimal computational requirements	Can produce discontinuous results

Table 1: Pros and cons of piecewise constant interpolation

3.1.4 Use Cases

Piecewise constant interpolation is rarely used for low-dimensional data but becomes more common with high-dimensional data where more complex methods become computationally prohibitive.

Note

While simple, piecewise constant interpolation is generally not recommended for applications requiring smooth transitions or precise estimates between data points.

3.2 Linear Interpolation

Linear interpolation fits a straight line between the two nearest points on either side of the query point.

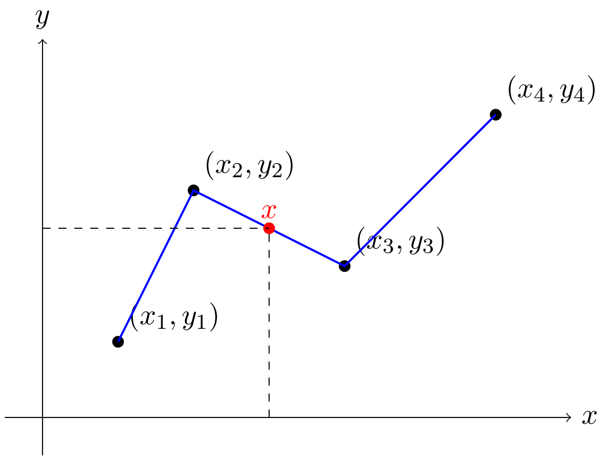

Figure 4: Linear interpolation between adjacent data points.

3.2.1 Method

For a query point x such that $x_i \leq x \leq x_{i+1}$:

$$y = y_i + \frac{y_{i+1} - y_i}{x_{i+1} - x_i}(x - x_i) \tag{4}$$

This can also be expressed as:

$$y = \frac{x_{i+1} - x}{x_{i+1} - x_i} y_i + \frac{x - x_i}{x_{i+1} - x_i} y_{i+1} \tag{5}$$

3.2.2 Algorithm

3.2.3 Advantages and Disadvantages

3.3 Polynomial Interpolation

Polynomial interpolation fits a smooth polynomial of degree $N - 1$ through all N data points.

Algorithm 2 Linear Interpolation

procedure LINEARINTERPOLATION($x, \{(x_i, y_i)\}_{i=1}^{N}$)
  if $x \leq x_1$ **then**
    return y_1 ▷ Extrapolation case
  else if $x \geq x_N$ **then**
    return y_N ▷ Extrapolation case
  end if
  for $i \leftarrow 1$ to $N-1$ **do**
    if $x_i \leq x \leq x_{i+1}$ **then**
      $t \leftarrow \frac{x-x_i}{x_{i+1}-x_i}$ ▷ Normalized position in interval [0,1]
      return $y_i + t \times (y_{i+1} - y_i)$
    end if
  end for
end procedure

Advantages	**Disadvantages**
Simple to understand and implement	Not smooth at knot points (where segments meet)
More accurate than constant interpolation	Only uses two neighboring points, ignoring all other data
Computationally efficient	Limited precision for highly non-linear data
Works well for approximately linear data	First derivative is discontinuous at data points

Table 2: Pros and cons of linear interpolation

3.3.1 Method

The polynomial interpolant $P(x)$ of degree $\leq N-1$ satisfies:

$$P(x_i) = y_i \quad \text{for} \quad i = 1, 2, \ldots, N \tag{6}$$

One common form is the Lagrange polynomial:

$$P(x) = \sum_{i=1}^{N} y_i \cdot L_i(x) \tag{7}$$

where $L_i(x)$ are the Lagrange basis polynomials:

$$L_i(x) = \prod_{\substack{j=1 \\ j \neq i}}^{N} \frac{x - x_j}{x_i - x_j} \tag{8}$$

Note

For N data points, the unique interpolating polynomial has degree at most $N-1$.

Warning

High-degree polynomial interpolation can lead to oscillations near the endpoints, a phenomenon known as Runge's phenomenon, particularly with equidistant points.

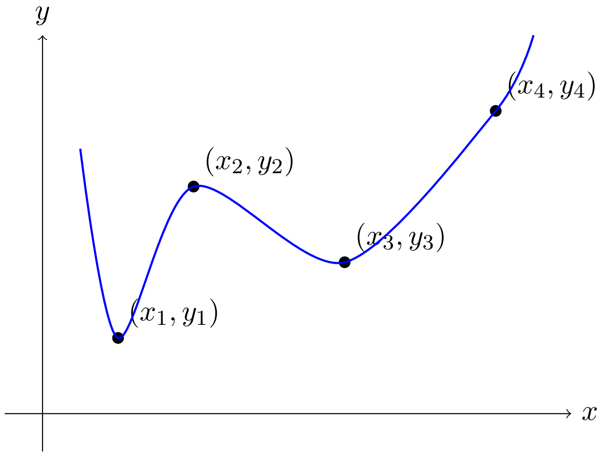

Figure 5: Polynomial interpolation through all data points.

3.3.2 Advantages and Disadvantages

Advantages	Disadvantages
Smooth across the entire domain	Computationally expensive for large datasets
Considers all data points	Prone to oscillations (Runge's phenomenon)
Exact fit through all data points	High-degree polynomials can lead to unrealistic behavior
Single continuous function	Very sensitive to outliers

Table 3: Pros and cons of polynomial interpolation

3.4 Spline Interpolation

Spline interpolation fits piecewise low-order polynomials (typically cubic) between adjacent data points while maintaining smoothness at the connection points.

3.4.1 Method

For cubic splines, a separate cubic polynomial is defined for each interval $[x_i, x_{i+1}]$:

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3 \tag{9}$$

The coefficients are determined to ensure:

- $S_i(x_i) = y_i$ and $S_i(x_{i+1}) = y_{i+1}$ (interpolation)
- $S'_i(x_{i+1}) = S'_{i+1}(x_{i+1})$ (first derivative continuity)
- $S''_i(x_{i+1}) = S''_{i+1}(x_{i+1})$ (second derivative continuity)

3.4.2 Advantages and Disadvantages

4 Extrapolation Considerations

Extrapolation uses the same techniques as interpolation but applies them outside the range of observed data. This introduces significant uncertainty and potential for error.

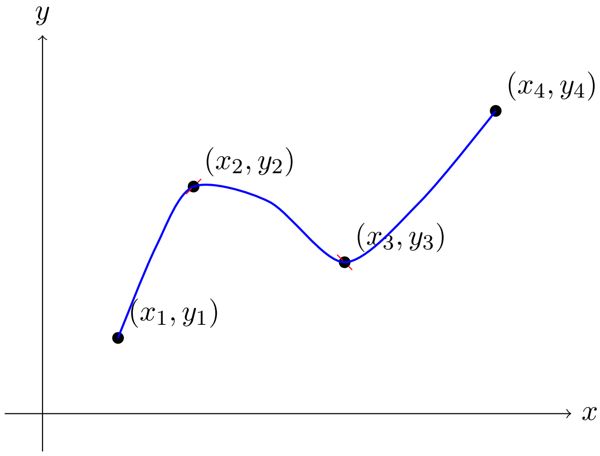

Figure 6: Cubic spline interpolation with smooth transitions at knots.

Advantages	**Disadvantages**
Smooth curves with continuous derivatives	More complex implementation than linear or constant methods
Avoids oscillation issues of high-degree polynomials	Requires solving a system of equations
Good balance between accuracy and computational efficiency	Not as precise as high-degree polynomials for certain functions
Local changes only affect nearby segments	Requires additional boundary conditions

Table 4: Pros and cons of spline interpolation

4.1 Limitations of Extrapolation

- Assumes that patterns within the observed data continue beyond its boundaries
- Accuracy decreases with distance from the known data range
- Different interpolation methods can lead to dramatically different extrapolation results
- Cannot account for unknown behavior changes or regime shifts beyond the observed data

Warning

Extrapolation is inherently risky and should be used with caution. The further away from the observed data, the less reliable the predictions become.

4.2 Best Practices for Extrapolation

- Use the simplest model that adequately fits the observed data
- Include confidence or prediction intervals to indicate increasing uncertainty
- When possible, validate extrapolated predictions against domain knowledge
- Consider multiple extrapolation methods and compare results
- Limit the range of extrapolation to minimize error

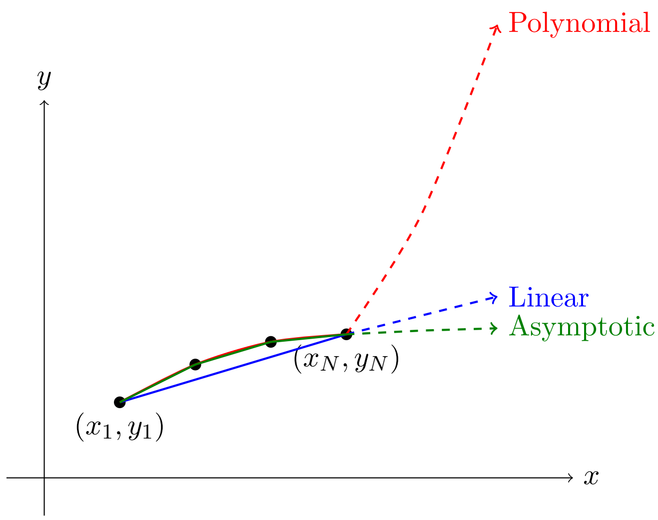

Figure 7: Different extrapolation methods can lead to dramatically different predictions.

5 Implementation Considerations

5.1 Data Preprocessing

- Sort data points by x values to ensure proper ordering
- Remove duplicate x values to prevent ambiguity
- Consider normalizing or scaling data for numerical stability
- Identify and handle outliers that may affect the interpolation quality

5.2 Method Selection

Selection criteria for interpolation methods:

- For small, sparse datasets: Linear interpolation often provides a good balance
- For large datasets with smooth underlying functions: Spline interpolation typically works well
- For data with sharp transitions: Piecewise methods with appropriate breakpoints
- For theoretical perfect fit: Polynomial interpolation (with awareness of oscillation risks)
- For high-dimensional data: Nearest neighbor methods may be necessary

5.3 Computational Efficiency

Method	Preprocessing Time	Query Time	Memory
Piecewise Constant	$O(1)$	$O(N)$ or $O(\log N)$	$O(N)$
Linear	$O(N \log N)$ for sorting	$O(\log N)$	$O(N)$
Polynomial (Lagrange)	$O(N)$	$O(N^2)$	$O(N)$
Cubic Spline	$O(N)$	$O(\log N)$	$O(N)$

Table 5: Computational complexity comparison

5.4 Implementation Example: Cubic Spline in Python

Python Implementation

```python
import numpy as np
from scipy.interpolate import CubicSpline

# Sample data
x = np.array([0, 1, 2, 3, 4, 5])
y = np.array([0, 1, 4, 9, 16, 25])

# Create cubic spline interpolation
cs = CubicSpline(x, y)

# Points to evaluate
x_new = np.linspace(0, 5, 100)

# Interpolated values
y_new = cs(x_new)
```

6 Comparison of Methods

Feature	Nearest Neighbor	Linear	Polynomial	Cubic Spline
Smoothness	None	C	C	C^2
Computational Cost	Very Low	Low	High	Medium
Implementation Difficulty	Very Easy	Easy	Medium	Medium
Accuracy	Low	Medium	High	High
Robustness to Outliers	High	Medium	Very Low	Medium
Suitability for Extrapolation	Poor	Limited	Poor	Limited

Table 6: Qualitative comparison of interpolation methods

7 Conclusion

Interpolation and extrapolation are essential techniques in numerical analysis, data science, and engineering for estimating unknown values based on observed data. The choice of method depends on various factors including the nature of the data, computational resources, desired accuracy, and smoothness requirements.

Key takeaways:

- Interpolation (within data range) is generally more reliable than extrapolation (outside data range)
- Simpler methods (nearest neighbor, linear) are computationally efficient but less accurate
- More complex methods (polynomial, spline) provide smoother and often more accurate results but at higher computational cost
- Spline interpolation often represents the best balance between smoothness, accuracy, and computational efficiency
- Extrapolation should be used with caution and with appropriate uncertainty quantification

While modern computational tools have made these techniques accessible and efficient, understanding their theoretical foundations remains crucial for their appropriate application.

References

[1] Chapra, S.C., & Canale, R.P. (2015). *Numerical Methods for Engineers*. McGraw-Hill Education.

[2] Press, W.H., Teukolsky, S.A., Vetterling, W.T., & Flannery, B.P. (2007). *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press.

[3] De Boor, C. (1978). *A Practical Guide to Splines*. Springer-Verlag.

[4] Burden, R.L., & Faires, J.D. (2011). *Numerical Analysis*. Cengage Learning.

[5] Atkinson, K.E. (1989). *An Introduction to Numerical Analysis*. John Wiley & Sons.