MAT 103: Numerical Analysis I

Topic 4: Approximation of Functions by Polynomials

Dr. Anna Fome

2026-06-30

Table of contents

"Polynomial approximation is the art of replacing something complicated with something simple — and knowing how much you have lost in doing so."

1 Introduction

Suppose you are given a function $f(x)$ that is:

- defined only at a **few data points** (e.g. measurements from an experiment), or
- too **complicated** to differentiate or integrate analytically, or
- a **black box** — you can evaluate it but have no formula.

In all these cases, we need a way to:

1. **Estimate** f at points where we have no data.
2. **Approximate** derivatives $f'(x)$, $f''(x)$.
3. **Approximate** definite integrals $\int_a^b f(x)\,dx$.

The key idea of this topic is: **replace** $f(x)$ by a polynomial $P(x)$ that passes through the known data points. Polynomials are easy to evaluate, differentiate, and integrate — so once we have $P(x)$, we can do all three tasks above.

This process is called **polynomial interpolation**.

> **Definition**
>
> **Polynomial interpolation** is the construction of a polynomial $P(x)$ that passes exactly through a given set of data points:
>
> $$(x_0, f_0),\ (x_1, f_1),\ \ldots,\ (x_n, f_n) \tag{1}$$

where $f_i = f(x_i)$. The x_i are called **nodes** (or interpolation points).

By the end of this topic you should be able to:

- Build and apply the Lagrange interpolating polynomial.
- State and use the Lagrange error term.
- Build a forward and backward difference table.
- Apply Newton's Forward and Backward Difference formulas.
- Apply Stirling's and Bessel's central difference formulas.
- Use an interpolating polynomial to approximate derivatives and integrals.
- Use MATLAB/MAPLE for interpolation.

2 Lagrange Interpolation

2.1 The Main Idea

Given $n+1$ data points $(x_0, f_0), (x_1, f_1), \dots, (x_n, f_n)$, we want a polynomial $P_n(x)$ of degree **at most** n that satisfies:

$$P_n(x_i) = f_i \quad \text{for every } i = 0, 1, \dots, n \tag{2}$$

The clever trick in Lagrange interpolation is to build $P_n(x)$ from $n+1$ simpler pieces called **Lagrange basis polynomials**, one for each node.

2.2 The Lagrange Basis Polynomials

For each node x_i, define:

$$L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j} = \frac{(x - x_0) \cdots (x - x_{i-1})(x - x_{i+1}) \cdots (x - x_n)}{(x_i - x_0) \cdots (x_i - x_{i-1})(x_i - x_{i+1}) \cdots (x_i - x_n)} \tag{3}$$

The key property is:

$$L_i(x_j) = \begin{cases} 1 & \text{if } j = i \\ 0 & \text{if } j \neq i \end{cases} \tag{4}$$

In words: $L_i(x)$ equals **1 at its own node** and **0 at every other node**.

This is exactly what we need: it means when we add the pieces together, each f_i contributes only at x_i and nowhere else.

2.3 The Lagrange Interpolating Polynomial

$$\boxed{P_n(x) = \sum_{i=0}^{n} f_i \cdot L_i(x) = f_0 L_0(x) + f_1 L_1(x) + \cdots + f_n L_n(x)} \tag{5}$$

> **Why Does This Work?**
>
> Check: at $x = x_k$, all $L_i(x_k) = 0$ except $L_k(x_k) = 1$ by Eq. 4. So $P_n(x_k) = f_k \cdot 1 = f_k$ from Eq. 5 The polynomial passes through every data point — exactly as required by Eq. 2.

2.4 Example 4.1 — Linear Lagrange Interpolation ($n = 1$)

Two data points: $x_0 = 1$, $f_0 = 2$ and $x_1 = 4$, $f_1 = 8$.

Estimate $f(2.5)$.

Step 1 — Build the basis polynomials using Eq. 3:

$$L_0(x) = \frac{x - x_1}{x_0 - x_1} = \frac{x - 4}{1 - 4} = \frac{x - 4}{-3}$$

$$L_1(x) = \frac{x - x_0}{x_1 - x_0} = \frac{x - 1}{4 - 1} = \frac{x - 1}{3}$$

Step 2 — Evaluate at $x = 2.5$:

$$L_0(2.5) = \frac{2.5 - 4}{-3} = \frac{-1.5}{-3} = 0.5$$

$$L_1(2.5) = \frac{2.5 - 1}{3} = \frac{1.5}{3} = 0.5$$

Step 3 — Compute $P_1(2.5)$ using Eq. 5:

$$P_1(2.5) = f_0 \cdot L_0(2.5) + f_1 \cdot L_1(2.5) = 2(0.5) + 8(0.5) = 1 + 4 = 5$$

Result: $f(2.5) \approx 5$.

(Note: $L_0 + L_1 = 0.5 + 0.5 = 1$ — the basis polynomials always sum to 1.)

2.5 Example 4.2 — Quadratic Lagrange Interpolation ($n = 2$)

Three data points:

x	1	3	5
$f(x)$	1	9	25

Estimate $f(2)$.

(Note: these values come from $f(x) = x^2$.)

Step 1 — Build the three basis polynomials using Eq. 3:

$$L_0(x) = \frac{(x-3)(x-5)}{(1-3)(1-5)} = \frac{(x-3)(x-5)}{(-2)(-4)} = \frac{(x-3)(x-5)}{8}$$

$$L_1(x) = \frac{(x-1)(x-5)}{(3-1)(3-5)} = \frac{(x-1)(x-5)}{(2)(-2)} = \frac{(x-1)(x-5)}{-4}$$

$$L_2(x) = \frac{(x-1)(x-3)}{(5-1)(5-3)} = \frac{(x-1)(x-3)}{(4)(2)} = \frac{(x-1)(x-3)}{8}$$

Step 2 — Evaluate each L_i at $x = 2$:

$$L_0(2) = \frac{(2-3)(2-5)}{8} = \frac{(-1)(-3)}{8} = \frac{3}{8}$$

$$L_1(2) = \frac{(2-1)(2-5)}{-4} = \frac{(1)(-3)}{-4} = \frac{-3}{-4} = \frac{3}{4}$$

$$L_2(2) = \frac{(2-1)(2-3)}{8} = \frac{(1)(-1)}{8} = \frac{-1}{8}$$

Check: $L_0 + L_1 + L_2 = \frac{3}{8} + \frac{3}{4} - \frac{1}{8} = \frac{3+6-1}{8} = \frac{8}{8} = 1$

Step 3 — Compute $P_2(2)$ using Eq. 5:

$$P_2(2) = 1 \cdot \frac{3}{8} + 9 \cdot \frac{3}{4} + 25 \cdot \left(-\frac{1}{8}\right) = \frac{3}{8} + \frac{27}{4} - \frac{25}{8} = \frac{3 + 54 - 25}{8} = \frac{32}{8} = 4$$

Result: $P_2(2) = 4$.

True value: $f(2) = 2^2 = 4$ (exact, because $f(x) = x^2$ is a degree-2 polynomial and we used 3 nodes — so the interpolant reproduces f exactly).

2.6 Example 4.3 — Lagrange with Real Data

The following table gives values of $f(x) = \ln(x)$:

x	2	4	6
$f(x) = \ln(x)$	0.6931	1.3863	1.7918

Estimate $\ln(5)$ using Lagrange interpolation.

(True value: $\ln(5) = 1.6094$)

Step 1 — Basis polynomials at $x = 5$, using Eq. 3:

$$L_0(5) = \frac{(5-4)(5-6)}{(2-4)(2-6)} = \frac{(1)(-1)}{(-2)(-4)} = \frac{-1}{8}$$

$$L_1(5) = \frac{(5-2)(5-6)}{(4-2)(4-6)} = \frac{(3)(-1)}{(2)(-2)} = \frac{-3}{-4} = \frac{3}{4}$$

$$L_2(5) = \frac{(5-2)(5-4)}{(6-2)(6-4)} = \frac{(3)(1)}{(4)(2)} = \frac{3}{8}$$

Check: $-\frac{1}{8} + \frac{3}{4} + \frac{3}{8} = \frac{-1+6+3}{8} = \frac{8}{8} = 1$

Step 2 — Interpolated value using Eq. 5:

$$P_2(5) = 0.6931\left(-\frac{1}{8}\right) + 1.3863\left(\frac{3}{4}\right) + 1.7918\left(\frac{3}{8}\right)$$

$$= -0.0866 + 1.0397 + 0.6719 = 1.6250$$

Result: $\ln(5) \approx 1.6250$

Error: $|1.6250 - 1.6094| = 0.0156$ — reasonable for such widely spaced nodes.

2.7 The Error Term in Lagrange Interpolation

Whenever we use $P_n(x)$ to approximate $f(x)$, there is an error. The **Lagrange error formula** tells us exactly how large that error can be.

> **Lagrange Error Formula**
>
> If f has $n+1$ continuous derivatives, the error at any point x is:

$$f(x) - P_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^{n} (x - x_i) \tag{6}$$

for some unknown ξ between the smallest and largest of $x, x_0, x_1, \ldots, x_n$.

What does Eq. 6 tell us?

- The error depends on how large $f^{(n+1)}$ is — if f changes rapidly, the error is larger, else the error is small. The term $(n+1)!$ in the denominator helps to reduce the error.
- The error also depends on $\prod(x - x_i)$ — the further x is from the nodes, the larger this product, and the larger the error.
- Using **more nodes** (larger n) increases the degree and reduces the error, provided $f^{(n+1)}$ stays manageable.

Practical bound: Since ξ is unknown, we replace $f^{(n+1)}(\xi)$ with its maximum value M_{n+1} on the interval:

$$|f(x) - P_n(x)| \leq \frac{M_{n+1}}{(n+1)!} \left| \prod_{i=0}^{n} (x - x_i) \right| \tag{7}$$

2.7.1 Example 4.4 — Error Bound

For Example 4.2 ($f(x) = x^2$, nodes at $1, 3, 5$, query at $x = 2$):

$n = 2$, so we need $f^{(3)}(x)$. Since $f(x) = x^2$, $f^{(3)}(x) = 0$ everywhere.

By Eq. 6:

$$\text{Error} = \frac{0}{3!} \times \ldots = 0$$

The interpolation is **exact** — as expected, since $P_2(x)$ reproduces any polynomial of degree ≤ 2 without error.

For $f(x) = \ln(x)$ (Example 4.3), $f^{(3)}(x) = \frac{2}{x^3}$. On $[2, 6]$, the maximum is $\frac{2}{2^3} = 0.25$.

Applying Eq. 7:

$$|E| \leq \frac{0.25}{6} |(5-2)(5-4)(5-6)| = \frac{0.25}{6} \times 3 = 0.125$$

Our actual error was 0.0156 — well within this bound

3 Finite Difference Interpolation Polynomials

3.1 Why Another Method?

Lagrange interpolation works well for any set of nodes $x_0, x_1, \ldots, x_n$. But when the nodes are **equally spaced** — that is, $x_i = x_0 + i \cdot h$ for a fixed step size h — we can use a much more systematic approach called **finite difference interpolation**.

This approach:

- organises all the data in a neat **difference table** (built once, used many times),
- gives simple formulas for interpolation, differentiation, and integration,
- is historically very important and still widely used in numerical analysis.

3.2 The Difference Operator Δ

For equally spaced nodes with step size h, the **forward difference operator** Δ is:

$$\Delta f_i = f_{i+1} - f_i \tag{8}$$

This is simply the change in f from one node to the next. We can apply Δ repeatedly to get **higher-order differences**:

$$\Delta^2 f_i = \Delta(\Delta f_i) = \Delta f_{i+1} - \Delta f_i = f_{i+2} - 2f_{i+1} + f_i \tag{9}$$

$$\Delta^3 f_i = \Delta(\Delta^2 f_i) = \Delta^2 f_{i+1} - \Delta^2 f_i \tag{10}$$

In general:

$$\Delta^k f_i = \Delta^{k-1} f_{i+1} - \Delta^{k-1} f_i \tag{11}$$

3.3 Building the Difference Table

The **forward difference table** organises all the differences in a triangular array:

x_i	f_i	Δf_i	$\Delta^2 f_i$	$\Delta^3 f_i$	$\Delta^4 f_i$
x_0	f_0	Δf_0	$\Delta^2 f_0$	$\Delta^3 f_0$	$\Delta^4 f_0$
x_1	f_1	Δf_1	$\Delta^2 f_1$	$\Delta^3 f_1$	
x_2	f_2	Δf_2	$\Delta^2 f_2$		
x_3	f_3	Δf_3			
x_4	f_4				

Rule: every entry = (entry below it) − (entry beside it on the left). Equivalently: every entry = (entry below-left) − (entry beside-left).

> How to Build the Table — One Step at a Time
>
> Work **column by column**, left to right. Each new column is obtained by subtracting consecutive entries of the previous column, using Eq. 11. The top entry of each column is what we call $\Delta^k f_0$.

3.3.1 Example 4.5 — Building a Difference Table

Build the forward difference table for:

x	1	2	3	4
$f(x) = x^3$	1	8	27	64

Here $h = 1$, $x_0 = 1$.

Column 1 — First differences Δf_i, using Eq. 8:

$$\Delta f_0 = f_1 - f_0 = 8 - 1 = 7$$
$$\Delta f_1 = f_2 - f_1 = 27 - 8 = 19$$
$$\Delta f_2 = f_3 - f_2 = 64 - 27 = 37$$

Column 2 — Second differences $\Delta^2 f_i$, using Eq. 11:

$$\Delta^2 f_0 = \Delta f_1 - \Delta f_0 = 19 - 7 = 12$$
$$\Delta^2 f_1 = \Delta f_2 - \Delta f_1 = 37 - 19 = 18$$

Column 3 — Third differences $\Delta^3 f_i$:

$$\Delta^3 f_0 = \Delta^2 f_1 - \Delta^2 f_0 = 18 - 12 = 6$$

Complete table:

x_i	f_i	Δf_i	$\Delta^2 f_i$	$\Delta^3 f_i$
1	**1**	**7**	**12**	**6**
2	8	19	18	
3	27	37		
4	64			

The **bolded top diagonal** — $f_0 = 1$, $\Delta f_0 = 7$, $\Delta^2 f_0 = 12$, $\Delta^3 f_0 = 6$ — is what we will use in the Newton Forward formula.

> **Useful Pattern**
>
> For a polynomial of degree k, the k-th differences are **constant** and all higher differences are **zero**. Since $f(x) = x^3$ is degree 3, $\Delta^3 f = 6$ (constant) and $\Delta^4 f = 0$ everywhere.

3.4 Newton's Forward Difference Formula

When we want to interpolate **near the beginning of the table** (i.e., x is close to x_0), we use Newton's Forward Difference formula.

Define the dimensionless variable:

$$s = \frac{x - x_0}{h} \tag{12}$$

So $s = 0$ at x_0, $s = 1$ at x_1, $s = 2$ at x_2, and so on. For a point **between** x_0 and x_1, we have $0 < s < 1$.

The formula is:

$$\boxed{P_n(x) = f_0 + s\,\Delta f_0 + \frac{s(s-1)}{2!}\,\Delta^2 f_0 + \frac{s(s-1)(s-2)}{3!}\,\Delta^3 f_0 + \cdots} \tag{13}$$

Or more compactly, using binomial-style notation:

$$P_n(x) = \sum_{k=0}^{n} \binom{s}{k} \Delta^k f_0, \quad \text{where } \binom{s}{k} = \frac{s(s-1)(s-2)\cdots(s-k+1)}{k!} \tag{14}$$

Only the top row of the difference table is used: f_0, Δf_0, $\Delta^2 f_0$, $\Delta^3 f_0$, ...

3.4.1 Example 4.6 — Newton Forward Interpolation

Using the table from Example 4.5, estimate $f(1.5)$.

Here $x_0 = 1$, $h = 1$, $x = 1.5$.

By Eq. 12:

$$s = \frac{1.5 - 1}{1} = 0.5$$

From the top row: $f_0 = 1$, $\Delta f_0 = 7$, $\Delta^2 f_0 = 12$, $\Delta^3 f_0 = 6$.

Applying Eq. 13:

$$P_3(1.5) = f_0 + s\,\Delta f_0 + \frac{s(s-1)}{2}\,\Delta^2 f_0 + \frac{s(s-1)(s-2)}{6}\,\Delta^3 f_0$$

Compute each term:

Term	Calculation	Value
f_0	1	1
$s\,\Delta f_0$	0.5×7	3.5
$\frac{s(s-1)}{2}\,\Delta^2 f_0$	$\frac{0.5 \times (-0.5)}{2} \times 12 = \frac{-0.25}{2} \times 12$	-1.5
$\frac{s(s-1)(s-2)}{6}\,\Delta^3 f_0$	$\frac{0.5 \times (-0.5) \times (-1.5)}{6} \times 6 = \frac{0.375}{6} \times 6$	0.375

$$P_3(1.5) = 1 + 3.5 - 1.5 + 0.375 = 3.375$$

True value: $(1.5)^3 = 3.375$ (exact, because $f(x) = x^3$ is degree 3 and we used 4 nodes)

3.4.2 Example 4.7 — Newton Forward on a Real Table

The following table gives values of e^x:

x	0.0	0.1	0.2	0.3
e^x	1.000000	1.105171	1.221403	1.349859

Estimate $e^{0.15}$.

(True value: $e^{0.15} = 1.161834$)

Step 1 — Build the difference table ($h = 0.1$, $x_0 = 0.0$):

x	f	Δf	$\Delta^2 f$	$\Delta^3 f$
0.0	**1.000000**	**0.105171**	**0.011061**	**0.001163**
0.1	1.105171	0.116232	0.012224	
0.2	1.221403	0.128456		
0.3	1.349859			

Step 2 — Compute s, using Eq. 12:

$$s = \frac{0.15 - 0.0}{0.1} = 1.5$$

Step 3 — Apply Eq. 13 (using top row values in bold):

Term	Calculation	Value
f_0	1.000000	1.000000
$s\,\Delta f_0$	1.5×0.105171	0.157757
$\frac{s(s-1)}{2}\Delta^2 f_0$	$\frac{1.5 \times 0.5}{2} \times 0.011061 = 0.375 \times 0.011061$	0.004148
$\frac{s(s-1)(s-2)}{6}\Delta^3 f_0$	$\frac{1.5 \times 0.5 \times (-0.5)}{6} \times 0.001163 = -0.0625 \times 0.001163$	−0.000073

$$P_3(0.15) = 1.000000 + 0.157757 + 0.004148 - 0.000073 = 1.161832$$

True value: $e^{0.15} = 1.161834$ — error of only 0.000002!

3.5 Newton's Backward Difference Formula

When we want to interpolate **near the end of the table** (i.e., x is close to x_n, the last node), we use the **Backward Difference formula**.

First define the **backward difference operator** ∇:

$$\nabla f_i = f_i - f_{i-1} \tag{15}$$

$$\nabla^2 f_i = \nabla f_i - \nabla f_{i-1} = f_i - 2f_{i-1} + f_{i-2} \tag{16}$$

Now define s measuring distance **backwards** from the last node x_n:

$$s = \frac{x - x_n}{h} \qquad \text{(note: } s \leq 0 \text{ for } x \leq x_n\text{)} \tag{17}$$

The **Newton Backward Difference Formula** is:

$$\boxed{P_n(x) = f_n + s\,\nabla f_n + \frac{s(s+1)}{2!}\,\nabla^2 f_n + \frac{s(s+1)(s+2)}{3!}\,\nabla^3 f_n + \cdots} \tag{18}$$

Only the bottom row (last diagonal) of the difference table is used: $f_n,\ \nabla f_n,\ \nabla^2 f_n,\ \nabla^3 f_n, \ldots$

The backward differences can be read from the **bottom-right diagonal** of the same difference table built in the forward direction:

x_i	f_i	Δf_i	$\Delta^2 f_i$	$\Delta^3 f_i$
x_0	f_0	Δf_0	$\Delta^2 f_0$	$\mathbf{{}^3f_0 = {}^3f_3}$
x_1	f_1	Δf_1	$\mathbf{{}^2f_1 = {}^2f_3}$	
x_2	f_2	$\mathbf{f_2 = f_3}$		
x_3	$\mathbf{f_3}$			

So $\nabla f_n = \Delta f_{n-1}$, $\nabla^2 f_n = \Delta^2 f_{n-2}$, etc.

3.5.1 Example 4.8 — Newton Backward Interpolation

Using the e^x table from Example 4.7, estimate $e^{0.25}$.

(True value: $e^{0.25} = 1.284025$*)*

$x_n = 0.3$, $h = 0.1$, $x = 0.25$.

By Eq. 17:

$$s = \frac{0.25 - 0.3}{0.1} = -0.5$$

Bottom-right diagonal (backward differences at $x_n = 0.3$):

$$f_n = 1.349859, \quad \nabla f_n = \Delta f_2 = 0.128456, \quad \nabla^2 f_n = \Delta^2 f_1 = 0.012224, \quad \nabla^3 f_n = \Delta^3 f_0 = 0.001163$$

Apply Eq. 18:

Term	Calculation	Value
f_n	1.349859	1.349859
$s\,\nabla f_n$	$(-0.5)(0.128456)$	-0.064228
$\frac{s(s+1)}{2}\nabla^2 f_n$	$\frac{(-0.5)(0.5)}{2} \times 0.012224 = -0.125 \times 0.012224$	-0.001528
$\frac{s(s+1)(s+2)}{6}\nabla^3 f_n$	$\frac{(-0.5)(0.5)(1.5)}{6} \times 0.001163 = -0.0625 \times 0.001163$	-0.000073

$$P_3(0.25) = 1.349859 - 0.064228 - 0.001528 - 0.000073 = 1.284030$$

True value: $e^{0.25} = 1.284025$ — error of only 0.000005

3.6 Stirling's and Bessel's Formulas

When x is **near the middle** of the table, neither the forward nor the backward formula gives the best accuracy — they are optimised for the two ends.

Stirling's formula and **Bessel's formula** are central difference formulas designed for interpolation near the middle of the table. They use values from **both sides** of the point of interest, which gives better accuracy.

3.6.1 When to Use Which Formula

Formula	Best used when x is near…	Uses values from…
Newton Forward (Eq. 13)	Beginning of table ($x \approx x_0$)	Top row
Newton Backward (Eq. 18)	End of table ($x \approx x_n$)	Bottom diagonal
Stirling's (Eq. 19)	Middle, $s \approx 0$	Both sides of x_0
Bessel's (Eq. 21)	Middle, $s \approx 0.5$	Both sides

3.6.2 Stirling's Formula

With $s = (x - x_0)/h$ measured from the middle node x_0, and using the **central difference** notation $\delta f_{i+1/2} = f_{i+1} - f_i$, Stirling's formula is:

$$P(x) = f_0 + s\,\mu\delta f_0 + \frac{s^2}{2!}\delta^2 f_0 + \frac{s(s^2-1)}{3!}\mu\delta^3 f_0 + \frac{s^2(s^2-1)}{4!}\delta^4 f_0 + \cdots \tag{19}$$

where

$$\mu\delta f_0 = \frac{1}{2}\left(\Delta f_0 + \Delta f_{-1}\right) \tag{20}$$

is the **mean of adjacent differences**.

Reading Eq. 19: odd-order terms (1st, 3rd) use the *average* of the two differences straddling x_0; even-order terms (2nd, 4th) use a single, directly-centred difference. Stirling's formula is most accurate when x is close to a tabulated node, i.e. s close to 0.

3.6.3 Bessel's Formula

Bessel's formula is particularly accurate when $0.25 < s < 0.75$ (i.e., x is roughly midway between two nodes):

$$P(x) = \frac{f_0 + f_1}{2} + \left(s - \frac{1}{2}\right)\Delta f_0 + \frac{s(s-1)}{2!} \cdot \frac{\Delta^2 f_{-1} + \Delta^2 f_0}{2} + \frac{\left(s - \frac{1}{2}\right) s(s-1)}{3!}\Delta^3 f_{-1} + \cdots \tag{21}$$

Reading Eq. 21: the starting value is the *average* of the two function values straddling the gap; even-order terms (2nd, 4th) use the *average* of the two differences straddling the midpoint; odd-order terms (1st, 3rd) use a single, directly-centred difference. This is the mirror image of Stirling's formula, because Bessel's is centred *between* two nodes rather than *on* one node.

> **In Practice**
>
> For most examination problems and practical computations, Newton's Forward and Backward formulas are sufficient. Stirling's and Bessel's formulas provide better accuracy when interpolating near the centre of the table and are covered here for completeness.

3.6.4 Example 4.8b — Stirling's and Bessel's Formulas

Using the table $f(x) = x^2$ below, estimate $f(1.5)$ using both Stirling's and Bessel's formulas.

x	0	1	2	3	4
$f(x) = x^2$	0	1	4	9	16

Difference table:

x_i	f_i	Δf_i	$\Delta^2 f_i$
0	0	1	2
1	1	3	2
2	4	5	2
3	9	7	
4	16		

(True value: $f(1.5) = 2.25$*)*

Stirling's formula, centred at $x_0 = 1$, so $s = \dfrac{1.5 - 1}{1} = 0.5$:

Here $\Delta f_{-1} = 1$ (row $x = 0$), $\Delta f_0 = 3$ (row $x = 1$), $\Delta^2 f_{-1} = 2$ (row $x = 0$).

By Eq. 20: $\mu\delta f_0 = \dfrac{1+3}{2} = 2$.

Applying Eq. 19 (up to 2nd order):

$$P(1.5) = 1 + (0.5)(2) + \frac{(0.5)^2}{2}(2) = 1 + 1 + 0.25 = 2.25$$

Bessel's formula, between $x_0 = 1$ and $x_1 = 2$, so $s = 0.5$:

Here $f_0 = 1, f_1 = 4, \Delta f_0 = 3, \Delta^2 f_{-1} = 2, \Delta^2 f_0 = 2$.

Applying Eq. 21 (up to 2nd order):

$$P(1.5) = \frac{1+4}{2} + (0.5 - 0.5)(3) + \frac{(0.5)(-0.5)}{2} \cdot \frac{2+2}{2} = 2.5 + 0 - 0.25 = 2.25$$

Both formulas give the exact value, 2.25, since $f(x) = x^2$ is degree 2 and we used differences up to exactly the 2nd order.

4 Using Interpolating Polynomials for Derivatives and Integrals

We have spent this entire topic building polynomials $P_n(x)$ that pass through known data points. This section answers the natural next question: **now that we have** $P_n(x)$, what can we do with it?

The answer is: anything we could do with the original function $f(x)$, but more easily — because $P_n(x)$ is just a polynomial.

- **Differentiating** $P_n(x)$ gives us an approximation to $f'(x)$, $f''(x)$, and higher derivatives.
- **Integrating** $P_n(x)$ gives us an approximation to $\displaystyle\int f(x)\,dx$.

Both ideas rest on the same trick: substitute $x = x_0 + sh$ into Newton's Forward formula (Eq. 13), then differentiate or integrate with respect to s instead of x — since s is dimensionless and the resulting expressions are much simpler.

4.1 Numerical Differentiation

4.1.1 Deriving the First Derivative Formula

Recall Newton's Forward formula (Eq. 13), written in terms of $s = \dfrac{x - x_0}{h}$:

$$P_n(x) = f_0 + s\,\Delta f_0 + \frac{s(s-1)}{2!}\Delta^2 f_0 + \frac{s(s-1)(s-2)}{3!}\Delta^3 f_0 + \cdots$$

To differentiate $P_n(x)$ with respect to x, we use the chain rule. Since $x = x_0 + sh$, we have $\dfrac{ds}{dx} = \dfrac{1}{h}$, so:

$$\frac{d}{dx} = \frac{ds}{dx} \cdot \frac{d}{ds} = \frac{1}{h}\frac{d}{ds} \tag{22}$$

> **Why This Substitution Helps**
>
> Differentiating directly with respect to x would force us to track h inside every term. By switching to s via Eq. 22, we differentiate a clean polynomial in s first, and only divide by h (or h^2, for second derivatives) at the very end. This keeps the algebra simple.

We now differentiate each term of $P_n(x)$ with respect to s:

$$\frac{dP_n}{ds} = \Delta f_0 + \frac{2s-1}{2!}\Delta^2 f_0 + \frac{3s^2 - 6s + 2}{3!}\Delta^3 f_0 + \cdots \tag{23}$$

Evaluated at $x = x_0$ (i.e. $s = 0$):

$$\left.\frac{dP_n}{ds}\right|_{s=0} = \Delta f_0 - \frac{1}{2}\Delta^2 f_0 + \frac{1}{3}\Delta^3 f_0 - \cdots$$

Dividing by h using Eq. 22 gives:

$$\boxed{f'(x_0) \approx \frac{1}{h}\left[\Delta f_0 - \frac{1}{2}\Delta^2 f_0 + \frac{1}{3}\Delta^3 f_0 - \frac{1}{4}\Delta^4 f_0 + \cdots\right]} \tag{24}$$

> **The Pattern in the Coefficients**
>
> Notice the coefficients $1, -\frac{1}{2}, \frac{1}{3}, -\frac{1}{4}, \ldots$ in Eq. 24 alternate in sign and follow $\dfrac{(-1)^{k+1}}{k}$ for the k-th difference term. This is the same pattern that appears in the Taylor series for $\ln(1+s)$ — not a coincidence, since Newton's Forward formula and this series are deeply related through the calculus of finite differences.

4.1.2 Deriving the Second Derivative Formula

To get $f''(x_0)$, we differentiate $P_n(x)$ **twice** with respect to s, then divide by h^2 (since each derivative with respect to x contributes a factor of $\frac{1}{h}$, by Eq. 22):

$$\left.\frac{d^2P_n}{ds^2}\right|_{s=0} = \Delta^2 f_0 - \Delta^3 f_0 + \frac{11}{12}\Delta^4 f_0 - \cdots$$

$$\boxed{f''(x_0) \approx \frac{1}{h^2}\left[\Delta^2 f_0 - \Delta^3 f_0 + \frac{11}{12}\Delta^4 f_0 - \cdots\right]} \tag{25}$$

A Quick Sanity Check on the Formula

The leading term of Eq. 25 is $\frac{\Delta^2 f_0}{h^2}$. This matches the familiar **central second-difference approximation**:

$$f''(x_0) \approx \frac{f_2 - 2f_1 + f_0}{h^2} = \frac{\Delta^2 f_0}{h^2}$$

The extra terms ($-\Delta^3 f_0$, $+\frac{11}{12}\Delta^4 f_0$, …) are simply **correction terms** that improve accuracy beyond this basic approximation.

4.1.3 Example 4.9 — Numerical Differentiation (First Derivative)

Using the e^x table from Example 4.7, estimate $(e^x)'$ at $x = 0$.

(True value: $\left.\frac{d}{dx}e^x\right|_{x=0} = e^0 = 1$)

From the top row: $\Delta f_0 = 0.105171$, $\Delta^2 f_0 = 0.011061$, $\Delta^3 f_0 = 0.001163$. Step size $h = 0.1$.

Applying Eq. 24:

$$f'(0) \approx \frac{1}{0.1}\left[0.105171 - \frac{0.011061}{2} + \frac{0.001163}{3}\right]$$

$$= 10\,[0.105171 - 0.005531 + 0.000388]$$

$$= 10 \times 0.100028 = 1.000280$$

True value: 1 — error of only 0.000280

4.1.4 Example 4.10 — Numerical Differentiation at an Interior Node

Example 4.9 estimated f' at x_0, the **first** node of the table. But Eq. 24 works just as well at **any** node, provided we re-centre the formula there — that is, treat that node as a new "x_0" and read the top row of differences starting from that row.

Estimate $(e^x)'$ at $x = 0.1$, using the same table.

(True value: $\left.\frac{d}{dx}e^x\right|_{x=0.1} = e^{0.1} = 1.105171$*)*

Step 1 — Re-centre the table at $x_0 = 0.1$. Reading the differences starting from the row $x = 0.1$:

$$\Delta f_0 = 0.116232, \qquad \Delta^2 f_0 = 0.012224$$

(Only two differences are available here, since the table has just four points and we have used up one row by shifting the centre — this is a normal limitation when working near the end of a small table.)

Step 2 — Apply Eq. 24, with $h = 0.1$:

$$f'(0.1) \approx \frac{1}{0.1}\left[0.116232 - \frac{0.012224}{2}\right] = 10\,[0.116232 - 0.006112]$$

$$= 10 \times 0.110120 = 1.101200$$

Step 3 — Compare with the true value:

$$f'(0.1) = e^{0.1} = 1.105171$$

$$\text{Error} = |1.101200 - 1.105171| = 0.003971$$

> **i** Why Is the Error Larger Here Than in Example 4.9?
>
> In Example 4.9, we had **three** correction terms available ($\Delta f_0, \Delta^2 f_0, \Delta^3 f_0$). Here, shifting the centre to $x = 0.1$ leaves only **two** terms, because the table runs out of rows below $x = 0.1$. Fewer terms means a less accurate approximation — this is why **larger tables** generally give better numerical derivatives, especially for nodes other than the very first one.

4.1.5 Example 4.11 — Numerical Second Derivative

Using an extended e^x table (five points instead of four), estimate $(e^x)''$ at $x = 0$.

(True value: $\left.\frac{d^2}{dx^2}e^x\right|_{x=0} = e^0 = 1$*)*

Step 1 — Build the extended table by adding one more point, $x = 0.4$, $f(0.4) = e^{0.4} = 1.491825$:

x	f	Δf	$\Delta^2 f$	$\Delta^3 f$	$\Delta^4 f$
0.0	1.000000	0.105171	0.011061	0.001163	0.000122
0.1	1.105171	0.116232	0.012224	0.001286	
0.2	1.221403	0.128456	0.013510		
0.3	1.349859	0.141966			
0.4	1.491825				

Step 2 — Apply Eq. 25, using all three available terms:

$$f''(0) \approx \frac{1}{h^2}\left[\Delta^2 f_0 - \Delta^3 f_0 + \frac{11}{12}\Delta^4 f_0\right]$$

$$= \frac{1}{(0.1)^2}\left[0.011061 - 0.001163 + \frac{11}{12}(0.000122)\right]$$

$$= 100\,[0.011061 - 0.001163 + 0.000112]$$

$$= 100 \times 0.010010 = 1.0010$$

Step 3 — Compare with the true value:

$$f''(0) = e^0 = 1, \qquad \text{Error} = |1.0010 - 1| = 0.0010$$

Comparing to a Shorter Calculation

If we had stopped after only the first term of Eq. 25 (the basic central-difference approximation):

$$f''(0) \approx \frac{\Delta^2 f_0}{h^2} = \frac{0.011061}{0.01} = 1.1061$$

— giving an error of 0.1061, over **100 times larger** than the error from using all three terms (0.0010). This illustrates why the extra correction terms matter, especially for second derivatives.

4.1.6 Connecting Polynomials to Integration: The Next Frontier

Just as we have differentiated Newton's forward difference interpolating polynomial to approximate derivatives, we can integrate it to approximate definite integrals. Suppose we want to evaluate a definite integral over a structured grid bounded by x_0 and x_n.

To simplify the calculus, we map the physical variable x onto our dimensionless tracking variable s using the exact same change of variable transformation:

$$x = x_0 + sh \implies dx = h\,ds$$

By transforming the boundaries of integration alongside the differential operator, our limits shift systematically:

- When $x = x_0$: $x_0 = x_0 + sh \implies s = 0$
- When $x = x_n$: Since $x_n = x_0 + nh$, substituting yields $x_0 + nh = x_0 + sh \implies s = n$

Substituting these parameters directly into the classic definite integral structure yields a foundational, scaled calculus identity:

$$\int_{x_0}^{x_n} f(x)\,dx \approx \int_{x_0}^{x_n} P_n(x)\,dx = \int_0^n P_n(s) \cdot (h\,ds) = h \int_0^n P_n(s)\,ds \tag{26}$$

Evaluating the polynomial integration sequence in terms of s completely isolates the uniform grid step size h from the inner calculus operations.

4.1.7 Coming Up in Topic 5(a): The Newton-Cotes Framework

We have successfully built a bridge connecting polynomial interpolation to both components of continuous calculus. The transformation formula in Eq. 26 acts as a master key.

In the next section, we will systematically change the polynomial degree n to unlock the family of **Newton-Cotes formulas**:

- What geometric shape arises if we set $n = 1$ to integrate a straight secant line across two grid points?
- How dramatically will our calculation error drop if we shift to $n = 2$ to capture local curvature with a parabolic arc?

Turn to **Topic 5(a)** to watch these algebraic substitutions unfold into the foundational rules of numerical quadrature.

> **Looking Ahead**
>
> In **Topic 5**, we study the **Newton–Cotes integration formulas** in full detail, including Simpson's $\frac{1}{3}$ Rule, Simpson's $\frac{3}{8}$ Rule, and a careful treatment of their error bounds.

5 Tutorial Questions

Show all working clearly.

5.1 Section A: Concepts

Question 1

(a) Define polynomial interpolation. Why is it useful?

(b) Explain why polynomials are particularly convenient for approximating functions in numerical computation.

(c) State two situations where polynomial interpolation is the only practical approach to evaluate a function.

Question 2

State the difference between Newton's Forward (Eq. 13) and Backward (Eq. 18) Difference formulas. When should each be used? Give an example situation for each.

5.2 Section B: Lagrange Interpolation

Question 3

Given the data:

x	0	1	3
$f(x)$	1	3	55

(a) Find the three Lagrange basis polynomials $L_0(x)$, $L_1(x)$, $L_2(x)$.

(b) Write down the interpolating polynomial $P_2(x)$.

(c) Estimate $f(2)$ using $P_2(x)$.

(d) Verify by evaluating $P_2(0)$, $P_2(1)$, and $P_2(3)$ — do they match the data?

Question 4

The following table gives values of $f(x) = \sqrt{x}$:

x	1	4	9
$f(x)$	1	2	3

(a) Use Lagrange interpolation to estimate $\sqrt{6}$.

(b) The true value is $\sqrt{6} = 2.449490$. Calculate the absolute and relative error.

(c) Use the Lagrange error formula (Eq. 7) to find an upper bound on the error. *(Hint:* $f^{(3)}(x) = \frac{3}{8}x^{-5/2}$)

Question 5

Four data points are given:

x	-1	0	1	2
$f(x)$	5	1	1	7

(a) Find the Lagrange interpolating polynomial $P_3(x)$.

(b) Evaluate $P_3(0.5)$.

5.3 Section C: Finite Difference Tables

Question 6

Build the complete forward difference table for:

x	0	1	2	3	4
$f(x)$	2	5	12	23	38

(a) Show all columns Δf, $\Delta^2 f$, $\Delta^3 f$, $\Delta^4 f$.

(b) What is $\Delta^3 f_0$?

(c) Is $f(x)$ a polynomial? If so, what is its degree? How can you tell from the table?

Question 7

For the data:

x	10	20	30	40	50
$\sin(x^\circ)$	0.1736	0.3420	0.5000	0.6428	0.7660

Build the forward difference table. Use **5 decimal places** throughout.

5.4 Section D: Newton Forward, and Backward Difference Formulas

Question 8

Using the difference table from **Question 6**:

(a) Use Newton's Forward formula (Eq. 13) to estimate $f(1.5)$. Show each term separately.

(b) Use Newton's Backward formula (Eq. 18) to estimate $f(3.5)$. Show each term separately.

(c) Compare your answers with the exact values, given that $f(x) = 2x^2 + x + 2$.

Question 9

Using the sin table from **Question 7** ($h = 10°$):

(a) Use Newton's Forward formula to estimate $\sin(15°)$. *(True value:* $\sin(15°) = 0.2588$*)*

(b) Use Newton's Backward formula to estimate $\sin(45°)$. *(True value:* $\sin(45°) = 0.7071$*)*

(c) Calculate the error in each case. Which estimate is more accurate and why?

Question 10

The following data is given for $f(x) = e^x$:

x	1.0	1.2	1.4	1.6	1.8
e^x	2.7183	3.3201	4.0552	4.9530	6.0496

(a) Build the forward difference table.

(b) Estimate $e^{1.1}$ using Newton's Forward formula (use up to Δ^3).

(c) Estimate $e^{1.7}$ using Newton's Backward formula (use up to Δ^3).

(d) Compare both results with true values and compute percentage errors.

5.5 Section E: Numerical Differentiation

Question 12

Using the e^x table from **Question 11**:

(a) Estimate $f'(1.0)$ using Eq. 24, truncated to:

$$f'(x_0) \approx \frac{1}{h}\left[\Delta f_0 - \frac{1}{2}\Delta^2 f_0 + \frac{1}{3}\Delta^3 f_0\right]$$

(b) The true value is $f'(1.0) = e^{1.0} = 2.7183$. What is the error?

(c) Estimate $f''(1.0)$ using Eq. 25, truncated to:

$$f''(x_0) \approx \frac{1}{h^2}\left[\Delta^2 f_0 - \Delta^3 f_0\right]$$

Compare with the true value $e^{1.0} = 2.7183$.

End of Topic 4 Tutorial Questions