Polynomials

Why This Matters

Polynomials are functions built from **powers of x added together**. They're everywhere:

- **Curve fitting**: Approximating complex data
- **Physics**: Trajectories, orbits, energy
- **Computer graphics**: Bezier curves, splines, smoothing
- **Optimization**: Finding max/min values
- **Signal processing**: Filters, interpolation
- **Machine learning**: Polynomial regression

Polynomials are simple enough to understand yet powerful enough to model complex behavior.

The Big Picture: Building Curves from Powers

Linear: y = x (straight line) **Quadratic**: $y = x^2$ (parabola, one curve) **Cubic**: $y = x^3$ (S-shape, two curves) **Higher**: More complex curves

Key insight: Adding power terms creates increasingly flexible curves.

---

1. What Is a Polynomial?

Definition

A **polynomial** is a sum of **power terms**:

```
P(x) = aₙxⁿ + aₙ₋₁xⁿ⁻¹ + ... + a₁x + a₀

aₙ, aₙ₋₁, ..., a₁, a₀ = coefficients (constants)
n = degree (highest power)
```

Examples

```
f(x) = 3x² - 2x + 1        (degree 2, quadratic)
g(x) = x³ - 5x             (degree 3, cubic)
h(x) = 4x⁴ - 3x² + 7x - 2  (degree 4, quartic)
```

NOT Polynomials

```
f(x) = 1/x     (negative exponent)
g(x) = √x      (fractional exponent)
h(x) = 2ˣ      (x in exponent)
k(x) = sin(x)  (not a power of x)
```

Polynomials only have **non-negative integer exponents**.

Standard Form

Descending order of powers:

```
P(x) = aₙxⁿ + ... + a₁x + a₀

highest power first
```

---

2. Degree and Classification

Degree

The **degree** is the highest power of x.

```
Degree 0: P(x) = 5              (constant)
Degree 1: P(x) = 2x + 3         (linear)
Degree 2: P(x) = x² - 4x + 1    (quadratic)
Degree 3: P(x) = x³ + 2x² - x   (cubic)
Degree 4: P(x) = x⁴ - 3x²       (quartic)
Degree 5: P(x) = x⁵ + x         (quintic)
```

General names:

```
Degree n: nth-degree polynomial
```

Leading Coefficient

The **coefficient of the highest power term**:

```
3x⁴ - 2x² + 5

Leading term: 3x⁴
Leading coefficient: 3
```

Why it matters: Determines end behavior (what happens as $x \to \pm\infty$).

---

3. Quadratic Functions (Degree 2)

Standard Form

```
f(x) = ax² + bx + c

a ≠ 0 (otherwise it's linear)
```

Graph: Parabola

```
a > 0: Opens upward ∪
a < 0: Opens downward ∩

    │     /─\         ─\─/
    │    /   \  or     │
────┼───/─────\───   ──┼──
    │  vertex      vertex
```

Vertex: The highest or lowest point (turning point).

Vertex Form

```
f(x) = a(x - h)² + k

(h, k) = vertex coordinates

a > 0: minimum at vertex
a < 0: maximum at vertex
```

Example: $f(x) = 2(x - 3)^2 + 1$

- Vertex: (3, 1)
- Opens upward ($a = 2 > 0$)
- Minimum value: 1

Finding the Vertex from Standard Form

```
Given: f(x) = ax² + bx + c

Vertex x-coordinate: h = -b/(2a)
Vertex y-coordinate: k = f(h)
```

Example: $f(x) = x^2 - 4x + 3$

```
h = -(-4)/(2×1) = 4/2 = 2
k = f(2) = 2² - 4(2) + 3 = 4 - 8 + 3 = -1

Vertex: (2, -1)
```

Roots (Zeros)

Where the parabola crosses the x-axis (where $f(x) = 0$)

Quadratic formula:

```
If ax² + bx + c = 0, then:

       -b ± √(b² - 4ac)
x = ──────────────────────
            2a
```

Discriminant: $b^2 - 4ac$

```
> 0: Two real roots (crosses x-axis twice)
= 0: One real root (touches x-axis once)
< 0: No real roots (doesn't cross x-axis)
```

Example: $x^2 - 5x + 6 = 0$

```
a = 1, b = -5, c = 6
b² - 4ac = 25 - 24 = 1 > 0  (two roots)
```

```
x = (5 ± √1) / 2 = (5 ± 1) / 2

x = 3 or x = 2
```

Factored Form

```
f(x) = a(x - r₁)(x - r₂)

r₁, r₂ = roots (where f(x) = 0)
```

Example: $f(x) = (x - 2)(x - 3)$

- Roots at $x = 2$ and $x = 3$
- Expands to: $x^2 - 5x + 6$

Applications

Projectile motion:

```
h(t) = -16t² + v₀t + h₀

h = height
t = time
v₀ = initial velocity
h₀ = initial height
```

Profit optimization:

```
P(x) = -x² + 100x - 1000

Maximum at vertex: x = 50 units
```

---

4. Cubic Functions and Beyond (Degree 3+)

Cubic (Degree 3)

```
f(x) = ax³ + bx² + cx + d
```

Graph shapes:

```
a > 0:  ╱     a < 0:  ╲
       ╱               ╲
     ╱╲               ╱╲
    ╱  ╲             ╱  ╲
```

Characteristics:

- Up to 2 turning points
- Up to 3 real roots
- S-shaped curve

Higher Degrees

Quartic (degree 4):

```
f(x) = ax⁴ + bx³ + cx² + dx + e

- Up to 3 turning points
- Up to 4 real roots
- W or M shape
```

General pattern:

```
Degree n polynomial:
- Up to n roots
- Up to n-1 turning points
```

---

5. Operations on Polynomials

Addition/Subtraction

Combine like terms:

```
P(x) = 3x² + 2x + 1
Q(x) = x² - 5x + 3

P(x) + Q(x) = (3x² + x²) + (2x - 5x) + (1 + 3)
            = 4x² - 3x + 4
```

Programming:

```javascript
// Polynomials as arrays of coefficients [a₀, a₁, a₂, ...]
function addPolynomials(p1, p2) {
  const maxLen = Math.max(p1.length, p2.length);
  const result = [];
  for (let i = 0; i < maxLen; i++) {
    result[i] = (p1[i] || 0) + (p2[i] || 0);
  }
  return result;
}

// [1, 2, 3] represents 1 + 2x + 3x²
// [3, -5, 1] represents 3 - 5x + x²
addPolynomials([1, 2, 3], [3, -5, 1]);  // [4, -3, 4]
```

Multiplication

Distribute and combine:

```
(x + 2)(x + 3) = x² + 3x + 2x + 6
               = x² + 5x + 6
```

FOIL (for binomials):

```
(a + b)(c + d) = ac + ad + bc + bd
                 First Outer Inner Last
```

Example:

```
(2x + 1)(x - 3) = 2x² - 6x + x - 3
                = 2x² - 5x - 3
```

Division

Long division or **synthetic division** (complex, rarely done by hand).

Example use: Simplifying rational functions.

---

6. Factoring Polynomials

Why Factor?

Factoring breaks a polynomial into simpler pieces (factors).

Benefits:

- Find roots easily
- Simplify expressions
- Solve equations

Common Patterns

Greatest Common Factor (GCF)

```
6x³ + 9x² = 3x²(2x + 3)
```

Difference of Squares

```
a² - b² = (a + b)(a - b)

x² - 9 = (x + 3)(x - 3)
x² - 16 = (x + 4)(x - 4)
```

Perfect Square Trinomials

```
a² + 2ab + b² = (a + b)²
a² - 2ab + b² = (a - b)²

x² + 6x + 9 = (x + 3)²
x² - 10x + 25 = (x - 5)²
```

Quadratic Trinomials

```
x² + bx + c = (x + m)(x + n)

where m + n = b and m × n = c
```

Example: $x^2 + 7x + 12$

```
Find m, n where:
m + n = 7
m × n = 12

m = 3, n = 4

x² + 7x + 12 = (x + 3)(x + 4)
```

Sum/Difference of Cubes

```
a³ + b³ = (a + b)(a² - ab + b²)
a³ - b³ = (a - b)(a² + ab + b²)

x³ + 8 = (x + 2)(x² - 2x + 4)
x³ - 27 = (x - 3)(x² + 3x + 9)
```

Factoring Strategy

1. **GCF first**: Factor out common terms
2. **Count terms**:
   - 2 terms: Difference of squares, sum/difference of cubes
   - 3 terms: Trinomial patterns
   - 4+ terms: Grouping
3. **Check**: Multiply back to verify

---

7. Roots and Zeros

Definitions

Root (or zero): A value where $P(x) = 0$

```
If P(r) = 0, then r is a root
```

Graphically: Where the curve crosses the x-axis.

Finding Roots

Factor and solve:

```
x² - 5x + 6 = 0
(x - 2)(x - 3) = 0

x = 2 or x = 3
```

Quadratic formula (for degree 2).

Numerical methods (for degree 5+, usually can't solve algebraically).

Multiplicity

How many times a root appears:

```
P(x) = (x - 2)²(x + 1)

x = 2 is a root of multiplicity 2 (appears twice)
x = -1 is a root of multiplicity 1

Graph touches x-axis at x = 2 (doesn't cross)
Graph crosses x-axis at x = -1
```

Fundamental Theorem of Algebra

A polynomial of degree n has exactly n roots (counting multiplicity, including complex roots).

```
Degree 2: 2 roots
Degree 3: 3 roots
Degree 5: 5 roots
```

Not all roots are real. Some might be complex (involve $i = \sqrt{-1}$).

---

8. End Behavior

What It Means

What happens as $x \to \pm\infty$?

Leading Term Dominance

For large |x|, only the leading term matters:

```
P(x) = 3x⁴ - 100x³ + 5000x² - x + 999

As x → ±∞, behaves like 3x⁴
```

Even Degree

```
aₙxⁿ where n is even

aₙ > 0: Both ends up   ⁄‾＼ or ╱╲╱╲
aₙ < 0: Both ends down ＼_⁄ or ╲╱╲╱
```

Odd Degree

```
aₙxⁿ where n is odd

aₙ > 0: Left down, right up    ⁄
aₙ < 0: Left up, right down  ＼
```

Examples

```
f(x) = x³
- Odd degree, positive leading coefficient
- Left down (-∞), right up (+∞)

g(x) = -2x⁴ + 100x
- Even degree, negative leading coefficient
- Both ends down

h(x) = x² - 10000x + 999999
- Even degree, positive leading coefficient
- Both ends up (parabola shape dominates)
```

---

9. Applications

Curve Fitting

Approximate data with polynomials:

```javascript
// Fit polynomial to data points
function polyfit(points, degree) {
  // Uses least squares (complex math)
  // Returns coefficients [a₀, a₁, ..., aₙ]
}

// Example: fit quadratic to data
const data = [{x:0,y:1}, {x:1,y:3}, {x:2,y:7}, {x:3,y:13}];
const coeffs = polyfit(data, 2);  // [1, 0, 2]
// Approximation: y ≈ 1 + 2x²
```

Interpolation

Estimate values between known points:

Lagrange interpolation, splines, Bezier curves all use polynomials.

Physics

Energy, potential, forces often polynomial:

```
Potential energy: U(x) = ½kx²  (quadratic)
Taylor series: sin(x) ≈ x - x³/6 + x⁵/120 - ...  (polynomial approximation)
```

Computer Graphics

Bezier curves (parametric polynomials):

```
Cubic Bezier:
P(t) = (1-t)³P₀ + 3(1-t)²tP₁ + 3(1-t)t²P₂ + t³P₃
```

```
t ∈ [0, 1]
P₀, P₁, P₂, P₃ = control points
```

---

10. Polynomial Evaluation

Direct Evaluation

```
P(x) = 2x³ - 3x² + 5x - 1

P(2) = 2(8) - 3(4) + 5(2) - 1
     = 16 - 12 + 10 - 1
     = 13
```

Horner's Method (Efficient)

Rewrite using nested multiplication:

```
P(x) = 2x³ - 3x² + 5x - 1
     = ((2x - 3)x + 5)x - 1

Only 3 multiplications instead of 6!
```

Programming:

```
function horner(coeffs, x) {
  // coeffs = [a₀, a₁, a₂, ...] (from constant to highest)
  let result = coeffs[coeffs.length - 1];
  for (let i = coeffs.length - 2; i >= 0; i--) {
    result = result * x + coeffs[i];
  }
  return result;
}

// P(x) = -1 + 5x - 3x² + 2x³
horner([-1, 5, -3, 2], 2);  // 13
```

Much faster for high-degree polynomials.

Common Mistakes & Misconceptions

❌ "x^2 and 2x are like terms"

No. Like terms have the same exponent:

```
3x² + 5x² = 8x²   ✓
3x² + 5x ≠ 8x³    ✗ (can't combine)
```

❌ "$(x + 3)^2 = x^2 + 9$"

No! Must expand properly:

```
(x + 3)² = (x + 3)(x + 3) = x² + 6x + 9
```

❌ "All polynomials can be factored over the reals"

Some can't: $x^2 + 1$ has no real factors (roots are $\pm i$).

❌ "Degree tells you number of real roots"

Degree tells you **total roots** (including complex). Some might not be real.

❌ "Dividing by x - r gives the quotient"

Only if r is a root. Otherwise you get a quotient plus a remainder.

Tiny Practice

Identify degree and leading coefficient:

1. $P(x) = 5x^3 - 2x + 7$
2. $Q(x) = -x^4 + 3x^2 - 1$

Expand: 3. $(x + 2)(x - 3)$ 4. $(x - 1)^2$

Factor: 5. $x^2 - 9$ 6. $x^2 + 7x + 12$

Find roots: 7. $x^2 - 4 = 0$ 8. $x^2 - 5x + 6 = 0$

Evaluate: 9. $P(x) = x^3 - 2x + 1$, find $P(2)$

▶ Answers

Summary Cheat Sheet

Definition

```
Polynomial: P(x) = aₙxⁿ + ... + a₁x + a₀

Degree: highest power
Coefficients: aₙ, ..., a₁, a₀
```

Types by Degree

```
0: Constant
1: Linear (y = mx + b)
2: Quadratic (y = ax² + bx + c)
3: Cubic
4: Quartic
5+: Higher-degree
```

Quadratic Formula

```
ax² + bx + c = 0

       -b ± √(b² - 4ac)
x = ──────────────────────
            2a

Discriminant b² - 4ac:
> 0: two roots
= 0: one root
< 0: no real roots
```

Factoring Patterns

```
Difference of squares: a² - b² = (a+b)(a-b)
Perfect square: a² ± 2ab + b² = (a±b)²
Trinomial: x² + bx + c = (x+m)(x+n)
  where m+n=b, m×n=c
```

End Behavior

```
Even degree: Both ends same direction
Odd degree: Opposite directions

Sign of leading coefficient determines up/down
```

Programming

```javascript
// Coefficients as array [a₀, a₁, a₂, ...]
function evaluate(coeffs, x) {
  return coeffs.reduce((sum, c, i) => sum + c * x**i, 0);
}

// Horner's method (faster)
function horner(coeffs, x) {
  let result = coeffs[coeffs.length - 1];
  for (let i = coeffs.length - 2; i >= 0; i--) {
    result = result * x + coeffs[i];
  }
  return result;
}
```

Next Steps

Polynomials are versatile functions that model curves and data. You now understand:

- What polynomials are
- Quadratics (parabolas)
- Factoring and roots

- Applications

Next, we'll enter the realm of calculus, starting with **Limits**—the foundation for understanding change.

Continue to: 10-limits.md