

# MATHEMATICAL OPERATIONS

450 Numbered Reference Entries  
From Arithmetic to Advanced Theory

Operations • Functions • Operators  
Constructions • Symbolic and Numerical Procedures

### What Does an Operation Do?

A specified rule turns permitted inputs into an output. The inputs may be numbers, functions, vectors, sets, spaces, or entire mathematical structures.

Inputs

Rule + Domain

Output

|            |                                                                                            |
|------------|--------------------------------------------------------------------------------------------|
| Numbers    | $7 + 5 = 12$ and $2 \uparrow\uparrow 3 = 2^{2^2} = 16$                                     |
| Functions  | $(f \circ g)(x) = f(g(x))$ and $D(x^3) = 3x^2$                                             |
| Structures | $A \cup B, V \otimes W, H^p(X) \smile H^q(X)$                                              |
| Theory     | Fractional derivatives, tropical arithmetic, ordinals, operads, and cohomology operations. |

This is a broad, organized catalog, not a claim that mathematics has only 450 operations. Families, useful special cases, and related constructions are included and labeled. A finite list cannot exhaust all possible operations.

# Contents

---

The entry numbers remain consecutive across all 30 chapters. Page numbers and chapter titles are clickable in a PDF reader.

|                                            |   |
|--------------------------------------------|---|
| How to Read This Reference .....           | 4 |
| Worked Comparisons and Domain Checks ..... | 5 |

| Ch. | Subject Area                                 | Entries | Page |
|-----|----------------------------------------------|---------|------|
| 1   | Arithmetic and Basic Scalar Operations       | 1–18    | 6    |
| 2   | Elementary and Special Function Evaluations  | 19–38   | 7    |
| 3   | Rounding, Integer, and Counting Operations   | 39–54   | 8    |
| 4   | Modular and Arithmetic-Function Operations   | 55–68   | 9    |
| 5   | Hyperoperations and Iterated Powers          | 69–78   | 9    |
| 6   | Complex and Hypercomplex Operations          | 79–92   | 10   |
| 7   | Set Operations and Set Constructions         | 93–108  | 11   |
| 8   | Logical and Order-Theoretic Operations       | 109–124 | 12   |
| 9   | Function and Relation Operations             | 125–142 | 12   |
| 10  | Vector and Coordinate Operations             | 143–158 | 13   |
| 11  | Matrix Operations and Matrix Functions       | 159–180 | 14   |
| 12  | Tensor, Exterior, and Clifford Operations    | 181–192 | 15   |
| 13  | Group, Ring, and Module Operations           | 193–212 | 16   |
| 14  | Polynomials and Formal Power Series          | 213–228 | 17   |
| 15  | Differentiation and Local Calculus Operators | 229–244 | 18   |

## Reading Paths

Start with Chapters 1–5 for numbers and arithmetic; Chapters 7–11 for sets, functions, and linear algebra; Chapters 15–19 for calculus and transforms. Chapters 24–28 introduce more abstract and theoretical frameworks.

## How to Use an Entry

Read the name and label, then the definition, then the example or formula. Keep the stated domain restrictions attached to the rule. Follow the chapter source numbers for more detail, or use the alphabetical index to find an unfamiliar name.

## What Is Counted?

A numbered entry can name one operation, a family of related operations, a useful special case, or a construction. For example, squaring is included separately for lookup even though it is exponentiation with exponent two. The final chapter explicitly separates procedures.

# Contents (Continued)

The entry numbers remain consecutive across all 30 chapters. Page numbers and chapter titles are clickable in a PDF reader.

| Ch. | Subject Area                                     | Entries | Page |
|-----|--------------------------------------------------|---------|------|
| 16  | Integration and Accumulation                     | 245–258 | 19   |
| 17  | Integral and Discrete Transforms                 | 259–274 | 20   |
| 18  | Sequences, Series, and Convolution               | 275–288 | 21   |
| 19  | Fractional, Discrete, and Generalized Calculus   | 289–302 | 22   |
| 20  | Probability and Measure Operations               | 303–314 | 23   |
| 21  | Statistics and Information Functionals           | 315–326 | 24   |
| 22  | Geometric and Convex-Set Operations              | 327–340 | 25   |
| 23  | Graph and Network Operations                     | 341–354 | 25   |
| 24  | Topological Operations and Constructions         | 355–368 | 26   |
| 25  | Differential Geometry and Geometric Operators    | 369–380 | 27   |
| 26  | Category-Theoretic Constructions                 | 381–394 | 28   |
| 27  | Homology, Cohomology, and Higher Operations      | 395–408 | 28   |
| 28  | Alternative and Generalized Operation Frameworks | 409–422 | 29   |
| 29  | Optimization and Variational Constructions       | 423–434 | 30   |
| 30  | Related Symbolic and Numerical Procedures        | 435–450 | 31   |

|                                                 |    |
|-------------------------------------------------|----|
| Conventions, Pitfalls, and New Operations ..... | 32 |
| Sources and Further Study .....                 | 33 |
| Alphabetical Index .....                        | 35 |

## Entry Labels

**OP:** Operation. **FUN:** Function evaluation or scalar functional. **OPER:** Operator. **CON:** Construction. **PROC:** Procedure. These labels describe the entry's role in this book, not disjoint universal categories.

## Common Notation

|                            |                                                                                  |
|----------------------------|----------------------------------------------------------------------------------|
| $\mathbb{N}_0, \mathbb{Z}$ | Nonnegative integers; all integers.                                              |
| $\mathbb{R}, \mathbb{C}$   | Real numbers; complex numbers.                                                   |
| $f : X \rightarrow Y$      | A function with domain $X$ and codomain $Y$ .                                    |
| $x \mapsto f(x)$           | The rule describing the output for an input $x$ .                                |
| $\Sigma, \Pi, \int$        | Summation, product, and integration; each needs its relevant index or domain.    |
| PV                         | A specified principal-value convention for a singular integral.                  |
| $\cong$                    | Isomorphism or a context-specific equivalence, not necessarily literal equality. |

# How to Read This Reference

## A Precise Starting Definition

An  $n$ -ary internal operation on a set  $X$  is a function  $f : X^n \rightarrow X$ : it assigns one output in  $X$  to every allowed ordered list of  $n$  inputs from  $X$ . A mixed-type operation instead has a signature such as  $f : X_1 \times \cdots \times X_n \rightarrow Y$ . A partial operation is defined only on a specified subset of the possible input lists. [1][3]

$$\underbrace{(x_1, \dots, x_n)}_{\text{Inputs / Operands}} \xrightarrow{f} \underbrace{f(x_1, \dots, x_n)}_{\text{Output / Result}}$$

| Arity      | Number of Inputs           | Example                                                                      |
|------------|----------------------------|------------------------------------------------------------------------------|
| Nullary    | Zero                       | A selected constant $c \in X$ .                                              |
| Unary      | One                        | $x \mapsto -x$ , $x \mapsto  x $ , or a derivative operator $f \mapsto f'$ . |
| Binary     | Two                        | $(x, y) \mapsto x + y$ or $(A, B) \mapsto A \cup B$ .                        |
| Ternary    | Three                      | $(x, y, z) \mapsto x + y + z$ .                                              |
| $n$ -ary   | A fixed finite $n$         | $(x_1, \dots, x_n) \mapsto \sum_{i=1}^n x_i$ .                               |
| Infinitary | An infinite indexed family | A convergent sequence mapped to its limit; the domain is restricted.         |

## What the Entry Labels Mean

|      |                                                                                                                                                 |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| OP   | An operation combining or transforming mathematical inputs. It may be a familiar special case or an explicitly specified generalized operation. |
| FUN  | Evaluation of a named function, numerical summary, or invariant. Such evaluations are operations in a broad functional sense.                   |
| OPER | An operator acting on functions, fields, sequences, forms, distributions, or other structured inputs.                                           |
| CON  | A construction of an object or structure, often determined only up to an appropriate isomorphism.                                               |
| PROC | A symbolic or numerical procedure. These occupy Entries 435–450 and are not presented as extra primitive arithmetic operations.                 |

## Closure and Domain Come First

Subtraction is an internal operation on  $\mathbb{Z}$  but not on  $\mathbb{N}_0$ , because  $2 - 5 = -3$ . Ordinary division is a partial operation on  $\mathbb{R}$  when input pairs with a zero divisor are excluded. The real square-root function maps  $[0, \infty)$  to  $[0, \infty)$ ; extending it to complex inputs requires a branch convention. [1][4]

A symbol alone does not completely define an operation. Specify the objects, the domain, the output type, and any branch, order, sign, normalization, or convergence convention.

## Why There Is No Final Finite List

For every real number  $c$ , define a binary operation on  $\mathbb{R}$  by  $x \star_c y = x + y + c$ . If  $c \neq d$ , then  $0 \star_c 0 = c \neq d = 0 \star_d 0$ , so the operations differ. This explicit construction already produces infinitely many operations. A catalog of named operations is therefore a selection, not a boundary on what mathematics can define.

# Worked Comparisons and Domain Checks

## 1. Operation Versus Property

Addition is an operation. Commutativity is a property that an operation may satisfy:

$$x + y = y + x.$$

Subtraction does not satisfy that property:  $7 - 2 = 5$ , but  $2 - 7 = -5$ . Equality itself is a relation between objects. One can separately define a Boolean-valued test of that relation. [1][14]

## 2. Function Composition

Let  $f(x) = x^2$  and  $g(x) = x + 1$ . Then

$$(f \circ g)(x) = (x + 1)^2,$$

$$(g \circ f)(x) = x^2 + 1.$$

They are different functions. The order in a composition records which function acts first. [3]

## 3. Matrix Versus Entrywise Product

With  $A = \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix}$  and  $B = \begin{pmatrix} 3 & 0 \\ 4 & 2 \end{pmatrix}$ ,

$$AB = \begin{pmatrix} 11 & 4 \\ 4 & 2 \end{pmatrix}, \quad A \odot B = \begin{pmatrix} 3 & 0 \\ 0 & 2 \end{pmatrix}.$$

Matrix multiplication sums row-column products. The Hadamard product multiplies matching entries. They are not interchangeable. [17][19]

## 4. A Root Versus All Solutions

The principal real root is a function:

$$\sqrt{25} = 5.$$

Solving an equation produces a solution set:

$$\{x \in \mathbb{R} : x^2 = 25\} = \{-5, 5\}.$$

The symbol  $\sqrt{25}$  does not mean both values at once. [4]

## 5. Modular Inversion

Modulo seven,  $3 \cdot 5 = 15 \equiv 1$ , so the inverse of [3] is [5]. Modulo six, [2] has no inverse: all products  $2k$  remain even modulo six and never equal [1]. Nonzero does not always mean invertible. [1][8]

## 6. Power Versus Power Tower

Exponentiation gives  $2^3 = 8$ . Integer tetration gives

$$2 \uparrow\uparrow 3 = 2^{(2^2)} = 16.$$

The tower is evaluated from the top down. Extending its height from integers to real or complex numbers requires more than simply writing the same symbols. [9][10]

## 7. Differentiation Versus Integration

For  $f(x) = x^2$ ,

$$Df = 2x, \quad \int_0^1 f(x) dx = \frac{1}{3}.$$

The derivative is a function. The definite integral here is a number. Antidifferentiation instead produces  $x^3/3 + C$ , with a constant or normalization needed to select one function. [23]

## 8. Two Fractional Derivatives

For  $0 < \alpha < 1$  and lower endpoint zero,

$$D_{RL}^\alpha 1 = \frac{x^{-\alpha}}{\Gamma(1-\alpha)}, \quad D_C^\alpha 1 = 0 \quad (x > 0).$$

Both are legitimate definitions with different rules. Calling an operator “the fractional derivative” without specifying which one can be ambiguous. [32]

## 9. Changing the Arithmetic

Under the min-plus tropical convention,

$$3 \oplus 5 = \min(3, 5) = 3, \quad 3 \otimes 5 = 3 + 5 = 8.$$

Ordinary addition and multiplication have not become false. The symbols have been assigned different operations in a specified structure. [46]

## 10. Value Versus Minimizer

For  $f(x) = (x - 3)^2 + 2$  on  $\mathbb{R}$ ,

$$\min_x f(x) = 2, \quad \operatorname{argmin}_x f(x) = \{3\}.$$

The first returns the optimal value; the second returns the set of points attaining it. [39]

# Numbered Catalog

Read each entry as a short definition plus a representative formula or example. Domain restrictions in the text remain part of the definition. Source numbers at chapter openings link to the reference section.

## 1. Arithmetic and Basic Scalar Operations

ENTRIES 1–18

These act on numbers. Special cases such as squaring are included for lookup, not counted as new algebraic primitives. [1] [2]

### 1. Addition [OP]

Combines two numbers into their sum. On the integers, the result remains an integer.

$$7 + 5 = 12$$

### 2. Subtraction [OP]

Takes the additive difference of two numbers by adding the opposite of the second number.

$$a - b = a + (-b)$$

### 3. Multiplication [OP]

Produces a product. With a nonnegative integer multiplier it agrees with repeated addition; other number systems extend the rule.

$$4 \times 3 = 12$$

### 4. Division [OP]

Solves multiplication by a nonzero divisor in a field. Ordinary real and complex division excludes a zero denominator.

$$a/b = c \iff bc = a \quad (b \neq 0)$$

### 5. Additive Negation [OP]

Sends a number to its additive inverse, the number that adds to it to give zero.

$$-7 + 7 = 0$$

### 6. Reciprocation [OP]

Sends a nonzero number to its multiplicative inverse. Their product is one.

$$5^{-1} = 1/5$$

### 7. Successor [OP]

Returns the next integer or natural number by adding one. Ordinal successor is a related but distinct construction.

$$S(n) = n + 1$$

### 8. Predecessor [OP]

Subtracts one. It is total on the integers but needs a restriction or convention at zero on the nonnegative integers.

$$P(8) = 7$$

### 9. Absolute Value [FUN]

Returns a real number's distance from zero, ignoring its sign. Complex modulus generalizes it.

$$|-9| = 9$$

### 10. Exponentiation [OP]

Raises a base to a power. Domains and branches depend on the number system and exponent.

$$2^5 = 32$$

### 11. Squaring [OP]

Multiplies a number by itself, the exponentiation special case with exponent two.

$$(-4)^2 = 16$$

### 12. Cubing [OP]

Multiplies three copies of a number, the exponentiation special case with exponent three.

$$(-2)^3 = -8$$

### 13. Principal Square Root [FUN]

Returns the nonnegative real number whose square is the nonnegative real input.

$$\sqrt{49} = 7$$

### 14. Real Cube Root [FUN]

Returns the unique real number whose cube is the input. Negative inputs are allowed.

$$\sqrt[3]{-27} = -3$$

### 15. Real nth Root [FUN]

For a positive integer order  $n$ , inverts the power  $x \mapsto x^n$ . Even order selects the nonnegative root of a nonnegative input; odd order allows every real input.

$$\sqrt[5]{32} = 2$$

### 16. Percentage Scaling [OP]

Applies a factor of one hundredth per percent. This is derived multiplication, not a new primitive operation.

$$15\% \text{ of } 80 = (15/100)80 = 12$$

**17. Ratio Formation [OP]**

Forms a quotient comparing quantities, with nonzero second quantity. Units and scale must be specified when relevant.

$$a : b \longleftrightarrow a/b \quad (b \neq 0)$$

**18. Weighted Combination [OP]**

Adds inputs after multiplying them by specified weights. Weights summing to one give an affine combination.

$$0.3(10) + 0.7(20) = 17$$

---

**2. Elementary and Special Function Evaluations**

ENTRIES 19–38

Evaluating named functions is included in the broad sense of operation. Trigonometric angles use radians; inverse functions require selected ranges. [3] [4] [5] [6]

**19. Exponential Evaluation [FUN]**

Evaluates the natural exponential. A fixed positive base also defines an exponential function.

$$\exp(x) = e^x, \quad e^0 = 1$$

**20. Logarithm Evaluation [FUN]**

Returns the exponent producing a positive real input from a positive base other than one.

$$\log_2 8 = 3$$

**21. Sine Evaluation [FUN]**

Returns the vertical coordinate of the point at the given angle on the unit circle.

$$\sin(\pi/2) = 1$$

**22. Cosine Evaluation [FUN]**

Returns the horizontal coordinate of the point at the given angle on the unit circle.

$$\cos(\pi) = -1$$

**23. Tangent Evaluation [FUN]**

Divides sine by cosine wherever cosine is nonzero.

$$\tan x = \sin x / \cos x$$

**24. Cotangent Evaluation [FUN]**

Divides cosine by sine wherever sine is nonzero.

$$\cot x = \cos x / \sin x$$

**25. Secant Evaluation [FUN]**

Takes the reciprocal of cosine wherever cosine is nonzero.

$$\sec x = 1/\cos x$$

**26. Cosecant Evaluation [FUN]**

Takes the reciprocal of sine wherever sine is nonzero.

$$\csc x = 1/\sin x$$

**27. Arcsine Evaluation [FUN]**

Returns the angle in  $[-\pi/2, \pi/2]$  whose sine equals the input in  $[-1, 1]$ .

$$\arcsin(1/2) = \pi/6$$

**28. Arccosine Evaluation [FUN]**

Returns the angle in  $[0, \pi]$  whose cosine equals the input in  $[-1, 1]$ .

$$\arccos(1/2) = \pi/3$$

**29. Arctangent Evaluation [FUN]**

Returns the angle in  $(-\pi/2, \pi/2)$  whose tangent equals the real input.

$$\arctan(1) = \pi/4$$

**30. Reciprocal-Trigonometric Inversion [FUN]**

Evaluates arcsecant, arccosecant, or arccotangent on a selected one-to-one branch. Range conventions vary.

$$\operatorname{arcsec}(2) = \pi/3 \text{ on a usual branch}$$

**31. Hyperbolic Sine [FUN]**

Takes half the difference between opposite exponentials; it differs from ordinary sine.

$$\sinh x = (e^x - e^{-x})/2$$

**32. Hyperbolic Cosine [FUN]**

Takes half the sum of opposite exponentials.

$$\cosh x = (e^x + e^{-x})/2$$

**33. Hyperbolic Tangent [FUN]**

Divides hyperbolic sine by hyperbolic cosine. Its real values lie strictly between minus one and one.

$$\tanh x = \sinh x / \cosh x$$

**34. Reciprocal Hyperbolic Evaluation [FUN]**

Evaluates hyperbolic secant, cosecant, or cotangent by reciprocation or division, excluding zero denominators.

$$\operatorname{sech} x = 1/\cosh x$$

**35. Inverse Hyperbolic Evaluation [FUN]**

Inverts hyperbolic functions on specified domains. Real inverse hyperbolic sine is defined on all real numbers.

$$\operatorname{arsinh} x = \ln(x + \sqrt{x^2 + 1})$$

**36. Gamma Function Evaluation [FUN]**

Evaluates a factorial-related function extending beyond integer inputs, with poles at nonpositive integers.

$$\Gamma(n+1) = n! \quad (n \geq 0)$$

**37. Beta Function Evaluation [FUN]**

Evaluates an integral pairing positive real parameters, equivalently a quotient of gamma functions.

$$B(a, b) = \Gamma(a)\Gamma(b)/\Gamma(a+b)$$

**38. Lambert W Evaluation [FUN]**

Inverts  $w \mapsto we^w$ . Different real or complex branches must be distinguished.

$$W(z)e^{W(z)} = z$$

**3. Rounding, Integer, and Counting Operations**

ENTRIES 39–54

Rounding conventions and nonnegative-integer restrictions matter. Counting functions are not extra primitive arithmetic operations. [6] [7] [8]

**39. Floor [FUN]**

Returns the greatest integer less than or equal to a real input.

$$\lfloor -2.3 \rfloor = -3$$

**40. Ceiling [FUN]**

Returns the least integer greater than or equal to a real input.

$$\lceil -2.3 \rceil = -2$$

**41. Truncation Toward Zero [FUN]**

Removes the fractional part by moving toward zero, unlike floor for negative nonintegers.

$$\text{trunc}(-2.8) = -2$$

**42. Rounding to Nearest [FUN]**

Selects a nearest permitted value at a specified precision. A tie-breaking rule is part of the definition.

$$\text{round}(2.6) = 3$$

**43. Fractional Part [FUN]**

Subtracts the floor, giving a value in  $[0, 1)$  for every real input.

$$\{-2.3\} = -2.3 - (-3) = 0.7$$

**44. Sign Function [FUN]**

Records whether a real input is negative, zero, or positive.

$$\text{sgn}(x) \in \{-1, 0, 1\}$$

**45. Factorial [FUN]**

Multiplies positive integers up to a nonnegative integer input; the empty product defines zero factorial.

$$5! = 120, \quad 0! = 1$$

**46. Double Factorial [FUN]**

Multiplies every second integer down to one or two. It does not mean applying factorial twice.

$$7!! = 7 \cdot 5 \cdot 3 \cdot 1 = 105$$

**47. Rising Factorial [FUN]**

Multiplies a specified number of successive factors increasing by one.

$$x^{\overline{n}} = x(x+1) \cdots (x+n-1)$$

**48. Falling Factorial [FUN]**

Multiplies a specified number of successive factors decreasing by one.

$$x^{\underline{n}} = x(x-1) \cdots (x-n+1)$$

**49. Binomial Coefficient [FUN]**

Counts unordered selections of  $k$  objects from  $n$  distinct objects, for  $0 \leq k \leq n$ .

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

**50. Multinomial Coefficient [FUN]**

Counts divisions of distinct objects into labeled groups of prescribed sizes adding to  $n$ .

$$\binom{n}{n_1, \dots, n_r} = \frac{n!}{\prod_i n_i!}$$

**51. Greatest Common Divisor [OP]**

Returns the greatest positive common divisor of integers not both zero. Conventionally,  $\text{gcd}(0, 0) = 0$ .

$$\text{gcd}(18, 30) = 6$$

**52. Least Common Multiple [OP]**

Returns the least positive common multiple of nonzero integers; conventionally zero when an input is zero.

$$\text{lcm}(6, 8) = 24$$

**53. Divisor Sum [FUN]**

Adds powers of the positive divisors of a positive integer. The exponent selects a member of the family.

$$\sigma_k(n) = \sum_{d|n} d^k$$

**54. Euler Totient [FUN]**

Counts integers from one through  $n$  relatively prime to the positive integer  $n$ .

$$\varphi(12) = 4$$

## 4. Modular and Arithmetic-Function Operations

ENTRIES 55–68

Congruence is a relation; these operations work with residues, prime-power information, or arithmetic functions. [1] [8]

### 55. Remainder Modulo a Positive Integer [OP]

Returns the unique remainder from zero through  $m - 1$  for integer input and positive integer modulus.

$$17 \bmod 5 = 2$$

### 56. Integer Quotient [OP]

Returns the quotient in Euclidean division with a positive integer divisor.

$$a = mq + r, \quad 0 \leq r < m$$

### 57. Modular Addition [OP]

Adds residue classes and reduces modulo the fixed modulus.

$$[4] + [3] = [2] \bmod 5$$

### 58. Modular Subtraction [OP]

Subtracts residue classes and reduces modulo the fixed modulus.

$$[1] - [4] = [2] \bmod 5$$

### 59. Modular Multiplication [OP]

Multiplies residue classes and reduces modulo the fixed modulus.

$$[4][3] = [2] \bmod 5$$

### 60. Modular Exponentiation [OP]

Raises a residue to a nonnegative integer power and reduces modulo the modulus.

$$3^4 \bmod 7 = 4$$

### 61. Modular Inversion [OP]

Finds a multiplicative inverse modulo  $m$ , existing exactly for inputs relatively prime to  $m$ .

$$3^{-1} = 5 \bmod 7$$

### 62. Modular Division by a Unit [OP]

Multiplies by a modular inverse. Nonzero residues need not be invertible for composite moduli.

$$[a]/[b] = [a][b]^{-1}$$

### 63. Prime-Adic Valuation [FUN]

Measures the exponent of a chosen prime in a nonzero rational number's prime factorization.

$$v_2(40) = 3$$

### 64. Legendre Symbol [FUN]

For an odd prime, reports whether an integer is zero modulo the prime, a nonzero square, or a nonsquare.

$$\left(\frac{2}{7}\right) = 1$$

### 65. Jacobi Symbol [FUN]

Extends the Legendre-symbol product to positive odd composite denominators. Value one need not imply a square residue.

$$\left(\frac{a}{mn}\right) = \left(\frac{a}{m}\right) \left(\frac{a}{n}\right)$$

### 66. Dirichlet Convolution [OP]

Combines arithmetic functions by summing products over divisor pairs.

$$(f * g)(n) = \sum_{d|n} f(d)g(n/d)$$

### 67. Dirichlet Inversion [OPER]

Finds a convolution inverse of an arithmetic function with nonzero value at one;  $\varepsilon(1) = 1$  and otherwise  $\varepsilon(n) = 0$ .

$$f * f^{-1} = \varepsilon$$

### 68. Mobius Inversion [OPER]

Recovers a function from its divisor sums by convolution with the number-theoretic Mobius function.

$$F(n) = \sum_{d|n} f(d) \Rightarrow f = \mu * F$$

## 5. Hyperoperations and Iterated Powers

ENTRIES 69–78

Integer-height hyperoperations are exact recursive definitions. Real-height extensions and inverses require additional domains, branches, or regularity conditions. [9] [10]

### 69. Repeated Function Iteration [OPER]

Composes a function with itself an integer number of times; this differs from pointwise exponentiation.

$$f^{\circ 3}(x) = f(f(f(x)))$$

### 70. Tetration [OP]

Builds a right-associated power tower of nonnegative integer height; height zero is conventionally one.

$$2 \uparrow\uparrow 3 = 2^{(2^2)} = 16$$

### 71. Pentation [OP]

Iterates tetration using a specified hyperoperation convention. For positive bases a standard recurrence starts at one for height zero.

$$a \uparrow\uparrow\uparrow (n+1) = a \uparrow\uparrow (a \uparrow\uparrow\uparrow n)$$

**72. Hexation [OP]**

Iterates pentation, one level higher in the same recursively specified hierarchy. The superscript here counts arrows.

$$a \uparrow^4 (n + 1) = a \uparrow^3 (a \uparrow^4 n)$$

**73. Higher Knuth-Arrow Operation [OP]**

Uses a positive integer arrow count to specify exponentiation and its successively iterated analogues.

$$a \uparrow^1 b = a^b$$

**74. Super-Root [FUN]**

Inverts a fixed-height power tower with respect to its base on a domain where the base is uniquely determined.

$$\text{sroot}_2(27) = 3 \quad (3^3 = 27)$$

**75. Super-Logarithm [FUN]**

Inverts a chosen tetration with respect to height. Continuous values require a selected continuous tetration extension.

$$\text{slog}_2(16) = 3$$

**76. Continuous Tetration [FUN]**

Extends integer-height towers using additional conditions. The functional equation alone does not select a unique global function.

$$T(t + 1) = a^{T(t)}, \quad T(0) = 1$$

**77. Fractional Function Iteration [OPER]**

Defines noninteger iterates in a specified theory, often through conjugacy. Existence and composition laws require hypotheses.

$$f^{\circ s} \circ f^{\circ t} = f^{\circ(s+t)}$$

**78. Infinite Power-Tower Limit [OPER]**

Takes the limit of finite towers when they converge. Satisfying a fixed-point equation alone is not a convergence proof.

$$L = \lim_{n \rightarrow \infty} a \uparrow\uparrow n \Rightarrow L = a^L$$

---

**6. Complex and Hypercomplex Operations**

ENTRIES 79–92

Branch choices, quaternion ordering, and algebra-specific rules cannot be silently replaced by ordinary real-number rules. [11] [12]

**79. Complex Conjugation [OP]**

Reverses the imaginary part while preserving the real part.

$$\overline{a + bi} = a - bi$$

**80. Real-Part Extraction [FUN]**

Returns a complex number's real coordinate.

$$\text{Re}(3 - 4i) = 3$$

**81. Imaginary-Part Extraction [FUN]**

Returns the real coefficient of the imaginary unit, not the imaginary term itself.

$$\text{Im}(3 - 4i) = -4$$

**82. Complex Modulus [FUN]**

Returns a complex number's Euclidean distance from zero.

$$|a + bi| = \sqrt{a^2 + b^2}$$

**83. Argument Extraction [FUN]**

Returns a direction angle for a nonzero complex number. A principal argument requires a selected interval.

$$\text{Arg}(i) = \pi/2$$

**84. Complex Exponential [FUN]**

Evaluates the exponential on complex inputs, combining scaling with rotation.

$$e^{a+bi} = e^a (\cos b + i \sin b)$$

**85. Complex Logarithm [FUN]**

Recovers logarithms of a nonzero complex number. A chosen branch gives a single-valued function on an appropriate domain.

$$\log z = \ln |z| + i(\theta + 2\pi k), \quad k \in \mathbb{Z}$$

**86. Complex Power [OP]**

Defines powers using a selected logarithm branch; changing the branch can change noninteger powers.

$$z^w = \exp(w \text{Log } z)$$

**87. Complex Root-Set Formation [CON]**

Returns all complex solutions of a positive integer-power equation. Nonzero input has exactly  $n$  distinct  $n$ th roots.

$$\{w : w^n = z\} \quad (n \geq 1)$$

**88. Quaternion Multiplication [OP]**

Multiplies in the quaternion algebra, where the basic imaginary units do not commute.

$$ij = k, \quad ji = -k$$

**89. Quaternion Conjugation [OP]**

Negates all three imaginary components of a quaternion.

$$\overline{a + bi + cj + dk} = a - bi - cj - dk$$

**90. Quaternion Inversion [OP]**

Inverts a nonzero quaternion using conjugation and squared norm.

$$q^{-1} = \bar{q}/|q|^2$$

**91. Quaternion Rotation Action [OP]**

Rotates a vector represented by a pure imaginary quaternion using a unit quaternion.

$$v \mapsto qvq^{-1} \quad (|q| = 1)$$

**92. Octonion Multiplication [OP]**

Uses an octonion multiplication table. Multiplication is not generally commutative or associative, so parentheses matter.

$$(ab)c \text{ need not equal } a(bc)$$

**7. Set Operations and Set Constructions**

ENTRIES 93–108

Complements require a universe. Some constructions create a new kind of set rather than staying inside the original set. [1] [13] [26]

**93. Union [OP]**

Collects elements in at least one of two sets.

$$\{1, 2\} \cup \{2, 3\} = \{1, 2, 3\}$$

**94. Intersection [OP]**

Collects elements common to both sets.

$$\{1, 2\} \cap \{2, 3\} = \{2\}$$

**95. Relative Difference [OP]**

Keeps elements in the first set but not the second.

$$\{1, 2\} \setminus \{2, 3\} = \{1\}$$

**96. Complement [OP]**

Removes a set from a specified universe. Changing the universe changes the complement.

$$A^c = U \setminus A$$

**97. Symmetric Difference [OP]**

Keeps elements belonging to exactly one of two sets.

$$A \Delta B = (A \setminus B) \cup (B \setminus A)$$

**98. Cartesian Product [CON]**

Forms ordered pairs with components from the two input sets.

$$A \times B = \{(a, b) : a \in A, b \in B\}$$

**99. Disjoint Union [CON]**

Combines tagged copies of sets so matching elements from different inputs remain distinct.

$$A \sqcup B = (A \times \{0\}) \cup (B \times \{1\})$$

**100. Power-Set Formation [CON]**

Forms the set of all subsets, including the empty set and the original set.

$$\mathcal{P}(\{a\}) = \{\emptyset, \{a\}\}$$

**101. Indexed Union [OP]**

Collects elements belonging to any member of an indexed family of sets.

$$\bigcup_{i \in I} A_i$$

**102. Indexed Intersection [OP]**

Collects elements belonging to every member of an indexed family. An empty family requires an ambient universe convention.

$$\bigcap_{i \in I} A_i$$

**103. Direct Image [OP]**

Applies a function to every element of a subset and collects the resulting values.

$$f[A] = \{f(a) : a \in A\}$$

**104. Inverse Image [OP]**

Collects domain elements whose images lie in a target subset. The function need not be invertible.

$$f^{-1}[B] = \{x : f(x) \in B\}$$

**105. Quotient-Set Formation [CON]**

Groups elements into equivalence classes under a specified equivalence relation.

$$X/\sim = \{[x] : x \in X\}$$

**106. Set Cardinality [FUN]**

Returns finite set size or, in the usual framework, the cardinal number of an infinite set.

$$|\{a, b, c\}| = 3$$

**107. Set Partition Formation [CON]**

Divides a set into nonempty disjoint blocks with union equal to the whole set. A partition rule must be specified.

$$X = \bigsqcup_{i \in I} B_i$$

**108. Generated Sigma-Algebra [CON]**

Forms the smallest sigma-algebra containing a specified collection of subsets of a fixed universe.

$$\sigma(\mathcal{A}) = \bigcap \{\mathcal{F} : \mathcal{F} \supseteq \mathcal{A}, \mathcal{F} \text{ a sigma-algebra}\}$$

## 8. Logical and Order-Theoretic Operations

ENTRIES 109–124

Connectives act on truth values or formulas. Equality and inequality are relations; returning their truth values is predicate evaluation. [1] [14]

### 109. Logical Negation [OP]

Reverses a truth value in classical two-valued logic, unlike numerical additive negation.

$$\neg \text{True} = \text{False}$$

### 110. Conjunction [OP]

Returns true exactly when both statements are true.

$$p \wedge q$$

### 111. Disjunction [OP]

Returns true when at least one statement is true. Ordinary mathematical OR is inclusive.

$$p \vee q$$

### 112. Exclusive OR [OP]

Returns true exactly when the truth values differ.

$$p \oplus q = (p \vee q) \wedge \neg(p \wedge q)$$

### 113. Material Implication [OP]

Is false exactly when the first statement is true and the second is false.

$$p \Rightarrow q \equiv \neg p \vee q$$

### 114. Logical Biconditional [OP]

Returns true exactly when the two truth values agree.

$$p \Leftrightarrow q$$

### 115. NAND [OP]

Negates conjunction, becoming false only when both inputs are true.

$$\text{NAND}(p, q) = \neg(p \wedge q)$$

### 116. NOR [OP]

Negates disjunction, becoming true only when both inputs are false.

$$\text{NOR}(p, q) = \neg(p \vee q)$$

### 117. Universal Quantification [OPER]

Binds a predicate variable and asserts that every element of the specified domain satisfies the predicate.

$$\forall x \in D \, P(x)$$

### 118. Existential Quantification [OPER]

Binds a variable and asserts that at least one element of the specified domain satisfies the predicate.

$$\exists x \in D \, P(x)$$

### 119. Minimum [OP]

Selects the smaller of comparable values or the smallest value of a finite list.

$$\min(4, 9) = 4$$

### 120. Maximum [OP]

Selects the larger of comparable values or the largest value of a finite list.

$$\max(4, 9) = 9$$

### 121. Infimum [FUN]

Returns the greatest lower bound of a set when it exists in the specified ordered system.

$$\inf(0, 1) = 0$$

### 122. Supremum [FUN]

Returns the least upper bound of a set when it exists in the specified ordered system.

$$\sup(0, 1) = 1$$

### 123. Lattice Meet [OP]

Returns the greatest element below both inputs. For subsets ordered by inclusion it is intersection.

$$A \wedge B = A \cap B$$

### 124. Lattice Join [OP]

Returns the least element above both inputs. For subsets ordered by inclusion it is union.

$$A \vee B = A \cup B$$

## 9. Function and Relation Operations

ENTRIES 125–142

Types distinguish pointwise operations, composition, and transformations of entire functions. A converse relation need not be an inverse function. [3] [13] [15] [43]

### 125. Function Evaluation [OP]

Applies a function to an allowed input and returns its uniquely assigned output.

$$f(x) = x^2 \Rightarrow f(3) = 9$$

### 126. Function Composition [OP]

Feeds one function's output into another, with compatible domains and codomains.

$$(f \circ g)(x) = f(g(x))$$

### 127. Pointwise Function Addition [OP]

Adds function values at the same input.

$$(f + g)(x) = f(x) + g(x)$$

### 128. Pointwise Function Multiplication [OP]

Multiplies function values at the same input rather than composing the functions.

$$(fg)(x) = f(x)g(x)$$

**129. Pointwise Function Division [OP]**

Divides values where the denominator function is nonzero.

$$(f/g)(x) = f(x)/g(x)$$

**130. Function Inversion [OPER]**

Reverses a bijection. A noninjective function first needs a suitable domain restriction.

$$f^{-1}(f(x)) = x$$

**131. Function Restriction [OPER]**

Keeps the rule but uses only a specified subset of a function's domain.

$$f|_A : A \rightarrow Y$$

**132. Zero Extension [OPER]**

Extends a function by assigning zero outside its original domain. This may destroy continuity or differentiability.

$$\tilde{f}(x) = 0 \quad (x \notin A)$$

**133. Currying [OPER]**

Converts a function on pairs into a function returning functions.

$$f(x, y) \longleftrightarrow [x \mapsto (y \mapsto f(x, y))]$$

**134. Uncurrying [OPER]**

Converts a function returning functions into one function on pairs.

$$g \mapsto [(x, y) \mapsto g(x)(y)]$$

**135. Partial Application [OPER]**

Fixes selected inputs of a multi-input function, leaving the others variable.

$$f(a, \cdot) : y \mapsto f(a, y)$$

**136. Argument Permutation [OPER]**

Reorders function inputs by a specified permutation.

$$g(x, y) = f(y, x)$$

**137. Pullback of Functions [OPER]**

Composes a target-space function with a given map into that space.

$$f^*h = h \circ f$$

**138. Relation Composition [OP]**

Connects endpoints when an intermediate element links them through the two input relations.

$$S \circ R = \{(x, z) : \exists y, xRy \text{ and } ySz\}$$

**139. Converse Relation [OP]**

Reverses every ordered pair in a relation without necessarily producing a function.

$$R^{-1} = \{(y, x) : (x, y) \in R\}$$

**140. Transitive Closure [OPER]**

Adds every pair connected by a positive finite relation chain. Adding identity pairs gives reflexive-transitive closure.

$$R^+ = \bigcup_{n \geq 1} R^n$$

**141. Functional Substitution [OPER]**

Replaces an input variable or expression with another permitted expression, respecting domains and variable binding.

$$f(x)|_{x=t+1} = f(t+1)$$

**142. Functional Conjugation [OPER]**

Changes a dynamical coordinate system through an invertible map. It differs from complex conjugation.

$$g = h^{-1} \circ f \circ h$$

**10. Vector and Coordinate Operations**

ENTRIES 143–158

Vector operations may return scalars, vectors, or tensors. The three-dimensional cross product does not extend unchanged to every dimension. [16] [17]

**143. Vector Addition [OP]**

Adds corresponding components of vectors in the same space.

$$(1, 2) + (3, 4) = (4, 6)$$

**144. Vector Subtraction [OP]**

Subtracts matching components to form a displacement vector.

$$(3, 4) - (1, 2) = (2, 2)$$

**145. Scalar Multiplication [OP]**

Multiplies a vector by a scalar from its field; the input types differ.

$$3(1, 2) = (3, 6)$$

**146. Dot Product [OP]**

Sums pairwise products of real coordinates and returns a scalar.

$$(1, 2) \cdot (3, 4) = 11$$

**147. Hermitian Inner Product [OP]**

Pairs complex vectors with conjugation in one slot. This book conjugates the first slot.

$$\langle u, v \rangle = \sum_i \overline{u_i} v_i$$

**148. Cross Product** [OP]

In oriented Euclidean three-space, returns a perpendicular vector whose length equals parallelogram area.

$$e_1 \times e_2 = e_3$$

**149. Outer Product** [OP]

Builds a matrix or tensor from component pairs without summing them.

$$(uv^T)_{ij} = u_i v_j$$

**150. Vector Norm** [FUN]

Returns size under a chosen norm, not necessarily Euclidean length.

$$\|v\|_2 = \sqrt{\sum_i |v_i|^2}$$

**151. Vector Normalization** [OPER]

Divides a nonzero vector by its norm to obtain a unit vector.

$$\hat{v} = v / \|v\|$$

**152. Orthogonal Projection onto a Line** [OP]

Extracts the component along a specified nonzero real vector.

$$\text{proj}_u v = \frac{v \cdot u}{u \cdot u} u$$

**153. Vector Rejection** [OP]

Removes a vector's projection onto a direction, leaving its perpendicular component.

$$\text{rej}_u v = v - \text{proj}_u v$$

**154. Scalar Triple Product** [OP]

Returns the oriented volume associated with three vectors.

$$u \cdot (v \times w)$$

**155. Vector Triple Product** [OP]

Takes a cross product with another cross product; parentheses are essential.

$$u \times (v \times w) = v(u \cdot w) - w(u \cdot v)$$

**156. Coordinate Change** [OPER]

Reexpresses the same vector in a new basis using an invertible change-of-basis matrix.

$$[v]_{\text{new}} = P^{-1}[v]_{\text{old}}$$

**157. Linear-Span Formation** [CON]

Forms all finite linear combinations of a given collection of vectors.

$$\text{span}\{v_i\} = \left\{ \sum_i a_i v_i \right\}$$

**158. Orthogonal-Complement Formation** [CON]

Collects vectors perpendicular to every vector in a specified subspace.

$$W^\perp = \{v : \langle w, v \rangle = 0 \forall w \in W\}$$

**11. Matrix Operations and Matrix Functions**  
ENTRIES 159–180

Matrix sizes must be compatible. Matrix multiplication and entrywise multiplication are different operations, and matrix inverses need not exist. [17] [18] [19]

**159. Matrix Addition** [OP]

Adds corresponding entries of matrices of the same size.

$$(A + B)_{ij} = A_{ij} + B_{ij}$$

**160. Matrix Multiplication** [OP]

Combines rows of the first matrix with columns of the second. The inner dimensions must match.

$$(AB)_{ij} = \sum_k A_{ik} B_{kj}$$

**161. Matrix-Vector Application** [OP]

Applies a linear transformation represented by a matrix to a compatible coordinate vector.

$$y = Ax$$

**162. Matrix Transpose** [OP]

Exchanges row and column indices without conjugating entries.

$$(A^T)_{ij} = A_{ji}$$

**163. Conjugate Transpose** [OP]

Transposes a complex matrix and conjugates each entry; also called the Hermitian adjoint.

$$A^* = \overline{A}^T$$

**164. Matrix Inversion** [OP]

Finds the matrix reversing a square invertible matrix under multiplication. Singular matrices have no such inverse.

$$AA^{-1} = A^{-1}A = I$$

**165. Determinant** [FUN]

Returns a scalar encoding oriented volume scaling for a square matrix, and detects invertibility over a field.

$$\det \begin{pmatrix} a & b \\ c & d \end{pmatrix} = ad - bc$$

**166. Matrix Trace** [FUN]

Adds the diagonal entries of a square matrix. It is distinct from determinant and rank.

$$\text{tr}(A) = \sum_i A_{ii}$$

**167. Matrix Rank** [FUN]

Returns the dimension of the column space, equivalently the row space. It measures the number of independent directions.

$$\text{rank}(I_n) = n$$

**168. Hadamard Product** [OP]

Multiplies matching entries of equal-size matrices rather than summing row-column products.

$$(A \odot B)_{ij} = A_{ij}B_{ij}$$

**169. Kronecker Product** [OP]

Replaces each entry of the first matrix with that entry times a copy of the second matrix.

$$A \otimes B = [a_{ij}B]$$

**170. Matrix Direct Sum** [CON]

Places matrices in separate diagonal blocks, with zeros in the off-diagonal blocks.

$$A \oplus B = \begin{pmatrix} A & 0 \\ 0 & B \end{pmatrix}$$

**171. Matrix Power** [OP]

Multiplies a square matrix by itself a nonnegative integer number of times; the zeroth power is the identity.

$$A^0 = I, \quad A^3 = AAA$$

**172. Matrix Exponential** [FUN]

Defines an exponential of a square matrix through a convergent power series. It is not entrywise exponentiation.

$$e^A = \sum_{k=0}^{\infty} A^k / k!$$

**173. Matrix Logarithm** [FUN]

Finds a matrix whose exponential is the input. Existence and uniqueness depend on the field and branch convention.

$$e^L = A$$

**174. Matrix Square Root** [FUN]

Finds a matrix whose square equals the input. A positive semidefinite Hermitian matrix has a unique positive semidefinite square root.

$$B^2 = A$$

**175. Moore-Penrose Pseudoinverse** [FUN]

Returns the uniquely characterized generalized inverse satisfying four Penrose equations, even for rectangular or singular matrices.

$$A^+ = V\Sigma^+U^* \quad (A = U\Sigma V^*)$$

**176. Matrix Commutator** [OP]

Measures the difference between the two multiplication orders in an associative algebra.

$$[A, B] = AB - BA$$

**177. Anticommutator** [OP]

Adds products taken in opposite orders. It is important in operator algebras and mathematical physics.

$$\{A, B\} = AB + BA$$

**178. Similarity Transformation** [OPER]

Changes a square matrix's basis representation without changing its underlying linear map.

$$A \mapsto P^{-1}AP$$

**179. Schur Complement** [OP]

Eliminates an invertible block from a block matrix, leaving a reduced block expression.

$$D - CA^{-1}B$$

**180. Partial Trace** [OPER]

Traces over one factor of a tensor-product space, leaving an operator on the other factor.

$$\text{tr}_B(A \otimes B) = A \text{tr}(B)$$

---

**12. Tensor, Exterior, and Clifford Operations**

ENTRIES 181–192

Tensor operations depend on slot types, symmetry, and often a metric. Exterior products are antisymmetric, unlike ordinary scalar multiplication. [12] [20] [21]

**181. Tensor Product** [OP]

Combines multilinear data into a higher-order tensor while preserving the distinction between input slots.

$$(u \otimes v)_{ij} = u_i v_j$$

**182. Tensor Contraction** [OPER]

Pairs a covariant and contravariant slot and sums over matching indices, reducing tensor order by two.

$$T^{ik}{}_{jk} \mapsto S^i{}_j$$

**183. Index Permutation** [OPER]

Reorders tensor slots according to a specified permutation.

$$T_{ijk} \mapsto T_{kij}$$

**184. Symmetrization** [OPER]

Averages a tensor over selected slot permutations, producing a symmetric component under the stated normalization.

$$T_{(ij)} = (T_{ij} + T_{ji})/2$$

**185. Antisymmetrization [OPER]**

Takes a signed average over selected slot permutations, producing an alternating component.

$$T_{[ij]} = (T_{ij} - T_{ji})/2$$

**186. Exterior Product [OP]**

Combines alternating tensors or differential forms by the wedge product. Swapping factors introduces a degree-dependent sign. In the sign rule,  $p$  and  $q$  are the respective degrees.

$$\alpha \wedge \beta = (-1)^{pq} \beta \wedge \alpha$$

**187. Interior Product [OPER]**

Inserts a vector into the first slot of a differential form, reducing its degree by one.

$$(\iota_X \omega)(X_2, \dots) = \omega(X, X_2, \dots)$$

**188. Hodge Star [OPER]**

Uses a metric and orientation to turn a  $k$ -form into an  $(n-k)$ -form. Signs depend on conventions and signature.

$$* : \Lambda^k \rightarrow \Lambda^{n-k}$$

**189. Raising an Index [OPER]**

Uses the inverse metric to convert a covariant index into a contravariant one.

$$v^i = g^{ij} v_j$$

**190. Lowering an Index [OPER]**

Uses the metric to convert a contravariant index into a covariant one.

$$v_i = g_{ij} v^j$$

**191. Clifford Product [OP]**

Combines inner and exterior multiplication according to a chosen quadratic form. For vectors it includes scalar and bivector parts.

$$uv = u \cdot v + u \wedge v$$

**192. Clifford Reversion [OPER]**

Reverses the order of vector factors in a Clifford product, extending linearly to multivectors.

$$v_1 \widetilde{v_2 \cdots v_k} = v_k \cdots v_2 v_1$$

**13. Group, Ring, and Module Operations**

ENTRIES 193–212

Abstract algebra specifies operations by their domains and axioms. A quotient operation must be independent of the representative chosen. [1] [44]

**193. Group Multiplication [OP]**

Combines two group elements using an associative law with an identity and inverses. The operation need not be numerical multiplication.

$$(ab)c = a(bc)$$

**194. Group Inversion [OP]**

Assigns each group element the unique element that cancels it on both sides.

$$gg^{-1} = g^{-1}g = e$$

**195. Group Conjugation [OP]**

Changes an element by multiplying with a group element and its inverse on opposite sides.

$$h \mapsto ghg^{-1}$$

**196. Group Commutator [OP]**

Forms a group element detecting noncommutativity. This book chooses one of the two common inverse conventions.

$$[g, h] = ghg^{-1}h^{-1}$$

**197. Group Action [OP]**

Applies a group element to an element of a set while respecting the group law and identity.

$$(gh) \cdot x = g \cdot (h \cdot x)$$

**198. Quotient-Group Multiplication [OP]**

Multiplies cosets of a normal subgroup. Normality makes the result independent of representatives.

$$(gN)(hN) = (gh)N$$

**199. Subgroup Generation [CON]**

Forms the smallest subgroup containing a specified subset, equivalently all finite words in that subset and its inverses.

$$\langle S \rangle \leq G$$

**200. Group Direct Product [CON]**

Forms tuples of group elements and multiplies them componentwise.

$$(g, h)(g', h') = (gg', hh')$$

**201. Semidirect Product [CON]**

Combines two groups using a specified action of one on the other by automorphisms.

$$(n, h)(n', h') = (n\phi_h(n'), hh')$$

**202. Free Product [CON]**

Combines groups with no relations between factors beyond their existing internal relations; elements are represented by reduced alternating words.

$$G * H$$

**203. Ring Addition and Multiplication [OP]**

Uses two operations satisfying ring axioms, including distributivity. Multiplication need not be commutative, depending on the definition.

$$a(b + c) = ab + ac$$

**204. Module Scalar Action [OP]**

Multiplies a module element by a ring element, generalizing vector-space scalar multiplication beyond fields.

$$r \cdot (m + n) = rm + rn$$

**205. Ideal Sum [OP]**

Forms the ideal consisting of sums of elements from two given ideals.

$$I + J = \{i + j : i \in I, j \in J\}$$

**206. Ideal Product [OP]**

Forms finite sums of products of elements from two ideals, not merely the set of individual products.

$$IJ = \left\{ \sum_k i_k j_k \right\}$$

**207. Ideal Radical [OPER]**

Collects ring elements with some positive power in the ideal, in a commutative ring.

$$\sqrt{I} = \{r : r^n \in I \text{ for some } n \geq 1\}$$

**208. Localization [CON]**

Adjoins inverses to a specified multiplicatively closed collection, subject to the usual localization equivalence.

$$R \mapsto S^{-1}R$$

**209. Tensor Product of Modules [CON]**

Builds the module representing balanced bilinear maps over a specified ring, with the required left/right module structures.

$$(mr) \otimes n = m \otimes (rn)$$

**210. Lie Bracket [OP]**

Combines Lie-algebra elements by a bilinear alternating operation satisfying the Jacobi identity. Matrix commutators are a principal example.

$$[x, [y, z]] + [y, [z, x]] + [z, [x, y]] = 0$$

**211. Jordan Product [OP]**

Symmetrizes multiplication in an associative algebra over a field where two is invertible, giving a commutative product.

$$A \circ B = (AB + BA)/2$$

**212. Frobenius Map [OPER]**

Raises elements of a commutative ring of prime characteristic  $p$  to their  $p$ th power; it is a ring homomorphism.

$$F(x) = x^p$$

---

**14. Polynomials and Formal Power Series**

ENTRIES 213–228

The coefficient ring matters. A formal power series is manipulated by coefficients; it does not automatically represent a convergent numerical series. [2] [22]

**213. Polynomial Addition [OP]**

Adds coefficients of equal powers, treating missing coefficients as zero.

$$(x^2 + 2x) + (3x - 1) = x^2 + 5x - 1$$

**214. Polynomial Multiplication [OP]**

Multiplies every term pair and combines like powers; coefficients combine by convolution.

$$(x + 1)(x + 2) = x^2 + 3x + 2$$

**215. Polynomial Composition [OP]**

Substitutes one polynomial into another. It is different from polynomial multiplication.

$$(p \circ q)(x) = p(q(x))$$

**216. Polynomial Quotient and Remainder [OP]**

Over a field, divides by a nonzero polynomial to produce unique quotient and smaller-degree remainder.

$$p = qd + r, \quad \deg r < \deg d$$

**217. Polynomial Greatest Common Divisor [OP]**

Finds a greatest common divisor, usually normalized to be monic over a field.

$$\gcd(x^2 - 1, x^2 - x) = x - 1$$

**218. Polynomial Resultant [FUN]**

Returns an elimination expression that vanishes exactly when two nonconstant polynomials over a field share a root in its algebraic closure.

$$\text{Res}_x(p, q)$$

**219. Polynomial Discriminant [FUN]**

Returns a coefficient expression detecting repeated roots, with normalization determined by degree and leading coefficient.

$$\text{disc}(ax^2 + bx + c) = b^2 - 4ac$$

**220. Coefficient Extraction [OPER]**

Selects the coefficient of a specified power or monomial in a polynomial or formal series.

$$[x^3](2 + 5x^3) = 5$$

**221. Formal Differentiation [OPER]**

Differentiates powers coefficientwise without requiring analytic convergence.

$$D\left(\sum_{n \geq 0} a_n x^n\right) = \sum_{n \geq 1} n a_n x^{n-1}$$

**222. Formal Integration [OPER]**

Integrates coefficientwise when division by positive integers is allowed in the coefficient system, with a selected constant term.

$$\sum_{n \geq 0} a_n x^n \mapsto \sum_{n \geq 0} \frac{a_n x^{n+1}}{n+1}$$

**223. Cauchy Product [OP]**

Multiplies formal or convergent series by summing coefficient pairs whose indices add to the desired index.

$$c_n = \sum_{k=0}^n a_k b_{n-k}$$

**224. Hadamard Series Product [OP]**

Multiplies corresponding coefficients rather than convolving them.

$$\left(\sum a_n x^n\right) \odot \left(\sum b_n x^n\right) = \sum a_n b_n x^n$$

**225. Formal Reciprocal [OPER]**

Inverts a formal series under multiplication when its constant coefficient is invertible.

$$(1 - x)^{-1} = \sum_{n \geq 0} x^n$$

**226. Formal Series Composition [OP]**

Substitutes a series into another; zero constant term in the inner series ensures each output coefficient uses finitely many terms.

$$f(g(x)) \quad \text{when } g(0) = 0$$

**227. Series Reversion [OPER]**

Finds a compositional inverse of a formal series with zero constant term and invertible linear coefficient.

$$f(g(x)) = g(f(x)) = x$$

**228. Formal Logarithm and Exponential [OPER]**

Applies the formal log to a series starting at one, or formal exponential to one starting at zero, over suitable coefficients.

$$\log(1 + u) = \sum_{n \geq 1} (-1)^{n+1} u^n / n$$

**15. Differentiation and Local Calculus Operators**

ENTRIES 229–244

Derivatives are defined only under their relevant existence conditions. Several operators return arrays or fields rather than a single number. [23] [24] [25]

**229. Ordinary Differentiation [OPER]**

Produces the instantaneous rate-of-change function wherever the defining difference-quotient limit exists.

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$

**230. Higher Differentiation [OPER]**

Applies differentiation repeatedly when the required derivatives exist.

$$D^n f = \frac{d^n f}{dx^n}$$

**231. Partial Differentiation [OPER]**

Differentiates with respect to one coordinate while holding the remaining independent coordinates fixed.

$$\frac{\partial}{\partial x}(x^2 y) = 2xy$$

**232. Mixed Partial Differentiation [OPER]**

Differentiates in multiple coordinate directions. Changing the order requires appropriate hypotheses, such as continuity of the mixed derivatives.

$$\frac{\partial^2 f}{\partial y \partial x}$$

**233. Total Differential [OPER]**

Gives the linear first-order change of a differentiable scalar function in all input directions together.

$$df = \sum_i \frac{\partial f}{\partial x_i} dx_i$$

**234. Directional Differentiation [OPER]**

Measures change along a specified direction, often a unit vector, using a directional limit. The gradient formula assumes differentiability.

$$D_v f(x) = \nabla f(x) \cdot v$$

**235. Gradient** [OPER]

Converts a differentiable scalar field into its vector of first partial derivatives in Euclidean coordinates.

$$\nabla f = (\partial_1 f, \dots, \partial_n f)$$

**236. Divergence** [OPER]

Measures the local net outward flux density of a differentiable vector field.

$$\nabla \cdot F = \sum_i \partial_i F_i$$

**237. Curl** [OPER]

In oriented Euclidean three-space, measures a vector field's local circulation tendency through an axial vector.

$$\nabla \times F$$

**238. Laplacian** [OPER]

Adds second derivatives in orthonormal Euclidean coordinates; more general settings require a specified geometry.

$$\Delta f = \sum_i \partial_i^2 f$$

**239. Jacobian Matrix** [OPER]

Assembles all first partial derivatives of a vector-valued function into a matrix.

$$(J_f)_{ij} = \partial f_i / \partial x_j$$

**240. Hessian Matrix** [OPER]

Assembles a scalar function's second partial derivatives into a matrix.

$$(H_f)_{ij} = \partial_i \partial_j f$$

**241. Frechet Derivative** [OPER]

Returns a bounded linear map giving a function's first-order approximation between normed spaces, when the required error is small enough.

$$f(x+h) = f(x) + Df(x)h + o(\|h\|)$$

**242. Gateaux Directional Derivative** [OPER]

Takes a directional difference-quotient limit in a vector space. Mere existence in all directions does not guarantee Frechet differentiability.

$$\delta f(x; h) = \lim_{t \rightarrow 0} \frac{f(x+th) - f(x)}{t}$$

**243. Variational Derivative** [OPER]

Represents a functional's first variation as an integral pairing when such a representation exists.

$$\delta J = \int \frac{\delta J}{\delta u}(x) \delta u(x) dx$$

**244. Euler-Lagrange Operator** [OPER]

Converts an integrand depending on a function and its derivatives into the expression governing stationary variations.

$$E(L) = \frac{\partial L}{\partial u} - \frac{d}{dx} \frac{\partial L}{\partial u'}$$

**16. Integration and Accumulation**

ENTRIES 245–258

Integration can produce numbers, functions, or families of functions. Improper and generalized integrals require precise limiting conventions. [11] [23] [24] [26]

**245. Definite Riemann Integration** [OPER]

Accumulates a bounded function over an interval through a common limit of suitable Riemann sums, when the function is integrable.

$$\int_a^b f(x) dx$$

**246. Antidifferentiation** [OPER]

Finds functions whose derivative is the input. Without a normalization, the output is a family differing by constants on an interval.

$$\int 2x dx = x^2 + C$$

**247. Cumulative Integration** [OPER]

Produces a particular accumulation function using a fixed starting point and variable endpoint.

$$F(x) = \int_a^x f(t) dt$$

**248. Improper Integration** [OPER]

Defines an integral with unbounded domain or singular behavior by specified limits, requiring the necessary limits to converge.

$$\int_1^\infty x^{-2} dx = 1$$

**249. Double Integration** [OPER]

Accumulates a scalar function over a two-dimensional region using area measure.

$$\iint_D f(x, y) dA$$

**250. Triple Integration** [OPER]

Accumulates a scalar function over a three-dimensional region using volume measure.

$$\iiint_V f(x, y, z) dV$$

**251. Scalar Line Integration** [OPER]

Accumulates a scalar field along a curve with respect to arc length.

$$\int_C f \, ds$$

**252. Vector Line Integration** [OPER]

Accumulates the tangential component of a vector field along an oriented curve.

$$\int_C \mathbf{F} \cdot d\mathbf{r}$$

**253. Scalar Surface Integration** [OPER]

Accumulates a scalar field over a surface with respect to surface area.

$$\iint_S f \, dS$$

**254. Flux Integration** [OPER]

Accumulates the normal component of a vector field through an oriented surface.

$$\iint_S \mathbf{F} \cdot \mathbf{n} \, dS$$

**255. Contour Integration** [OPER]

Integrates a complex-valued function along a specified oriented complex-plane path.

$$\int_\gamma f(z) \, dz$$

**256. Lebesgue Integration** [OPER]

Integrates measurable functions with respect to a measure, including many functions not Riemann integrable.

$$\int_X f \, d\mu$$

**257. Stieltjes Integration** [OPER]

Integrates one function against increments of another; the Riemann-Stieltjes or Lebesgue-Stieltjes interpretation must be specified.

$$\int_a^b f \, dg$$

**258. Cauchy Principal-Value Integration** [OPER]

Takes a prescribed balanced limit around singularities or infinity. A principal value can exist when the ordinary improper integral does not.

$$\text{PV} \int_{-1}^1 \frac{dx}{x} = 0$$

**17. Integral and Discrete Transforms**

ENTRIES 259–274

Transforms change representations. Their kernels, signs, normalization constants, and convergence conditions are part of the definition. [27] [28] [29]

**259. Fourier Transform** [OPER]

Converts a suitable function into a frequency representation. This book uses angular frequency with the negative exponential sign.

$$\widehat{f}(\omega) = \int_{\mathbb{R}} f(t) e^{-i\omega t} \, dt$$

**260. Inverse Fourier Transform** [OPER]

Reconstructs a function from its transform under appropriate inversion hypotheses and the matching normalization.

$$f(t) = \frac{1}{2\pi} \int_{\mathbb{R}} \widehat{f}(\omega) e^{i\omega t} \, d\omega$$

**261. Laplace Transform** [OPER]

Uses exponential weighting to transform a function on nonnegative time into a function of a complex frequency parameter.

$$F(s) = \int_0^\infty e^{-st} f(t) \, dt$$

**262. Inverse Laplace Transform** [OPER]

Recovers a time-domain function from its Laplace transform using a valid inversion method and region of convergence.

$$f = \mathcal{L}^{-1} F$$

**263. Bilateral Laplace Transform** [OPER]

Extends Laplace integration over the entire real line, generally with a strip of convergence.

$$F(s) = \int_{-\infty}^\infty e^{-st} f(t) \, dt$$

**264. Mellin Transform** [OPER]

Transforms a positive-domain function using power weights, making it useful for scaling behavior.

$$\mathcal{M}f(s) = \int_0^\infty x^{s-1} f(x) \, dx$$

**265. Z-Transform** [OPER]

Transforms a discrete sequence into a function of a complex variable, with a specified summation range and convergence region.

$$X(z) = \sum_{n \in \mathbb{Z}} x_n z^{-n}$$

**266. Discrete Fourier Transform [OPER]**

Converts a finite sequence into finitely many frequency coefficients using roots of unity. The FFT is an algorithm for this transform.

$$X_k = \sum_{n=0}^{N-1} x_n e^{-2\pi i k n / N}$$

**267. Discrete Cosine Transform [OPER]**

Expands finite data in a cosine basis. Several transform types differ in endpoint and normalization conventions.

$$X_k = \sum_{n=0}^{N-1} x_n \cos \frac{\pi(n+1/2)k}{N}$$

**268. Discrete Sine Transform [OPER]**

Expands finite data in a sine basis, with multiple types corresponding to different endpoint conventions.

$$X_k = \sum_{n=0}^{N-1} x_n \sin \frac{\pi(n+1)(k+1)}{N+1}$$

**269. Hilbert Transform [OPER]**

Applies a singular integral giving a phase-related companion to a suitable real-line function.

$$Hf(t) = \frac{1}{\pi} \text{PV} \int \frac{f(\tau)}{t-\tau} d\tau$$

**270. Hankel Transform [OPER]**

Uses a Bessel-function kernel to transform radial-type data, with a specified order and normalization.

$$H_\nu f(k) = \int_0^\infty f(r) J_\nu(kr) r dr$$

**271. Radon Transform [OPER]**

Integrates a function over lines or hyperplanes indexed by direction and offset.

$$Rf(\theta, s) = \int_{x \cdot \theta = s} f(x) d\sigma(x)$$

**272. Continuous Wavelet Transform [OPER]**

Pairs a function with translated and scaled copies of a chosen analyzing wavelet.

$$W_f(a, b) = |a|^{-1/2} \int f(t) \overline{\psi((t-b)/a)} dt$$

**273. Walsh-Hadamard Transform [OPER]**

Expands finite data in an orthogonal sign-pattern basis rather than a sinusoidal basis.

$$y = H_N x, \quad H_N H_N^T = NI$$

**274. Number-Theoretic Transform [OPER]**

Performs a finite Fourier-type transform in a modular field or ring containing the required roots of unity.

$$X_k = \sum_{n=0}^{N-1} x_n \omega^{kn} \pmod{p}$$

**18. Sequences, Series, and Convolution**

ENTRIES 275–288

A finite sum is always different in status from an infinite limiting process. Generalized summation must be labeled rather than substituted for ordinary convergence. [27] [28] [29] [30] [31]

**275. Finite Summation [OP]**

Adds the terms of a finite indexed family. An empty finite sum conventionally equals zero.

$$\sum_{k=1}^4 k = 10$$

**276. Finite Product [OP]**

Multiplies the terms of a finite indexed family. An empty product conventionally equals one.

$$\prod_{k=1}^4 k = 24$$

**277. Sequence Limit [OPER]**

Returns the value approached by a sequence when it converges in the specified space.

$$\lim_{n \rightarrow \infty} 1/n = 0$$

**278. Limit Superior [OPER]**

Returns the limiting supremum of sequence tails, possibly as an extended real value.

$$\limsup a_n = \lim_n \sup_{k \geq n} a_k$$

**279. Limit Inferior [OPER]**

Returns the limiting infimum of sequence tails, possibly as an extended real value.

$$\liminf a_n = \lim_n \inf_{k \geq n} a_k$$

**280. Infinite-Series Summation [OPER]**

Takes the limit of finite partial sums when that limit exists.

$$\sum_{n=0}^{\infty} a_n = \lim_N \sum_{n=0}^N a_n$$

**281. Infinite Product** [OPER]

Takes a limit of partial products, with the chosen convention about zero limits explicitly stated.

$$\prod_{n=1}^{\infty} a_n = \lim_N \prod_{n=1}^N a_n$$

**282. Cesaro Summation** [OPER]

Assigns a series a value when the arithmetic averages of its partial sums converge. It extends but differs from ordinary summation.

$$\frac{1}{N+1} \sum_{k=0}^N s_k \rightarrow L$$

**283. Abel Summation of a Series** [OPER]

Assigns a value by damping a series with powers of  $r$  and taking the limit as  $r$  approaches one from below.

$$\lim_{r \uparrow 1} \sum_{n \geq 0} a_n r^n$$

**284. Continuous Convolution** [OP]

Combines functions by integrating products against translated arguments, whenever the integral is well defined.

$$(f * g)(t) = \int f(\tau)g(t - \tau) d\tau$$

**285. Discrete Convolution** [OP]

Combines sequences by summing products over complementary indices. Support or convergence assumptions are required.

$$(a * b)_n = \sum_k a_k b_{n-k}$$

**286. Cross-Correlation** [OP]

Compares a function with a shifted second function. Conjugation and sign conventions must be fixed.

$$R_{fg}(\tau) = \int \overline{f(t)}g(t + \tau) dt$$

**287. Circular Convolution** [OP]

Convolve finite periodic sequences with indices interpreted modulo their common period.

$$(a *_N b)_n = \sum_{k=0}^{N-1} a_k b_{(n-k) \bmod N}$$

**288. Binomial Transform** [OPER]

Produces a sequence by binomially weighted sums of preceding terms; signed and unsigned conventions differ.

$$b_n = \sum_{k=0}^n \binom{n}{k} a_k$$

**19. Fractional, Discrete, and Generalized Calculus**

ENTRIES 289–302

These are established mathematical frameworks, not permission to disregard domains. Different fractional derivatives need not give the same answer. [23] [32] [33] [34]

**289. Forward Difference** [OPER]

Subtracts a function value from its value one fixed step later. Dividing by the step gives a difference quotient.

$$\Delta_h f(x) = f(x + h) - f(x)$$

**290. Backward Difference** [OPER]

Subtracts the value one step earlier from the present value.

$$\nabla_h f(x) = f(x) - f(x - h)$$

**291. Central Difference** [OPER]

Compares equally offset values around a point. A derivative approximation also divides by the full separation.

$$\delta_h f(x) = f(x + h/2) - f(x - h/2)$$

**292. Indefinite Discrete Summation** [OPER]

Finds an antidifference sequence, with an additive constant or initial value needed to make it unique.

$$F(n + 1) - F(n) = f(n)$$

**293. Jackson q-Derivative** [OPER]

Measures change under multiplicative scaling rather than additive displacement. The displayed formula excludes  $q = 1$  and  $x = 0$ .

$$D_q f(x) = \frac{f(qx) - f(x)}{(q - 1)x}$$

**294. Riemann-Liouville Fractional Integral** [OPER]

Integrates with a power-law kernel of positive order. The starting point and integrability assumptions are part of its definition.

$$I_a^\alpha f(x) = \frac{1}{\Gamma(\alpha)} \int_a^x (x - t)^{\alpha-1} f(t) dt$$

**295. Riemann-Liouville Fractional Derivative** [OPER]

Differentiates after a fractional integral. Its action on a constant can differ from that of a Caputo derivative. Here  $m - 1 < \alpha < m$ , with the same starting point for the integral.

$$D_{RL}^\alpha f = D^m I^{m-\alpha} f$$

**296. Caputo Fractional Derivative** [OPER]

Fractionally integrates an integer derivative, typically convenient for ordinary derivative initial data. Here  $m - 1 < \alpha < m$ .

$$D_C^\alpha f = I^{m-\alpha} D^m f$$

**297. Grunwald-Letnikov Derivative** [OPER]

Takes a scaled limit of fractional binomial differences. For a finite starting point  $a$ , the sum runs from  $k = 0$  to  $\lfloor (x - a)/h \rfloor$ ; existence hypotheses are required.

$$D_a^\alpha f(x) = \lim_{h \downarrow 0} h^{-\alpha} \cdot \sum_{k=0}^{\lfloor (x-a)/h \rfloor} (-1)^k \binom{\alpha}{k} f(x - kh)$$

**298. Fractional Laplacian** [OPER]

Defines a nonlocal power of the negative Laplacian, commonly for  $0 < s < 1$ . On the whole space, using the angular-frequency Fourier convention, its multiplier is  $|\xi|^{2s}$ .

$$\widehat{(-\Delta)^s f}(\xi) = |\xi|^{2s} \widehat{f}(\xi)$$

**299. Distributional Differentiation** [OPER]

Defines differentiation by transferring it to smooth compactly supported test functions with a minus sign.

$$\langle DT, \phi \rangle = -\langle T, \phi' \rangle$$

**300. Weak Differentiation** [OPER]

Identifies a locally integrable function representing a distributional derivative, when such a function exists.

$$\int g \phi = - \int f \phi'$$

**301. Mollification** [OPER]

Smooths a function or distribution by convolution with an appropriate smooth approximate identity.

$$f_\varepsilon = f * \rho_\varepsilon$$

**302. Spectral Fractional Power** [OPER]

Applies a scalar power to an operator through a suitable spectral calculus. Operator domain and spectrum restrictions remain essential.

$$A^s u_j = \lambda_j^s u_j$$

**20. Probability and Measure Operations**

ENTRIES 303–314

Probability operations require a probability model. Conditional and stochastic constructions include existence and integrability assumptions. [26] [35] [36]

**303. Expectation** [OPER]

Takes the probability-weighted average of an integrable random variable.

$$\mathbb{E}[X] = \int X dP$$

**304. Conditional Expectation** [OPER]

Averages an integrable random variable relative to a specified information sigma-algebra, producing a random variable defined up to almost-sure equality.

$$\mathbb{E}[X \mid \mathcal{G}]$$

**305. Conditional Probability** [OP]

Restricts attention to an event of positive probability and renormalizes. Conditioning on probability-zero events needs additional theory.

$$P(A \mid B) = P(A \cap B)/P(B)$$

**306. Probability Marginalization** [OPER]

Removes variables from a joint distribution by summing or integrating over them.

$$p_X(x) = \sum_y p_{X,Y}(x,y)$$

**307. Bayesian Updating** [OP]

Combines a likelihood and prior into a posterior distribution when the normalizing evidence is positive and finite.

$$p(\theta \mid x) \propto p(x \mid \theta)p(\theta)$$

**308. Pushforward Measure** [OPER]

Transports a measure through a measurable map by measuring inverse images of target sets.

$$(f_*\mu)(B) = \mu(f^{-1}[B])$$

**309. Product Measure** [CON]

Combines measures on product spaces under appropriate hypotheses. For probability measures, it models independent coordinate draws.

$$(\mu \otimes \nu)(A \times B) = \mu(A)\nu(B)$$

**310. Distribution Convolution** [OP]

Forms the law of a sum of independent random variables from their individual laws.

$$\mathcal{L}(X + Y) = \mathcal{L}(X) * \mathcal{L}(Y)$$

**311. Distribution Mixture [OP]**

Averages probability measures with nonnegative weights summing to one. This differs from summing independent random variables.

$$\mu = \sum_i w_i \mu_i, \quad \sum_i w_i = 1$$

**312. Characteristic-Function Transform [OPER]**

Maps a probability distribution to an always-defined complex-valued transform, with a chosen sign convention.

$$\phi_X(t) = \mathbb{E}[e^{itX}]$$

**313. Radon-Nikodym Differentiation [OPER]**

Expresses one measure as a density relative to another when the Radon-Nikodym theorem's hypotheses hold.

$$\nu(A) = \int_A \frac{d\nu}{d\mu} d\mu$$

**314. Ito Stochastic Integration [OPER]**

Integrates an appropriate adapted process against Brownian motion or a more general semimartingale, using the Ito convention.

$$\int_0^t H_s dW_s$$

---

**21. Statistics and Information Functionals**

ENTRIES 315–326

These map data or probability distributions to summaries. Sample formulas, population formulas, and normalization choices must not be confused. [35] [37] [38]

**315. Arithmetic Mean [FUN]**

Adds a nonempty finite collection of values and divides by the number of values.

$$\bar{x} = \frac{1}{n} \sum_i x_i$$

**316. Geometric Mean [FUN]**

Takes the  $n$ th root of the product of  $n$  positive real values.

$$G = \left( \prod_i x_i \right)^{1/n}$$

**317. Harmonic Mean [FUN]**

Takes the reciprocal of the arithmetic mean of reciprocals, ordinarily for positive inputs.

$$H = \frac{n}{\sum_i 1/x_i}$$

**318. Median [FUN]**

Returns the middle ordered observation, or commonly the mean of the two middle observations for an even sample size.

$$\text{median}(1, 4, 8) = 4$$

**319. Quantile [FUN]**

Returns a specified probability threshold of a distribution. Finite-sample interpolation conventions vary. The displayed generalized-inverse formula uses  $0 < p < 1$ .

$$Q(p) = \inf\{x : F(x) \geq p\}$$

**320. Variance [FUN]**

Averages squared deviations from the population mean, when the second moment is finite.

$$\text{Var}(X) = \mathbb{E}[(X - \mathbb{E}X)^2]$$

**321. Standard Deviation [FUN]**

Takes the nonnegative square root of variance, restoring the original measurement units.

$$\sigma_X = \sqrt{\text{Var}(X)}$$

**322. Covariance [OP]**

Averages products of centered random variables under appropriate moment conditions.

$$\text{Cov}(X, Y) = \mathbb{E}[(X - \mathbb{E}X)(Y - \mathbb{E}Y)]$$

**323. Pearson Correlation [OP]**

Normalizes covariance by positive finite standard deviations. It is undefined when either variance is zero.

$$\rho_{XY} = \text{Cov}(X, Y) / (\sigma_X \sigma_Y)$$

**324. Shannon Entropy [FUN]**

Measures a discrete distribution's uncertainty, using a specified logarithm base and the convention  $0 \log 0 = 0$ .

$$H(p) = - \sum_i p_i \log p_i$$

**325. Relative Entropy [OP]**

Compares two distributions by the Kullback-Leibler functional. It is asymmetric and can be infinite.

$$D_{KL}(p \parallel q) = \sum_i p_i \log(p_i / q_i)$$

**326. Mutual Information [OP]**

Measures dependence by comparing a joint distribution with the product of its marginals through relative entropy.

$$I(X; Y) = D_{KL}(p_{XY} \parallel p_X p_Y)$$

## 22. Geometric and Convex-Set Operations

ENTRIES 327–340

Transformations act on points or figures. Convex-set operations act on whole sets, and their outputs need not resemble either original input. [16] [39]

### 327. Translation [OP]

Adds a fixed displacement vector to every point of a figure.

$$x \mapsto x + a$$

### 328. Rotation [OP]

Applies an orientation-preserving orthogonal transformation about an origin, or its translated version about another center.

$$x \mapsto Rx, \quad R^T R = I, \det R = 1$$

### 329. Reflection [OP]

Mirrors points across a specified hyperplane; for a unit normal at the origin the formula subtracts twice the normal component.

$$x \mapsto x - 2(x \cdot n)n$$

### 330. Dilation [OP]

Scales displacement from a chosen center by a fixed factor.

$$x \mapsto c + \lambda(x - c)$$

### 331. Shear [OP]

Slides coordinates proportionally to other coordinates while preserving volume in the standard elementary example.

$$(x, y) \mapsto (x + ky, y)$$

### 332. Affine Transformation [OP]

Combines a linear transformation with a translation. It is invertible exactly when its linear part is invertible.

$$x \mapsto Ax + b$$

### 333. Perspective Projection [OP]

Maps a point along a viewing ray onto a selected plane, excluding points where the denominator vanishes.

$$(x, y, z) \mapsto (x/z, y/z) \quad (z \neq 0)$$

### 334. Circle or Sphere Inversion [OP]

Replaces distance from a fixed center by radius-squared divided by that distance, along the same ray.

$$x \mapsto c + \frac{r^2(x - c)}{\|x - c\|^2}$$

### 335. Convex-Hull Formation [CON]

Forms the smallest convex set containing  $S$ : all finite sums  $\sum_i \lambda_i x_i$  with  $x_i \in S$ ,  $\lambda_i \geq 0$ , and  $\sum_i \lambda_i = 1$ .

$$\text{conv}\{0, 2\} = [0, 2] \quad \text{in } \mathbb{R}$$

### 336. Affine-Hull Formation [CON]

Forms the smallest affine set containing  $S$ : all finite sums  $\sum_i \lambda_i x_i$  with  $x_i \in S$  and  $\sum_i \lambda_i = 1$ . Negative coefficients are allowed.

$$\text{aff}\{0, 2\} = \mathbb{R} \quad \text{in } \mathbb{R}$$

### 337. Minkowski Sum [OP]

Adds every point of one set to every point of another set. It differs from ordinary set union.

$$A + B = \{a + b : a \in A, b \in B\}$$

### 338. Pontryagin Set Difference [OP]

Keeps translations of the second set that fit entirely inside the first. It is not ordinary relative set difference.

$$A \ominus B = \{x : x + B \subseteq A\}$$

### 339. Polar-Set Formation [CON]

Forms points whose inner products with every original point are at most one, in the chosen pairing.

$$C^\circ = \{y : \langle y, x \rangle \leq 1 \quad \forall x \in C\}$$

### 340. Metric Projection onto a Convex Set [OPER]

Returns the unique nearest point in a nonempty closed convex set in Euclidean space. More general sets can have multiple nearest points.

$$P_C(x) = \underset{y \in C}{\operatorname{argmin}} \|x - y\|^2$$

## 23. Graph and Network Operations

ENTRIES 341–354

Assume finite simple graphs unless stated otherwise. Loops, parallel edges, labels, and attributes require explicit conventions. [40] [41]

### 341. Vertex Deletion [OPER]

Removes selected vertices and all edges incident to them.

$$G \mapsto G - v$$

### 342. Edge Deletion [OPER]

Removes specified edges while keeping all vertices.

$$G \mapsto G - e$$

### 343. Edge Contraction [OPER]

Identifies the endpoints of an edge. In a simple-graph convention, resulting loops are removed and parallel edges merged.

$$G \mapsto G/e$$

### 344. Edge Subdivision [OPER]

Replaces an edge with a path by inserting new vertices along it.

$$uv \mapsto u - w - v$$

**345. Graph Union** [OP]

Combines vertex and edge sets, identifying vertices with matching labels under the stated convention.

$$G \cup H = (V_G \cup V_H, E_G \cup E_H)$$

**346. Graph Disjoint Union** [OP]

Takes separate copies of graphs without adding edges between them.

$$G \sqcup H$$

**347. Graph Intersection** [OP]

Keeps common edges and the chosen common or fixed vertex set. The vertex convention must be stated.

$$E(G \cap H) = E(G) \cap E(H)$$

**348. Graph Complement** [OPER]

On a fixed vertex set, adds exactly the nonedges between distinct vertices and removes existing edges.

$$uv \in E(\overline{G}) \iff u \neq v, uv \notin E(G)$$

**349. Graph Join** [OP]

Starts with disjoint copies and adds every edge between the two vertex sets.

$$G \vee H$$

**350. Cartesian Graph Product** [OP]

Uses vertex pairs; adjacency changes one coordinate along an edge while keeping the other coordinate fixed.

$$(u, v) \sim (u', v) \text{ when } u \sim u'$$

**351. Tensor Graph Product** [OP]

Uses vertex pairs adjacent exactly when both corresponding coordinate pairs are adjacent. Also called direct or categorical graph product.

$$(u, v) \sim (u', v') \iff u \sim u' \text{ and } v \sim v'$$

**352. Strong Graph Product** [OP]

Combines Cartesian and tensor-product adjacency rules, permitting one or both coordinates to change along edges.

$$G \boxtimes H$$

**353. Line-Graph Formation** [CON]

Turns edges into vertices, with adjacency when the original edges share an endpoint.

$$V(L(G)) = E(G)$$

**354. Graph Power** [OPER]

Connects distinct vertices whose shortest-path distance in the original graph is at most a specified positive integer.

$$uv \in E(G^k) \iff 1 \leq d_G(u, v) \leq k$$

**24. Topological Operations and Constructions**

ENTRIES 355–368

Topology specifies continuity and neighborhoods. These constructions usually carry topologies, not just underlying sets. [42] [55]

**355. Topological Closure** [OPER]

Forms the smallest closed set containing an input set, including all its limit points.

$$\text{cl}(0, 1) = [0, 1] \text{ in } \mathbb{R}$$

**356. Topological Interior** [OPER]

Forms the largest open subset contained in the input set.

$$\text{int}[0, 1] = (0, 1) \text{ in } \mathbb{R}$$

**357. Topological Boundary** [OPER]

Keeps points in the closure but not the interior. Boundary depends on the ambient space.

$$\partial A = \text{cl } A \setminus \text{int } A$$

**358. Derived-Set Formation** [OPER]

Collects accumulation points: points whose every neighborhood contains another point of the set.

$$A' = \{x : x \text{ is an accumulation point of } A\}$$

**359. Subspace Topology** [CON]

Gives a subset the open sets obtained by intersecting it with ambient open sets.

$$\mathcal{T}_A = \{A \cap U : U \in \mathcal{T}\}$$

**360. Product Topology** [CON]

Gives a product of spaces the topology generated by restrictions in finitely many coordinates. It differs from the box topology for infinite products.

$$\prod_i X_i$$

**361. Quotient Topology** [CON]

Gives equivalence classes the topology in which a set is open exactly when its inverse image under the quotient map is open.

$$U \text{ open} \iff q^{-1}(U) \text{ open}$$

**362. Wedge Sum** [CON]

Joins pointed spaces by identifying their selected basepoints.

$$X \vee Y = (X \sqcup Y)/(x_0 \sim y_0)$$

**363. Smash Product** [CON]

Collapses the coordinate-basepoint subspaces in a product of pointed spaces to one point.

$$X \wedge Y = (X \times Y)/(X \vee Y)$$

**364. Suspension [CON]**

Attaches two cone directions to a space. The unreduced version collapses each end of a cylinder separately.

$$\Sigma X = (X \times [0, 1]) / (X \times \{0\}, X \times \{1\})$$

**365. Cone Formation [CON]**

Collapses one end of a cylinder on a space to a single vertex.

$$CX = (X \times [0, 1]) / (X \times \{1\})$$

**366. Mapping Cone [CON]**

Attaches a cone on a map's domain to its target along the map.

$$C_f = Y \cup_f CX$$

**367. Connected Sum [CON]**

Removes balls from suitably compatible manifolds and glues their boundary spheres, with dimension and orientation conventions.

$$M \# N$$

**368. Loop-Space Formation [CON]**

Forms a space of based loops with a specified mapping-space topology.

$$\Omega X = \{\gamma : [0, 1] \rightarrow X : \gamma(0) = \gamma(1) = x_0\}$$

## 25. Differential Geometry and Geometric Operators

ENTRIES 369–380

Smoothness, a connection, a metric, or an orientation may be required. Similar symbols can denote distinct operators in different settings. [16] [21] [55]

**369. Exterior Derivative [OPER]**

Differentiates a differential form into a form one degree higher, with a coordinate-independent definition and square zero.

$$d : \Omega^k \rightarrow \Omega^{k+1}, \quad d^2 = 0$$

**370. Pullback of Differential Forms [OPER]**

Transports a form backward along a smooth map by applying the derivative to its vector inputs.

$$(f^*\omega)_x(v_1, \dots) = \omega_{f(x)}(Df_x v_1, \dots)$$

**371. Tangent Pushforward [OPER]**

Sends tangent vectors forward using the derivative of a smooth map.

$$f_*v = Df_x(v)$$

**372. Lie Derivative [OPER]**

Measures change of a tensor field along the flow of a vector field. For forms it satisfies Cartan's formula.

$$\mathcal{L}_X \omega = \iota_X d\omega + d\iota_X \omega$$

**373. Vector-Field Lie Bracket [OP]**

Takes the commutator of vector fields acting as derivations on functions.

$$[X, Y]f = X(Yf) - Y(Xf)$$

**374. Covariant Derivative [OPER]**

Differentiates tensor fields relative to a specified connection, compensating for changes in local bases.

$$(\nabla_i v)^j = \partial_i v^j + \Gamma_{ik}^j v^k$$

**375. Parallel Transport [OPER]**

Moves a tangent vector along a specified path while keeping its covariant derivative along the path zero. It can depend on the path.

$$\nabla_{\dot{\gamma}} V = 0$$

**376. Geodesic Exponential Map [OPER]**

Maps an initial tangent vector to the endpoint at time one of its geodesic, where that geodesic is defined.

$$\exp_p(v) = \gamma_v(1)$$

**377. Laplace-Beltrami Operator [OPER]**

Extends the scalar Laplacian to a Riemannian manifold by taking the divergence of its gradient.

$$\Delta_g f = \operatorname{div}_g(\operatorname{grad}_g f)$$

**378. Codifferential [OPER]**

Acts as the formal adjoint of the exterior derivative under a metric-induced pairing, reducing form degree by one.

$$\delta : \Omega^k \rightarrow \Omega^{k-1}$$

**379. Hodge Laplacian [OPER]**

Combines the exterior derivative with its formal adjoint to act on differential forms. Sign conventions differ from some scalar Laplacians.

$$\Delta_H = d\delta + \delta d$$

**380. Poisson Bracket [OP]**

Pairs smooth functions on a Poisson manifold. In canonical position-momentum coordinates it uses cross-partial derivatives.

$$\{f, g\} = \sum_i (f_{q_i} g_{p_i} - f_{p_i} g_{q_i})$$

## 26. Category-Theoretic Constructions

ENTRIES 381–394

Universal constructions are characterized by mapping properties and are generally unique only up to canonical isomorphism, not literal equality of representatives. [43]

### 381. Morphism Composition [OP]

Composes arrows with matching source and target; categorical associativity and identity laws apply.

$$X \xrightarrow{f} Y \xrightarrow{g} Z \mapsto g \circ f$$

### 382. Opposite Category [CON]

Reverses every arrow and the order of composition while retaining the objects.

$$\mathrm{Hom}_{C^{op}}(X, Y) = \mathrm{Hom}_C(Y, X)$$

### 383. Functor Composition [OP]

Applies one structure-preserving mapping of categories after another.

$$(G \circ F)(X) = G(F(X))$$

### 384. Vertical Natural-Transformation Composition [OP]

Composes natural transformations componentwise between the same source and target categories.

$$(\beta \circ \alpha)_X = \beta_X \circ \alpha_X$$

### 385. Horizontal Natural-Transformation Composition [OP]

Combines natural transformations along composable functors. The resulting transformation compares the composed functors. Here  $\alpha : F \Rightarrow F'$  and  $\beta : G \Rightarrow G'$ .

$$(\beta * \alpha)_X = \beta_{F'X} \circ G(\alpha_X)$$

### 386. Categorical Product [CON]

Produces an object with projections through which each compatible pair of maps factors uniquely.

$$\mathrm{Hom}(T, A \times B) \cong \mathrm{Hom}(T, A) \times \mathrm{Hom}(T, B)$$

### 387. Categorical Coproduct [CON]

Produces an object with inclusions from which maps out are determined uniquely by maps out of its summands.

$$\mathrm{Hom}(A \sqcup B, T) \cong \mathrm{Hom}(A, T) \times \mathrm{Hom}(B, T)$$

### 388. Categorical Pullback [CON]

Produces the universal object mapping to two objects with matching composites to a third object.

$$A \times_C B$$

### 389. Categorical Pushout [CON]

Produces the universal object receiving maps from two objects whose restrictions from a third object agree.

$$A \sqcup_C B$$

### 390. Equalizer [CON]

Universally selects where two parallel morphisms agree. In sets, it is a subset of their common domain.

$$\mathrm{Eq}(f, g) = \{x : f(x) = g(x)\}$$

### 391. Coequalizer [CON]

Universally identifies the outputs of two parallel morphisms. In sets, it is a quotient by the generated equivalence relation.

$$qf = qg$$

### 392. Diagram Limit [CON]

Forms the universal compatible cone over a diagram, when it exists. Products and pullbacks are examples.

$$\varprojlim D$$

### 393. Diagram Colimit [CON]

Forms the universal compatible cocone under a diagram, when it exists. Coproducts and pushouts are examples.

$$\varinjlim D$$

### 394. Kan Extension [CON]

Extends a functor along another functor by a left or right universal property, when the relevant extension exists.

$$\mathrm{Lan}_K F, \quad \mathrm{Ran}_K F$$

## 27. Homology, Cohomology, and Higher Operations

ENTRIES 395–408

These are established theoretical operations. Coefficients, degrees, and indeterminacy are essential; a named higher operation need not have a single-valued output. [42] [44] [45] [55]

### 395. Boundary Operator on Chains [OPER]

Takes the oriented boundary of a chain. The boundary of a boundary is zero.

$$\partial : C_n \rightarrow C_{n-1}, \quad \partial^2 = 0$$

### 396. Coboundary Operator [OPER]

Acts on cochains by dualizing a boundary map, increasing degree and squaring to zero.

$$d : C^n \rightarrow C^{n+1}, \quad d^2 = 0$$

### 397. Homology Formation [CON]

Keeps cycles modulo boundaries in a chain complex. It is a construction rather than numerical division.

$$H_n(C) = \ker \partial_n / \mathrm{im} \partial_{n+1}$$

### 398. Cohomology Formation [CON]

Keeps cocycles modulo coboundaries in a cochain complex.

$$H^n(C) = \ker d^n / \mathrm{im} d^{n-1}$$

**399. Cup Product [OP]**

Multiplies cohomology classes, adding degrees, with a coefficient ring and graded sign rules.

$$H^p(X; R) \times H^q(X; R) \rightarrow H^{p+q}(X; R)$$

**400. Cap Product [OP]**

Pairs a cohomology class with a homology class, lowering homological degree by the cohomological degree.

$$H^p(X; R) \times H_n(X; R) \rightarrow H_{n-p}(X; R)$$

**401. External Homology Cross Product [OP]**

Combines homology classes from two spaces into a class on their product, adding dimensions.

$$H_p(X; R) \times H_q(Y; R) \rightarrow H_{p+q}(X \times Y; R)$$

**402. Connecting Homomorphism [OPER]**

Relates homology or cohomology groups in the long exact sequence associated with suitable short exact or topological data.

$$\delta : H^n(C) \rightarrow H^{n+1}(A)$$

**403. Bockstein Operation [OPER]**

Is a connecting cohomology homomorphism from a short exact coefficient sequence. The displayed mod- $p$  version uses  $0 \rightarrow \mathbb{Z}/p \rightarrow \mathbb{Z}/p^2 \rightarrow \mathbb{Z}/p \rightarrow 0$ .

$$\beta : H^n(X; \mathbb{Z}/p) \rightarrow H^{n+1}(X; \mathbb{Z}/p)$$

**404. Steenrod Square [OPER]**

Is a natural mod-two cohomology operation raising degree by its index, subject to standard axioms.

$$Sq^i : H^n(X; \mathbb{F}_2) \rightarrow H^{n+i}(X; \mathbb{F}_2)$$

**405. Steenrod Reduced Power [OPER]**

At an odd prime  $p$ , raises mod- $p$  cohomological degree by  $2i(p-1)$ . It generalizes but is not ordinary scalar exponentiation.

$$P^i : H^n \rightarrow H^{n+2i(p-1)}$$

**406. Massey Product [OP]**

Is a higher, generally partially defined cohomological operation with indeterminacy. A triple product requires suitable lower cup products to vanish.

$$\langle a, b, c \rangle \subseteq H^{|a|+|b|+|c|-1}$$

**407. Whitehead Product [OP]**

Pairs based homotopy classes into a higher-dimensional class. Here  $p, q \geq 2$ ; orientation and sign conventions must be fixed.

$$[\ , \ ] : \pi_p(X) \times \pi_q(X) \rightarrow \pi_{p+q-1}(X)$$

**408. Yoneda Product [OP]**

Composes extension classes by splicing extensions, adding their homological degrees.

$$\text{Ext}^p(B, C) \times \text{Ext}^q(A, B) \rightarrow \text{Ext}^{p+q}(A, C)$$

## 28. Alternative and Generalized Operation Frameworks

ENTRIES 409–422

These frameworks use precisely altered objects or rules. They are not interchangeable with ordinary arithmetic, and their existence is not a claim about physical reality. [45] [46] [47] [48] [49] [54]

**409. Tropical Addition [OP]**

In the min-plus convention, replaces ordinary addition by taking the minimum. The max-plus convention uses maximum instead.

$$a \oplus b = \min(a, b)$$

**410. Tropical Multiplication [OP]**

In min-plus arithmetic, replaces ordinary multiplication by ordinary addition, with appropriate conventions for infinity.

$$a \otimes b = a + b$$

**411. Nim Addition [OP]**

Adds nonnegative integer numbers by bitwise exclusive OR. It is not ordinary integer addition with carries.

$$3 \oplus 5 = 6$$

**412. Ordinal Addition [OP]**

Adds order types by placing the second well-ordered set after the first. It need not commute.

$$1 + \omega = \omega, \quad \omega + 1 > \omega$$

**413. Ordinal Multiplication [OP]**

Forms an ordered sequence of copies according to ordinal multiplication's specified order convention. It need not commute.

$$2 \cdot \omega = \omega, \quad \omega \cdot 2 > \omega$$

**414. Ordinal Exponentiation [OP]**

Defines powers by transfinite recursion, taking suprema at nonzero limit exponents.

$$\alpha^\lambda = \sup_{\beta < \lambda} \alpha^\beta \quad (\alpha > 0)$$

**415. Cardinal Addition [OP]**

Takes the cardinality of a disjoint union. Infinite cardinal addition differs from ordinal addition.

$$|A| + |B| = |A \sqcup B|$$

**416. Cardinal Multiplication [OP]**

Takes the cardinality of a Cartesian product.

$$|A| \cdot |B| = |A \times B|$$

**417. Cardinal Exponentiation [OP]**

Takes the cardinality of the set of all functions from the exponent's set to the base's set.

$$|A|^{|B|} = |\{f : B \rightarrow A\}|$$

**418. Standard-Part Map [OPER]**

In a hyperreal extension, sends a finite hyperreal to its unique infinitely close real number. Infinite hyperreals have no real standard part.

$$\text{st}(3 + \varepsilon) = 3 \quad (\varepsilon \text{ infinitesimal})$$

**419. Set-Valued Hyperoperation [OP]**

Combines inputs into a subset. A common hyperstructure convention requires a nonempty output, as displayed; broader versions permit the empty set. This differs from tetration-style hyperoperations.

$$\star : X \times X \rightarrow \mathcal{P}(X) \setminus \{\emptyset\}$$

**420. Operadic Substitution [OP]**

Inserts one multi-input operation into a selected input slot of another, obeying the operad's composition axioms.

$$\circ_i : O(m) \times O(n) \rightarrow O(m + n - 1)$$

**421. Totalized Reciprocation in a Meadow [OP]**

Extends reciprocation to zero under a specified alternative axiom system. An involutive meadow sets zero inverse to zero; ordinary field division remains restricted.

$$0^{-1} = 0 \quad \text{in this convention only}$$

**422. Deformation Star Product [OP]**

Deforms ordinary function multiplication into an associative formal product, typically with noncommutative higher-order terms.

$$f \star g = fg + \sum_{k \geq 1} \hbar^k B_k(f, g)$$

---

## 29. Optimization and Variational Constructions

ENTRIES 423–434

An optimization value is not the same as the point attaining it. Existence, convexity, differentiability, and domain assumptions control the constructions. [39] [50]

**423. Infimum-Value Operation [FUN]**

Maps an objective and a feasible set to the greatest lower bound of its feasible values, possibly infinite.

$$v = \inf_{x \in C} f(x)$$

**424. Argmin [OPER]**

Returns the set of feasible minimizers, which can be empty or contain multiple points. It is not the minimum value.

$$\operatorname{argmin}_{x \in C} f(x) = \{x \in C : f(x) = \inf_C f\}$$

**425. Argmax [OPER]**

Returns the set of points attaining the largest feasible objective value, when such points exist.

$$\operatorname{argmax}_{x \in C} f(x)$$

**426. Convex Subdifferential [OPER]**

Collects slopes of global affine lower supports to a convex function at a point. It can be set-valued or empty.

$$\partial f(x) = \{g : f(y) \geq f(x) + g \cdot (y - x) \quad \forall y\}$$

**427. Fenchel Conjugate [OPER]**

Transforms a function through a supremum over linear pairings minus function values, possibly producing infinite values.

$$f^*(y) = \sup_x (\langle y, x \rangle - f(x))$$

**428. Infimal Convolution [OP]**

Combines two functions by minimizing or taking the infimum of their sum over decompositions of the input.

$$(f \square g)(x) = \inf_y [f(y) + g(x - y)]$$

**429. Proximal Map [OPER]**

Balances an objective with squared distance. For a proper lower-semicontinuous convex function and positive parameter, its minimizer is unique.

$$\operatorname{prox}_{\lambda f}(v) = \operatorname{argmin}_x \left( f(x) + \frac{\|x - v\|^2}{2\lambda} \right)$$

**430. Moreau Envelope [OPER]**

Returns the optimal value of the proximal minimization problem rather than its minimizer.

$$e_{\lambda} f(v) = \inf_x \left( f(x) + \frac{\|x - v\|^2}{2\lambda} \right)$$

**431. Lagrangian Formation [CON]**

Combines an objective with equality and inequality constraints through multipliers; inequality multipliers require a sign convention.

$$L(x, \lambda, v) = f(x) + \lambda^T g(x) + v^T h(x)$$

**432. Lagrange Dual Function [OPER]**

Takes the infimum of a Lagrangian over primal variables to obtain a function of multipliers.

$$q(\lambda, v) = \inf_x L(x, \lambda, v)$$

**433. Gradient-Step Map** [OPER]

Updates a point by subtracting a chosen multiple of its gradient. Convergence requires more than the update formula alone.

$$x \mapsto x - \eta \nabla f(x)$$

**434. Soft Thresholding** [OPER]

Shrinks magnitudes toward zero by a fixed nonnegative amount, setting sufficiently small values to zero.

$$S_\lambda(x) = \text{sgn}(x) \max(|x| - \lambda, 0)$$

## 30. Related Symbolic and Numerical Procedures

ENTRIES 435–450

These are methods, not additional primitive algebraic operations. They are numbered separately within the catalog for practical reference. [17] [19] [22] [51] [52] [53]

**435. Algebraic Expansion** [PROC]

Rewrites products or powers as sums using valid distributive and exponent rules.

$$(x + 2)^2 = x^2 + 4x + 4$$

**436. Factorization** [PROC]

Rewrites an expression as a product of factors in a specified coefficient system; factorization depends on that system.

$$x^2 - 1 = (x - 1)(x + 1)$$

**437. Simplification** [PROC]

Rewrites expressions into a chosen simpler or canonical form while respecting domains. There is no universal best simplicity measure.

$$x/x = 1 \quad \text{only for } x \neq 0$$

**438. Completing the Square** [PROC]

Rewrites a quadratic using a shifted square, often to solve an equation or identify its geometry.

$$x^2 + 6x + 5 = (x + 3)^2 - 4$$

**439. Partial-Fraction Decomposition** [PROC]

Rewrites a rational function as a polynomial part plus simpler rational terms over a specified field.

$$\frac{1}{x(x+1)} = \frac{1}{x} - \frac{1}{x+1}$$

**440. Gaussian Elimination** [PROC]

Uses elementary row operations to solve linear systems or reach row-echelon form. Pivot selection is part of an implementation.

$$Ax = b \mapsto \text{row-equivalent system}$$

**441. Gram-Schmidt Orthogonalization** [PROC]

Subtracts projections to turn a linearly independent list into an orthogonal or orthonormal list spanning the same space.

$$u_k = v_k - \sum_{j < k} \text{proj}_{u_j} v_k$$

**442. Singular Value Decomposition** [PROC]

Factors a real or complex matrix into unitary factors and a rectangular diagonal singular-value matrix. Factors need not be unique.

$$A = U\Sigma V^*$$

**443. Groebner-Basis Computation** [PROC]

Produces an ideal basis suited to polynomial reduction under a selected monomial order and coefficient field.

$$I = \langle g_1, \dots, g_r \rangle$$

**444. Interpolation** [PROC]

Constructs a function agreeing with prescribed data in a chosen function class. Uniqueness depends on the class and data.

$$p(x_i) = y_i$$

**445. Numerical Quadrature** [PROC]

Approximates an integral by weighted function evaluations. Weights, nodes, and error assumptions specify the method.

$$\int_a^b f(x) dx \approx \sum_i w_i f(x_i)$$

**446. Newton Iteration** [PROC]

Approximates a root using tangent-line corrections where the derivative is nonzero. It need not converge from arbitrary starts.

$$x_{n+1} = x_n - f(x_n)/f'(x_n)$$

**447. Bisection** [PROC]

Repeatedly halves a sign-changing interval for a continuous real function, preserving a bracket around a root.

$$[a, b] \mapsto [a, m] \text{ or } [m, b]$$

**448. Fast Fourier Transform Algorithm** [PROC]

Computes a discrete Fourier transform efficiently by exploiting its structure. It is an algorithm for an existing transform, not a different transform.

$$\text{FFT}(x) \text{ computes } \text{DFT}(x)$$

**449. Automatic Differentiation [PROC]**

Applies the chain rule through elementary program operations, subject to differentiability and numerical arithmetic. It is not finite differencing.

$$D(f \circ g) = Df(g) Dg$$

**450. Summation by Parts [PROC]**

Rearranges finite sums using a discrete counterpart of integration by parts.

$$\sum_{k=m}^n a_k (b_{k+1} - b_k) = a_n b_{n+1} - a_m b_m - \sum_{k=m}^{n-1} b_{k+1} (a_{k+1} - a_k).$$

## Conventions, Pitfalls, and New Operations

### The Same Symbol Can Do Different Jobs

The star  $*$  may mean convolution, an adjoint, a group law, or a deformation product. The cross  $\times$  may mean numerical multiplication, a Cartesian product, or a three-dimensional vector cross product. The wedge  $\wedge$  can mean logical AND, a lattice meet, or an exterior product. Read the definitions of the objects around the symbol. [1][13][14][20][27]

### Do Not Erase the Domain

Canceling  $x$  in  $x/x$  requires  $x \neq 0$ . Applying a matrix inverse requires invertibility; a pseudoinverse is not the same operation. Taking an inverse trigonometric function requires a selected range. A formula may be correct on one domain and invalid on another. [4][6][19]

### Theoretical Does Not Mean Imaginary

A theoretical operation can be rigorously defined even when it is not part of school arithmetic. Cup products, tropical multiplication, standard-part maps, and operadic substitution act in distinct mathematical settings. Their validity is judged against their own definitions and axioms. Real-world applicability is a separate question. [42][45][46][47]

### Extensions Need Additional Choices

Complex roots, real-height tetration, fractional differentiation, generalized summation, and totalized reciprocals each need particular conventions or hypotheses. One successful construction does not establish that every competing construction is equivalent. [4][9][30][32][48]

### Operations Versus Algorithms

An operation specifies a mapping or result. An algorithm provides a sequence of steps to compute it or approximate it. A DFT is a transform; an FFT is an algorithm for

computing that transform. Matrix inversion is a mathematical operation; elimination is one computational tool that may help perform it. [17][19][28]

### How to Define a New Operation

Choose a set and an explicit rule, for example

$$a \diamond b = a + b + ab \quad (a, b \in \mathbb{R}).$$

The rule is closed on the reals. It is commutative because  $ab = ba$ . Its identity is zero, since  $a \diamond 0 = a$ . It is also associative:

$$\begin{aligned} (a \diamond b) \diamond c &= a + b + c + ab + ac + bc + abc \\ &= a \diamond (b \diamond c). \end{aligned}$$

To find an inverse, solve  $a + b + ab = 0$ . This gives  $b = -a/(1 + a)$  when  $a \neq -1$ . There is no inverse for  $a = -1$ , so the rule is not a group operation on all of  $\mathbb{R}$ . This example shows why both formulas and domains matter.

### A Compact Evaluation Checklist

For an unfamiliar entry, ask: What are the input objects? How many are required? Is the rule total or partial? What is the output type? Are results unique, or only determined up to an equivalence? Do order, signs, units, or convergence matter? What changes under a different convention?

### Scope of This Edition

The 450 entries include families, special cases, constructions, and 16 procedures. They are not 450 independent primitive operations. The reference emphasizes commonly encountered names and representative advanced frameworks, with brief definitions rather than full proofs of all existence theorems.

# Sources and Further Study

These references support the definitions and conventions discussed in the indicated chapters and provide routes to fuller theory. They are not claims that any one text contains every entry in a chapter. The explanations and worked comparisons in this reference are newly written; mathematical notation and underlying facts are not presented as exclusive inventions of this edition. Online sources were consulted on 28 September 2026. The explicitly identified print reference is included for further study.

[1] Thomas W. Judson. Abstract Algebra: Theory and Applications.

Sets, groups, rings, fields, modules, and algebraic structures.  
[https://math.libretexts.org/Bookshelves/Abstract\\_and\\_Geometric\\_Algebra/Abstract\\_Algebra%3A\\_Theory\\_and\\_Applications\\_%28Judson%29](https://math.libretexts.org/Bookshelves/Abstract_and_Geometric_Algebra/Abstract_Algebra%3A_Theory_and_Applications_%28Judson%29)

[2] NIST Digital Library of Mathematical Functions. Section 1.2: Elementary Algebra.

Arithmetic identities and polynomial operations.  
<https://dlmf.nist.gov/1.2>

[3] OpenStax. Precalculus 2e, Section 1.1: Functions and Function Notation.

Functions, inputs, outputs, and evaluation.  
<https://openstax.org/books/prec calculus-2e/pages/1-1-functions-and-function-notation>

[4] NIST Digital Library of Mathematical Functions. Section 4.2: Logarithm, Exponential, and Powers.

Definitions and complex branch conventions.  
<https://dlmf.nist.gov/4.2>

[5] NIST Digital Library of Mathematical Functions. Section 5.2: Gamma Function Definitions.

Gamma function and its singularities.  
<https://dlmf.nist.gov/5.2>

[6] SymPy Development Team. Elementary Functions. Trigonometric, hyperbolic, complex, integer, and related functions.

<https://docs.sympy.org/latest/modules/functions/elementary.html>

[7] SymPy Development Team. Combinatorial Functions. Factorials and counting-function conventions.

<https://docs.sympy.org/latest/modules/functions/combinatorial.html>

[8] SymPy Development Team. Number Theory.

Divisors, residues, arithmetic functions, and number-theoretic operations.

<https://docs.sympy.org/latest/modules/ntheory.html>

[9] R. Aldrovandi. Tetration: An Iterative Approach (2014). Integer iteration and the nontrivial problem of extending tetration.

<https://arxiv.org/abs/1410.3896>

[10] Marco Ripa and Luca Onnis. Number of Stable Digits of Any Integer Tetration (2022).

A research example of precisely defined integer tetration.  
<https://arxiv.org/abs/2210.07956>

[11] NIST Digital Library of Mathematical Functions. Section 1.9: Calculus of a Complex Variable.

Complex functions, contour integration, and residues.  
<https://dlmf.nist.gov/1.9>

[12] John C. Baez. The Octonions (2002).

Quaternion, octonion, Clifford, and related algebraic structures.

<https://arxiv.org/abs/math/0105155>

[13] SymPy Development Team. Sets.

Union, intersection, complements, products, and set-valued objects.

<https://docs.sympy.org/latest/modules/sets.html>

[14] SymPy Development Team. Logic.

Boolean connectives and logical-expression operations.

<https://docs.sympy.org/latest/modules/logic.html>

[15] The Stacks Project. Presheaves: Functoriality, Tag 00VC.

An advanced setting for composition and induced maps.  
<https://stacks.math.columbia.edu/tag/00VC>

[16] NIST Digital Library of Mathematical Functions. Section 1.6: Vectors and Vector-Valued Functions.

Vector algebra and geometric differential operations.  
<https://dlmf.nist.gov/1.6>

[17] Dan Margalit and Joseph Rabinoff. Interactive Linear Algebra: Matrix Multiplication.

Matrix multiplication and composition of linear maps.

<https://textbooks.math.gatech.edu/ila/matrix-multiplication.html>

[18] NIST Digital Library of Mathematical Functions. Section 1.3: Determinants, Linear Operators, and Spectral Expansions.

Linear algebra and operator notation.

<https://dlmf.nist.gov/1.3>

[19] SciPy Developers. Linear Algebra: `scipy.linalg`.

Matrix functions, decompositions, and computational conventions.

<https://docs.scipy.org/doc/scipy/reference/linalg.html>

[20] SymPy Development Team. Tensor Module.

Tensor products, index symmetries, and contractions.

<https://docs.sympy.org/latest/modules/tensor/tensor.html>

[21] SymPy Development Team. Differential Geometry.

Vector fields, differential forms, pullbacks, and covariant operations.

<https://docs.sympy.org/latest/modules/diffgeom.html>

[22] SymPy Development Team. Polynomial Manipulation: Reference.

Polynomial arithmetic, division, resultants, and Groebner bases.

<https://docs.sympy.org/latest/modules/polys/reference.html>

[23] NIST Digital Library of Mathematical Functions. Section 1.4: Calculus of One Variable.

Limits, derivatives, integrals, and Taylor-type operations.

<https://dlmf.nist.gov/1.4>

[24] NIST Digital Library of Mathematical Functions. Section 1.5: Calculus of Two or More Variables.

Multivariable differentiation and integration.

<https://dlmf.nist.gov/1.5>

[25] OpenStax. Calculus Volume 3, Section 6.5: Divergence and Curl.

Vector-calculus interpretations and definitions.

<https://openstax.org/books/calculus-volume-3/pages/6-5-divergence-and-curl>

[26] Terence Tao. An Introduction to Measure Theory (2011). Lebesgue integration, measurable functions, and measure constructions.

<https://teorth.github.io/tao-web/an-introduction-to-measure-theory.html>

[27] NIST Digital Library of Mathematical Functions. Section 1.14: Integral Transforms.

Fourier, Laplace, Mellin, Hilbert, and related transforms.

<https://dlmf.nist.gov/1.14>

[28] SymPy Development Team. Discrete Module.

Finite transforms and discrete convolutions.

<https://docs.sympy.org/latest/modules/discrete.html>

[29] NIST Digital Library of Mathematical Functions. Section 1.8: Fourier Series.

Frequency expansions, series, and convergence conventions.

<https://dlmf.nist.gov/1.8>

[30] NIST Digital Library of Mathematical Functions. Section 1.15: Summability Methods.

Ordinary and generalized summation must be distinguished.

<https://dlmf.nist.gov/1.15>

[31] NIST Digital Library of Mathematical Functions. Section 3.9: Acceleration of Convergence.

Sequence and series transformation methods.

<https://dlmf.nist.gov/3.9>

[32] Francesco Mainardi. Essentials of Fractional Calculus. Chapter 1 of Fractional Calculus and Waves in Linear Viscoelasticity; fractional integrals and derivatives.

<https://appliedmath.brown.edu/sites/default/files/fractional/11%20EssentialsofFractionalCalculus.pdf>

[33] NIST Digital Library of Mathematical Functions. Section 1.16: Distributions.

Generalized functions and distributional differentiation.

<https://dlmf.nist.gov/1.16>

[34] Mateusz Kwasnicki. Ten Equivalent Definitions of the Fractional Laplace Operator.

Whole-space fractional Laplacians and their hypotheses.

<https://arxiv.org/abs/1507.07356>

[35] Hossein Pishro-Nik. Introduction to Probability, Statistics, and Random Processes (2014).

Probability operations, expectations, moments, conditioning, and transforms.

<https://probabilitycourse.com/>

[36] Bernt Oksendal. Stochastic Differential Equations: An Introduction with Applications.

Springer, textbook; chapters on Ito integrals and stochastic differential equations. Print reference for Entry 314.

[37] SciPy Developers. Statistical Functions: `scipy.stats`.

Statistical summaries, distributions, and function conventions.

<https://docs.scipy.org/doc/scipy/reference/stats.html>

[38] SciPy Developers. Entropy and Relative Entropy.

Shannon entropy, logarithm bases, and Kullback-Leibler divergence.

<https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.entropy.html>

[39] Stephen Boyd and Lieven Vandenberghe. Convex Optimization (2004).

Convex sets, conjugates, subgradients, and Lagrange duality.

<https://stanford.edu/~boyd/cvxbook/>

[40] NetworkX Developers. Graph Operators.

Graph unions, joins, complements, and products.

<https://networkx.org/documentation/stable/reference/algorithms/operators.html>

[41] NetworkX Developers. Graph Minors.

Contraction and graph-minor conventions.

<https://networkx.org/documentation/stable/reference/algorithms/minors.html>

[42] Allen Hatcher. Algebraic Topology (2002; online corrected text).

Topological constructions, homology, cohomology, cup products, Bocksteins, and Steenrod operations.

<https://pi.math.cornell.edu/~hatcher/AT/AT.pdf>

[43] Emily Riehl. Category Theory in Context.

Categories, natural transformations, universal constructions, adjunctions, and Kan extensions.

<https://emilyriehl.github.io/files/context.pdf>

[44] The Stacks Project. Ext Groups, Tag 06XP.

Yoneda extensions and composition of extension classes.

<https://stacks.math.columbia.edu/tag/06XP>

[45] Jean-Louis Loday and Bruno Vallette. Algebraic Operads. Operadic composition, homotopy algebras, and higher structures.

<https://www.math.univ-paris13.fr/~vallette/Operads.html>

[46] David Speyer and Bernd Sturmfels. Tropical Mathematics (2004).

Min-plus tropical arithmetic and its geometric setting.  
<https://arxiv.org/abs/math/0408099>

[47] H. Jerome Keisler. Elementary Calculus: An Infinitesimal Approach.

Author-hosted online text; real and hyperreal numbers and standard parts.

<https://people.math.wisc.edu/~hkeisler/calc.html>

[48] J. A. Bergstra and C. A. Middelburg. Division by Zero in Non-Involutive Meadows.

Discusses totalized inverses and contrasts meadow conventions.  
<https://arxiv.org/abs/1406.2092>

[49] Maxim Kontsevich. Deformation Quantization of Poisson Manifolds, I.

Associative deformation star products; research paper.  
<https://arxiv.org/abs/q-alg/9709040>

[50] Neal Parikh and Stephen Boyd. Proximal Algorithms. Proximal maps, Moreau envelopes, and thresholding.

[https://web.stanford.edu/~boyd/papers/pdf/prox\\_algs.pdf](https://web.stanford.edu/~boyd/papers/pdf/prox_algs.pdf)

[51] NIST Digital Library of Mathematical Functions. Section 3.5: Quadrature.

Numerical integration methods.  
<https://dlmf.nist.gov/3.5>

[52] NIST Digital Library of Mathematical Functions. Section 3.8: Nonlinear Equations.

Root-finding and iteration methods.  
<https://dlmf.nist.gov/3.8>

[53] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind. Automatic Differentiation in Machine Learning: A Survey.

Distinguishes automatic differentiation from symbolic and numerical methods.

<https://arxiv.org/abs/1502.05767>

[54] So Nakamura and Manuel L. Reyes. Categories of Hypermagmas, Hypergroups, and Related Hyperstructures.

Journal of Algebra 676 (2025), 408–474. Includes conventions allowing empty hyperproducts.

<https://arxiv.org/abs/2304.09273>

[55] J. P. May. A Concise Course in Algebraic Topology.

Author-hosted text on homotopy, cohomology, and higher operations.

<https://math.uchicago.edu/~may/CONCISE/ConciseRevised.pdf>

## Alphabetical Index

Each line gives the entry name, its bracketed catalog number, and its page. Names and page numbers are clickable. Entries are indexed under their displayed names rather than every possible synonym.

|                                          |    |  |
|------------------------------------------|----|--|
| <b>A</b>                                 |    |  |
| Abel Summation of a Series [283]         | 22 |  |
| Absolute Value [9]                       | 6  |  |
| Addition [1]                             | 6  |  |
| Additive Negation [5]                    | 6  |  |
| Affine Transformation [332]              | 25 |  |
| Affine-Hull Formation [336]              | 25 |  |
| Algebraic Expansion [435]                | 31 |  |
| Anticommutator [177]                     | 15 |  |
| Antidifferentiation [246]                | 19 |  |
| Antisymmetrization [185]                 | 16 |  |
| Arccosine Evaluation [28]                | 7  |  |
| Arcsine Evaluation [27]                  | 7  |  |
| Arctangent Evaluation [29]               | 7  |  |
| Argmax [425]                             | 30 |  |
| Argmin [424]                             | 30 |  |
| Argument Extraction [83]                 | 10 |  |
| Argument Permutation [136]               | 13 |  |
| Arithmetic Mean [315]                    | 24 |  |
| Automatic Differentiation [449]          | 32 |  |
| <b>B</b>                                 |    |  |
| Backward Difference [290]                | 22 |  |
| Bayesian Updating [307]                  | 23 |  |
| Beta Function Evaluation [37]            | 7  |  |
| Bilateral Laplace Transform [263]        | 20 |  |
| Binomial Coefficient [49]                | 8  |  |
| Binomial Transform [288]                 | 22 |  |
| Bisection [447]                          | 31 |  |
| Bockstein Operation [403]                | 29 |  |
| Boundary Operator on Chains [395]        | 28 |  |
| <b>C</b>                                 |    |  |
| Cap Product [400]                        | 29 |  |
| Caputo Fractional Derivative [296]       | 23 |  |
| Cardinal Addition [415]                  | 29 |  |
| Cardinal Exponentiation [417]            | 30 |  |
| Cardinal Multiplication [416]            | 29 |  |
| Cartesian Graph Product [350]            | 26 |  |
| Cartesian Product [98]                   | 11 |  |
| Categorical Coproduct [387]              | 28 |  |
| Categorical Product [386]                | 28 |  |
| Categorical Pullback [388]               | 28 |  |
| Categorical Pushout [389]                | 28 |  |
| Cauchy Principal-Value Integration [258] | 20 |  |
| Cauchy Product [223]                     | 18 |  |
| Ceiling [40]                             | 8  |  |
| Central Difference [291]                 | 22 |  |
| Cesaro Summation [282]                   | 22 |  |
| Characteristic-Function form [312]       | 24 |  |
| Circle or Sphere Inversion [334]         | 25 |  |
| Circular Convolution [287]               | 22 |  |
| Clifford Product [191]                   | 16 |  |
| Clifford Reversion [192]                 | 16 |  |
| Coboundary Operator [396]                | 28 |  |
| Codifferential [378]                     | 27 |  |
| Coefficient Extraction [220]             | 18 |  |
| Coequalizer [391]                        | 28 |  |
| Cohomology Formation [398]               | 28 |  |
| Complement [96]                          | 11 |  |
| Completing the Square [438]              | 31 |  |
| Complex Conjugation [79]                 | 10 |  |
| Complex Exponential [84]                 | 10 |  |
| Complex Logarithm [85]                   | 10 |  |

|                                    |    |
|------------------------------------|----|
| Complex Modulus [82]               | 10 |
| Complex Power [86]                 | 10 |
| Complex Root-Set Formation [87]    | 10 |
| Conditional Expectation [304]      | 23 |
| Conditional Probability [305]      | 23 |
| Cone Formation [365]               | 27 |
| Conjugate Transpose [163]          | 14 |
| Conjunction [110]                  | 12 |
| Connected Sum [367]                | 27 |
| Connecting Homomorphism [402]      | 29 |
| Continuous Convolution [284]       | 22 |
| Continuous Tetration [76]          | 10 |
| Continuous Wavelet Transform [272] | 21 |
| Contour Integration [255]          | 20 |
| Converse Relation [139]            | 13 |
| Convex Subdifferential [426]       | 30 |
| Convex-Hull Formation [335]        | 25 |
| Coordinate Change [156]            | 14 |
| Cosecant Evaluation [26]           | 7  |
| Cosine Evaluation [22]             | 7  |
| Cotangent Evaluation [24]          | 7  |
| Covariance [322]                   | 24 |
| Covariant Derivative [374]         | 27 |
| Cross Product [148]                | 14 |
| Cross-Correlation [286]            | 22 |
| Cubing [12]                        | 6  |
| Cumulative Integration [247]       | 19 |
| Cup Product [399]                  | 29 |
| Curl [237]                         | 19 |
| Currying [133]                     | 13 |

**D**

|                                      |    |
|--------------------------------------|----|
| Definite Riemann Integration [245]   | 19 |
| Deformation Star Product [422]       | 30 |
| Derived-Set Formation [358]          | 26 |
| Determinant [165]                    | 14 |
| Diagram Colimit [393]                | 28 |
| Diagram Limit [392]                  | 28 |
| Dilation [330]                       | 25 |
| Direct Image [103]                   | 11 |
| Directional Differentiation [234]    | 18 |
| Dirichlet Convolution [66]           | 9  |
| Dirichlet Inversion [67]             | 9  |
| Discrete Convolution [285]           | 22 |
| Discrete Cosine Transform [267]      | 21 |
| Discrete Fourier Transform [266]     | 21 |
| Discrete Sine Transform [268]        | 21 |
| Disjoint Union [99]                  | 11 |
| Disjunction [111]                    | 12 |
| Distribution Convolution [310]       | 23 |
| Distribution Mixture [311]           | 24 |
| Distributional Differentiation [299] | 23 |
| Divergence [236]                     | 19 |
| Division [4]                         | 6  |

|                          |    |
|--------------------------|----|
| Divisor Sum [53]         | 8  |
| Dot Product [146]        | 13 |
| Double Factorial [46]    | 8  |
| Double Integration [249] | 19 |

**E**

|                                       |    |
|---------------------------------------|----|
| Edge Contraction [343]                | 25 |
| Edge Deletion [342]                   | 25 |
| Edge Subdivision [344]                | 25 |
| Equalizer [390]                       | 28 |
| Euler Totient [54]                    | 8  |
| Euler-Lagrange Operator [244]         | 19 |
| Exclusive OR [112]                    | 12 |
| Existential Quantification [118]      | 12 |
| Expectation [303]                     | 23 |
| Exponential Evaluation [19]           | 7  |
| Exponentiation [10]                   | 6  |
| Exterior Derivative [369]             | 27 |
| Exterior Product [186]                | 16 |
| External Homology Cross Product [401] | 29 |

**F**

|                                        |    |
|----------------------------------------|----|
| Factorial [45]                         | 8  |
| Factorization [436]                    | 31 |
| Falling Factorial [48]                 | 8  |
| Fast Fourier Transform Algorithm [448] | 31 |
| Fenchel Conjugate [427]                | 30 |
| Finite Product [276]                   | 21 |
| Finite Summation [275]                 | 21 |
| Floor [39]                             | 8  |
| Flux Integration [254]                 | 20 |
| Formal Differentiation [221]           | 18 |
| Formal Integration [222]               | 18 |
| Formal Logarithm and Exponential [228] | 18 |
| Formal Reciprocal [225]                | 18 |
| Formal Series Composition [226]        | 18 |
| Forward Difference [289]               | 22 |
| Fourier Transform [259]                | 20 |
| Fractional Function Iteration [77]     | 10 |
| Fractional Laplacian [298]             | 23 |
| Fractional Part [43]                   | 8  |
| Frechet Derivative [241]               | 19 |
| Free Product [202]                     | 17 |
| Frobenius Map [212]                    | 17 |
| Function Composition [126]             | 12 |
| Function Evaluation [125]              | 12 |
| Function Inversion [130]               | 13 |
| Function Restriction [131]             | 13 |
| Functional Conjugation [142]           | 13 |
| Functional Substitution [141]          | 13 |
| Functor Composition [383]              | 28 |

**G**

|                                      |    |
|--------------------------------------|----|
| Gamma Function Evaluation [36]       | 7  |
| Gateaux Directional Derivative [242] | 19 |
| Gaussian Elimination [440]           | 31 |
| Generated Sigma-Algebra [108]        | 11 |
| Geodesic Exponential Map [376]       | 27 |
| Geometric Mean [316]                 | 24 |
| Gradient [235]                       | 19 |
| Gradient-Step Map [433]              | 31 |
| Gram-Schmidt Orthogonalization [441] | 31 |
| Graph Complement [348]               | 26 |
| Graph Disjoint Union [346]           | 26 |
| Graph Intersection [347]             | 26 |
| Graph Join [349]                     | 26 |
| Graph Power [354]                    | 26 |
| Graph Union [345]                    | 26 |
| Greatest Common Divisor [51]         | 8  |
| Groebner-Basis Computation [443]     | 31 |
| Group Action [197]                   | 16 |
| Group Commutator [196]               | 16 |
| Group Conjugation [195]              | 16 |
| Group Direct Product [200]           | 16 |
| Group Inversion [194]                | 16 |
| Group Multiplication [193]           | 16 |
| Grunwald-Letnikov Derivative [297]   | 23 |

**H**

|                                                     |    |
|-----------------------------------------------------|----|
| Hadamard Product [168]                              | 15 |
| Hadamard Series Product [224]                       | 18 |
| Hankel Transform [270]                              | 21 |
| Harmonic Mean [317]                                 | 24 |
| Hermitian Inner Product [147]                       | 13 |
| Hessian Matrix [240]                                | 19 |
| Hexation [72]                                       | 10 |
| Higher Differentiation [230]                        | 18 |
| Higher Knuth-Arrow Operation [73]                   | 10 |
| Hilbert Transform [269]                             | 21 |
| Hodge Laplacian [379]                               | 27 |
| Hodge Star [188]                                    | 16 |
| Homology Formation [397]                            | 28 |
| Horizontal Natural-Transformation Composition [385] | 28 |
| Hyperbolic Cosine [32]                              | 7  |
| Hyperbolic Sine [31]                                | 7  |
| Hyperbolic Tangent [33]                             | 7  |

**I**

|                                     |    |
|-------------------------------------|----|
| Ideal Product [206]                 | 17 |
| Ideal Radical [207]                 | 17 |
| Ideal Sum [205]                     | 17 |
| Imaginary-Part Extraction [81]      | 10 |
| Improper Integration [248]          | 19 |
| Indefinite Discrete Summation [292] | 22 |

|                                    |    |
|------------------------------------|----|
| Index Permutation [183]            | 15 |
| Indexed Intersection [102]         | 11 |
| Indexed Union [101]                | 11 |
| Infimal Convolution [428]          | 30 |
| Infimum [121]                      | 12 |
| Infimum-Value Operation [423]      | 30 |
| Infinite Power-Tower Limit [78]    | 10 |
| Infinite Product [281]             | 22 |
| Infinite-Series Summation [280]    | 21 |
| Integer Quotient [56]              | 9  |
| Interior Product [187]             | 16 |
| Interpolation [444]                | 31 |
| Intersection [94]                  | 11 |
| Inverse Fourier Transform [260]    | 20 |
| Inverse Hyperbolic Evaluation [35] | 7  |
| Inverse Image [104]                | 11 |
| Inverse Laplace Transform [262]    | 20 |
| Ito Stochastic Integration [314]   | 24 |

## J

|                            |    |
|----------------------------|----|
| Jackson q-Derivative [293] | 22 |
| Jacobi Symbol [65]         | 9  |
| Jacobian Matrix [239]      | 19 |
| Jordan Product [211]       | 17 |

## K

|                         |    |
|-------------------------|----|
| Kan Extension [394]     | 28 |
| Kronecker Product [169] | 15 |

## L

|                                 |    |
|---------------------------------|----|
| Lagrange Dual Function [432]    | 30 |
| Lagrangian Formation [431]      | 30 |
| Lambert W Evaluation [38]       | 8  |
| Laplace Transform [261]         | 20 |
| Laplace-Beltrami Operator [377] | 27 |
| Laplacian [238]                 | 19 |
| Lattice Join [124]              | 12 |
| Lattice Meet [123]              | 12 |
| Least Common Multiple [52]      | 8  |
| Lebesgue Integration [256]      | 20 |
| Legendre Symbol [64]            | 9  |
| Lie Bracket [210]               | 17 |
| Lie Derivative [372]            | 27 |
| Limit Inferior [279]            | 21 |
| Limit Superior [278]            | 21 |
| Line-Graph Formation [353]      | 26 |
| Linear-Span Formation [157]     | 14 |
| Localization [208]              | 17 |
| Logarithm Evaluation [20]       | 7  |
| Logical Biconditional [114]     | 12 |
| Logical Negation [109]          | 12 |
| Loop-Space Formation [368]      | 27 |
| Lowering an Index [190]         | 16 |

## M

|                                 |    |
|---------------------------------|----|
| Mapping Cone [366]              | 27 |
| Massey Product [406]            | 29 |
| Material Implication [113]      | 12 |
| Matrix Addition [159]           | 14 |
| Matrix Commutator [176]         | 15 |
| Matrix Direct Sum [170]         | 15 |
| Matrix Exponential [172]        | 15 |
| Matrix Inversion [164]          | 14 |
| Matrix Logarithm [173]          | 15 |
| Matrix Multiplication [160]     | 14 |
| Matrix Power [171]              | 15 |
| Matrix Rank [167]               | 15 |
| Matrix Square Root [174]        | 15 |
| Matrix Trace [166]              | 14 |
| Matrix Transpose [162]          | 14 |
| Matrix-Vector Application [161] | 14 |
| Maximum [120]                   | 12 |
| Median [318]                    | 24 |
| Mellin Transform [264]          | 20 |

|                                           |    |
|-------------------------------------------|----|
| Metric Projection onto a Convex Set [340] | 25 |
|-------------------------------------------|----|

|                                     |    |
|-------------------------------------|----|
| Minimum [119]                       | 12 |
| Minkowski Sum [337]                 | 25 |
| Mixed Partial Differentiation [232] | 18 |
| Mobius Inversion [68]               | 9  |
| Modular Addition [57]               | 9  |
| Modular Division by a Unit [62]     | 9  |
| Modular Exponentiation [60]         | 9  |
| Modular Inversion [61]              | 9  |
| Modular Multiplication [59]         | 9  |
| Modular Subtraction [58]            | 9  |
| Module Scalar Action [204]          | 17 |
| Mollification [301]                 | 23 |
| Moore-Penrose Pseudoinverse [175]   | 15 |
| Moreau Envelope [430]               | 30 |
| Morphism Composition [381]          | 28 |
| Multinomial Coefficient [50]        | 8  |
| Multiplication [3]                  | 6  |
| Mutual Information [326]            | 24 |

## N

|                                  |    |
|----------------------------------|----|
| NAND [115]                       | 12 |
| Newton Iteration [446]           | 31 |
| Nim Addition [411]               | 29 |
| NOR [116]                        | 12 |
| Number-Theoretic Transform [274] | 21 |
| Numerical Quadrature [445]       | 31 |

## O

|                              |    |
|------------------------------|----|
| Octonion Multiplication [92] | 11 |
| Operadic Substitution [420]  | 30 |
| Opposite Category [382]      | 28 |
| Ordinal Addition [412]       | 29 |
| Ordinal Exponentiation [414] | 29 |

|                                         |    |
|-----------------------------------------|----|
| Ordinal Multiplication [413]            | 29 |
| Ordinary Differentiation [229]          | 18 |
| Orthogonal Projection onto a Line [152] | 14 |
| Orthogonal-Complement Formation [158]   | 14 |
| Outer Product [149]                     | 14 |

## P

|                                          |    |
|------------------------------------------|----|
| Parallel Transport [375]                 | 27 |
| Partial Application [135]                | 13 |
| Partial Differentiation [231]            | 18 |
| Partial Trace [180]                      | 15 |
| Partial-Fraction Decomposition [439]     | 31 |
| Pearson Correlation [323]                | 24 |
| Pentation [71]                           | 9  |
| Percentage Scaling [16]                  | 6  |
| Perspective Projection [333]             | 25 |
| Pointwise Function Addition [127]        | 12 |
| Pointwise Function Division [129]        | 13 |
| Pointwise Function Multiplication [128]  | 12 |
| Poisson Bracket [380]                    | 27 |
| Polar-Set Formation [339]                | 25 |
| Polynomial Addition [213]                | 17 |
| Polynomial Composition [215]             | 17 |
| Polynomial Discriminant [219]            | 17 |
| Polynomial Greatest Common Divisor [217] | 17 |
| Polynomial Multiplication [214]          | 17 |
| Polynomial Quotient and Remainder [216]  | 17 |
| Polynomial Resultant [218]               | 17 |
| Pontryagin Set Difference [338]          | 25 |
| Power-Set Formation [100]                | 11 |
| Predecessor [8]                          | 6  |
| Prime-Adic Valuation [63]                | 9  |
| Principal Square Root [13]               | 6  |
| Probability Marginalization [306]        | 23 |
| Product Measure [309]                    | 23 |
| Product Topology [360]                   | 26 |
| Proximal Map [429]                       | 30 |
| Pullback of Differential Forms [370]     | 27 |
| Pullback of Functions [137]              | 13 |
| Pushforward Measure [308]                | 23 |

## Q

|                                     |    |
|-------------------------------------|----|
| Quantile [319]                      | 24 |
| Quaternion Conjugation [89]         | 10 |
| Quaternion Inversion [90]           | 10 |
| Quaternion Multiplication [88]      | 10 |
| Quaternion Rotation Action [91]     | 11 |
| Quotient Topology [361]             | 26 |
| Quotient-Group Multiplication [198] | 16 |
| Quotient-Set Formation [105]        | 11 |

**R**

|                                               |    |
|-----------------------------------------------|----|
| Radon Transform [271]                         | 21 |
| Radon-Nikodym Differentiation [313]           | 24 |
| Raising an Index [189]                        | 16 |
| Ratio Formation [17]                          | 7  |
| Real Cube Root [14]                           | 6  |
| Real nth Root [15]                            | 6  |
| Real-Part Extraction [80]                     | 10 |
| Reciprocal Hyperbolic Evaluation [34]         | 7  |
| Reciprocal-Trigonometric Inversion [30]       | 7  |
| Reciprocation [6]                             | 6  |
| Reflection [329]                              | 25 |
| Relation Composition [138]                    | 13 |
| Relative Difference [95]                      | 11 |
| Relative Entropy [325]                        | 24 |
| Remainder Modulo a Positive Integer [55]      | 9  |
| Repeated Function Iteration [69]              | 9  |
| Riemann-Liouville Fractional Derivative [295] | 22 |
| Riemann-Liouville Fractional Integral [294]   | 22 |
| Ring Addition and Multiplication [203]        | 17 |
| Rising Factorial [47]                         | 8  |
| Rotation [328]                                | 25 |
| Rounding to Nearest [42]                      | 8  |

**S**

|                                  |    |
|----------------------------------|----|
| Scalar Line Integration [251]    | 20 |
| Scalar Multiplication [145]      | 13 |
| Scalar Surface Integration [253] | 20 |
| Scalar Triple Product [154]      | 14 |
| Schur Complement [179]           | 15 |
| Secant Evaluation [25]           | 7  |
| Semidirect Product [201]         | 16 |
| Sequence Limit [277]             | 21 |
| Series Reversion [227]           | 18 |
| Set Cardinality [106]            | 11 |
| Set Partition Formation [107]    | 11 |

|                                    |    |
|------------------------------------|----|
| Set-Valued Hyperoperation [419]    | 30 |
| Shannon Entropy [324]              | 24 |
| Shear [331]                        | 25 |
| Sign Function [44]                 | 8  |
| Similarity Transformation [178]    | 15 |
| Simplification [437]               | 31 |
| Sine Evaluation [21]               | 7  |
| Singular Value Decomposition [442] | 31 |
| Smash Product [363]                | 26 |
| Soft Thresholding [434]            | 31 |
| Spectral Fractional Power [302]    | 23 |
| Squaring [11]                      | 6  |
| Standard Deviation [321]           | 24 |
| Standard-Part Map [418]            | 30 |
| Steenrod Reduced Power [405]       | 29 |
| Steenrod Square [404]              | 29 |
| Stieltjes Integration [257]        | 20 |
| Strong Graph Product [352]         | 26 |
| Subgroup Generation [199]          | 16 |
| Subspace Topology [359]            | 26 |
| Subtraction [2]                    | 6  |
| Successor [7]                      | 6  |
| Summation by Parts [450]           | 32 |
| Super-Logarithm [75]               | 10 |
| Super-Root [74]                    | 10 |
| Supremum [122]                     | 12 |
| Suspension [364]                   | 27 |
| Symmetric Difference [97]          | 11 |
| Symmetrization [184]               | 15 |

**T**

|                                 |    |
|---------------------------------|----|
| Tangent Evaluation [23]         | 7  |
| Tangent Pushforward [371]       | 27 |
| Tensor Contraction [182]        | 15 |
| Tensor Graph Product [351]      | 26 |
| Tensor Product [181]            | 15 |
| Tensor Product of Modules [209] | 17 |
| Tetration [70]                  | 9  |
| Topological Boundary [357]      | 26 |
| Topological Closure [355]       | 26 |
| Topological Interior [356]      | 26 |

|                                           |    |
|-------------------------------------------|----|
| Total Differential [233]                  | 18 |
| Totalized Reciprocation in a Meadow [421] | 30 |
| Transitive Closure [140]                  | 13 |
| Translation [327]                         | 25 |
| Triple Integration [250]                  | 19 |
| Tropical Addition [409]                   | 29 |
| Tropical Multiplication [410]             | 29 |
| Truncation Toward Zero [41]               | 8  |

**U**

|                                |    |
|--------------------------------|----|
| Uncurrying [134]               | 13 |
| Union [93]                     | 11 |
| Universal Quantification [117] | 12 |

**V**

|                                                   |    |
|---------------------------------------------------|----|
| Variance [320]                                    | 24 |
| Variational Derivative [243]                      | 19 |
| Vector Addition [143]                             | 13 |
| Vector Line Integration [252]                     | 20 |
| Vector Norm [150]                                 | 14 |
| Vector Normalization [151]                        | 14 |
| Vector Rejection [153]                            | 14 |
| Vector Subtraction [144]                          | 13 |
| Vector Triple Product [155]                       | 14 |
| Vector-Field Lie Bracket [373]                    | 27 |
| Vertex Deletion [341]                             | 25 |
| Vertical Natural-Transformation Composition [384] | 28 |

**W**

|                                |    |
|--------------------------------|----|
| Walsh-Hadamard Transform [273] | 21 |
| Weak Differentiation [300]     | 23 |
| Wedge Sum [362]                | 26 |
| Weighted Combination [18]      | 7  |
| Whitehead Product [407]        | 29 |

**Y**

|                      |    |
|----------------------|----|
| Yoneda Product [408] | 29 |
|----------------------|----|

**Z**

|                      |    |
|----------------------|----|
| Z-Transform [265]    | 20 |
| Zero Extension [132] | 13 |