POSIX Nexus serves as a comprehensive cross-language reference hub that explores the implementation and behavior of POSIX-compliant functionality across a diverse set of programming environments. Built atop the foundational IEEE Portable Operating System Interface (POSIX) standards, this project emphasizes compatibility, portability, and interoperability between operating systems.

Abstract

Contents

I Notation

Summation Notation

Symbol	Σ (uppercase Greek Sigma)
Meaning	Summation — add a sequence of terms
Syntax	$\sum_{i=1}^{n} a_i$ means add all a_i from $i = 1$ to $i = n$
Example	$\sum_{i=1}^{4} i = 1 + 2 + 3 + 4 = 10$

Product Notation

Symbol	$\prod$ (uppercase Greek Pi)
Meaning	Product — multiply a sequence of terms
Syntax	$\prod_{i=1}^{n} a_i$ means multiply all a_i from $i = 1$ to $i = n$
Example	$\prod_{i=1}^{4} i = 1 \times 2 \times 3 \times 4 = 24$

Factorial Notation

Symbol	$n!$
Meaning	Multiply all positive integers from 1 to n
Syntax	$n! = n \times (n-1) \times (n-2) \times \cdots \times 1$
Example	$5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$

Integral Notation

Symbol	$\int$ (elongated, slanted "S")
Meaning	Integration — summing infinitely small quantities over an interval
Syntax	$\int_a^b f(x)\,dx$ means integrate $f(x)$ from $x = a$ to $x = b$
Example	$\int_0^2 x\,dx = 2$

Limit Notation

Symbol	lim
Meaning	Limit — the value a function approaches as input nears a point
Syntax	$\lim_{x\to a} f(x)$ means the value of $f(x)$ as x approaches a
Example	$\lim_{x\to 0} \frac{\sin(x)}{x} = 1$

Derivative Notation

Symbol	$\frac{df}{dx}$ or $f'(x)$
Meaning	Derivative — rate of change of a function with respect to its input
Syntax	$\frac{df}{dx}$ means how $f(x)$ changes as x changes
Example	If $f(x) = x^2$, then $f'(x) = 2x$

II Definitions

Function

Concept	Function
Definition	A function is a rule that takes an input and gives back exactly one output. It describes how one quantity depends on another.
Core Idea	You give it a number, and it gives you a result — like a machine that transforms input into output.
Example	The function $f(x) = x^2$ takes any number x and returns $x \times x$. So $f(3) = 9$.
Applications	Used to model relationships in math, science, engineering, economics, and everyday situations.

Derivative

Concept	Derivative
Definition	A derivative tells you how fast a function's output is changing compared to its input. It measures the rate of change — like speed for a moving object.
Core Idea	If a function is a machine, the derivative tells you how sensitive that machine is — how much the output jumps when you nudge the input.
Example	For $f(x) = x^2$, the derivative is $f'(x) = 2x$. At $x = 3$, the slope is 6, meaning the output is rising quickly.
Applications	Used to study motion, growth, optimization, and any situation where change matters.

Slope

Concept	Slope
Definition	Slope tells you how steep a line is — how much the output changes compared to the input. It's the rate of change between two points.
Core Idea	If you move one step in the input direction, the slope tells you how many steps the output moves.
Example	A line with slope 2 means every time x increases by 1, y increases by 2.
Applications	Used in graphs, physics (speed), economics (cost change), and in derivatives to describe how functions behave.

Tangent Line

Concept	Tangent Line
Definition	A tangent line is a straight line that touches a curve at exactly one point and moves in the same direction as the curve at that point. It shows the curve's slope or rate of change at that location.
Core Idea	It's the best straight-line approximation of the curve near a point — like zooming in until the curve looks flat.
Example	For the curve $f(x) = x^2$, the tangent line at $x = 2$ has slope 4, so it rises steeply.
Applications	Used in physics (instantaneous velocity), optimization (finding peaks), and calculus (defining derivatives).

Tangent Function

Concept	Tangent (tan)
Definition	The tangent of an angle is the ratio of the opposite side to the adjacent side in a right triangle. It's also defined as $\tan(x) = \frac{\sin(x)}{\cos(x)}$.
Core Idea	Tangent compares how tall something is versus how far it runs — it's a slope.
Example	For a 45° angle, $\tan(45°) = 1$, meaning rise equals run.
Applications	Used in trigonometry, navigation, physics, and calculus — especially for modeling slopes and angles.

III Functions

III Infinity Series

An infinite series is written as:

$$\sum_{n=1}^{\infty} a_n = a_1 + a_2 + a_3 + \cdots$$

Types of Infinite Series

Type	Description
Arithmetic Series	Constant difference between terms (usually diverges)
Geometric Series	Constant ratio between terms; converges if $\lvert r \rvert < 1$
Harmonic Series	$\sum \frac{1}{n}$; diverges slowly
Taylor Series	Approximates functions using derivatives at a point

Convergence vs Divergence

Convergent Series	The sum of terms approaches a finite value as more terms are added
Divergent Series	The sum grows without bound or fails to settle on a single value
Test Method	Use the limit of partial sums: $S_n = a_1 + a_2 + \cdots + a_n$
Convergence Condition	$\lim_{n\to\infty} S_n$ exists and is finite
Divergence Condition	$\lim_{n\to\infty} S_n$ does not exist or is infinite

III Taylor and Maclaurin Series

The Taylor series is a powerful tool in mathematical analysis that expresses functions as infinite polynomials based on their derivatives at a single point. Brook Taylor introduced this concept in 1715. Colin Maclaurin later developed a special case — the Maclaurin series — where the expansion is centered at zero. These series are foundational in calculus, physics, and numerical approximation.

Maclaurin-style emitter for sine/cosine

```
<MINTED>
```

Purpose of `nx_esp(x, y)`

- **Function** ⇒ Computes a single term of a Taylor (or Maclaurin) series
- **Formula** ⇒ Returns $\frac{x^y}{y!}$
- **Inputs** ⇒ x: the value being evaluated, y: the exponent and factorial index
- **Use** ⇒ Used inside a summation loop to build polynomial approximations of functions like e^x, $\sin(x)$, or $\cos(x)$

Audit of `nx_esp(x, y)`

- **Purpose** ⇒ Emit a Taylor series term: $\frac{x^y}{y!}$
- **Correctness** ⇒ Yes — matches the canonical form for functions with constant derivatives
- **Assumptions** ⇒ Assumes $f^{(y)}(0) = 1$, which holds for e^x, and alternates for sine/cosine
- **Limitations** ⇒ Does not handle functions with nontrivial derivative values (e.g., $\ln(1+x)$, $\tan(x)$)

Audit of `nx_ts(n, t, p, k)`

- **Purpose** ⇒ Emit a partial Taylor series with alternating signs and stepwise powers
- **Valid For** ⇒ Functions like $\sin(x)$, $\cos(x)$, and e^x with predictable derivatives
- **Limitations** ⇒ Does not handle arbitrary derivatives — assumes all $f^{(n)}(0) = 1$
- **Use** ⇒ Can approximate sine/cosine-like functions with controlled precision and step size

Argument Reduction — Audit Summary

- **Purpose** ⇒ Reduce a large input x to a smaller angle where the function converges faster
- **Why** ⇒ Taylor series converge slowly for large x; reducing the angle improves precision
- **How** ⇒ Use periodicity: $\sin(x) = \sin(x \mod 2\pi)$, $\cos(x) = \cos(x \mod 2\pi)$
- **Example** ⇒ Instead of computing $\sin(54)$, compute $\sin(54 \mod 2\pi) \approx \sin(3.77)$

Who Was Brook Taylor?

- **Born** ⇒ 1685 in England
- **Died** ⇒ 1731
- **Known for** ⇒ Inventing the general Taylor series and contributing to calculus and geometry
- **Legacy** ⇒ His 1715 work introduced the method of increments, laying the foundation for Taylor expansions

Who Was Colin Maclaurin?

- **Born** ⇒ 1698 in Scotland
- **Died** ⇒ 1746
- **Known for** ⇒ Advancing calculus, geometry, and mathematical physics
- **Legacy** ⇒ He formalized the Taylor series centered at zero, which became known as the Maclaurin series

> Maclaurin was a child prodigy — he entered university at age 11 and became a professor at 19. He worked closely with Newton's ideas and helped defend calculus against critics who doubted its rigor.

Taylor Series vs Maclaurin Series

- **Taylor Series** ⇒ Centered at any point a; uses derivatives at a
- **Maclaurin Series** ⇒ Special case of Taylor series centered at $a = 0$
- **Use Cases** ⇒ Function approximation, solving differential equations, physics simulations
- **Common Functions** ⇒ e^x, $\sin(x)$, $\cos(x)$, $\ln(1 + x)$

What Is the Maclaurin Series?
It's a Taylor series centered at $a = 0$. That means all derivatives are evaluated at zero:

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(0)}{n!} x^n$$

The Taylor series of a function $f(x)$ centered at a point a is:

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!} (x-a)^n$$

Bernoulli Numbers — Audit Summary

- **Definition** $\Rightarrow$ Rational numbers appearing in power series expansions and number theory
- **Generating Function** $\Rightarrow$ $\frac{x}{e^x - 1} = \sum_{n=0}^{\infty} \frac{B_n x^n}{n!}$
- **Odd Index Rule** $\Rightarrow$ All $B_{2n+1} = 0$ for $n \geq 1$
- **Applications** $\Rightarrow$ Used in tangent series, zeta function values, and Faulhaber's formula

$$\tan(x) = x + \frac{x^3}{3} + \frac{2x^5}{15} + \frac{17x^7}{315} + \cdots$$

$$\tan(x) = \sum_{n=1}^{\infty} \frac{(-1)^{n-1} \cdot 2^{2n} \cdot (2^{2n} - 1) \cdot B_{2n}}{(2n)!} \cdot x^{2n-1}$$

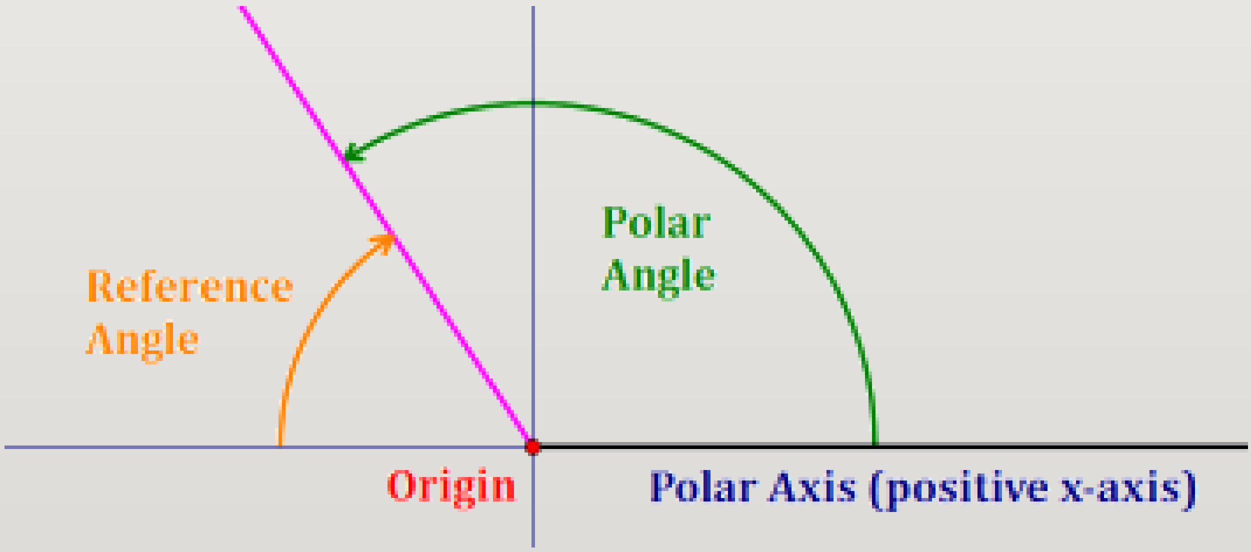
IV Math Terms

Standard Number Sets

Symbol	Definition
$\mathbb{Z}$	The set of all integers: $\{\ldots, -2, -1, 0, 1, 2, \ldots\}$. Includes negative, zero, and positive whole numbers.
$\mathbb{Q}$	The set of all rational numbers: numbers expressible as $\frac{p}{q}$ where $p \in \mathbb{Z}$, $q \in \mathbb{Z} \setminus \{0\}$.
$\mathbb{R}$	The set of all real numbers: includes both rational and irrational numbers. Forms a complete ordered field.
$\mathbb{C}$	The set of all complex numbers: numbers of the form $a + bi$ where $a, b \in \mathbb{R}$, and $i^2 = -1$.

V Trigonometry

V Cartesian (xy) Plane



Cartesian Plane — The Rectangular Canvas

- **Axes** ⇒ Two perpendicular axes x and y defining horizontal and vertical directions
- **Origin** ⇒ The central point $(0, 0)$ where axes intersect and measurements begin
- **Grid** ⇒ An infinite lattice of parallel lines marking unit intervals
- **Quadrants** ⇒ Four sign realms labeling positions of points relative to the origin

Rectangular vs Polar Coordinate Systems

Feature	Rectangular	Polar
Parameters	(x, y) along orthogonal axes	(r, θ) as radius and angle
Basis	Linear displacement on two lines	Radial distance and angular measure
Conversion	$x = r\cos\theta,\ y = r\sin\theta$	$r = \sqrt{x^2 + y^2},\ \theta = \operatorname{atan2}(y, x)$
Use Cases	Cartesian graphs, analytic geometry	Circular motion, waves, radial fields

$$x = r\cos\theta, \qquad y = r\sin\theta,$$
$$r = \sqrt{x^2 + y^2}, \qquad \theta = \operatorname{atan2}(y, x)$$

The Cartesian Plane, conceived by René Descartes in the 17th century, is the foundational rectangular framework where points are given by ordered pairs (x, y). This plane underlies the rectangular coordinate system, mapping linear displacements along perpendicular axes. Polar coordinates overlay this same plane with a circular measure, locating points by distance r from the origin and angle θ from the positive x-axis. Thus the Cartesian Plane serves as the common canvas for both rectangular and polar systems.

Key Insights

- The Cartesian Plane is the rectangular grid defined by x and y axes
- Rectangular coordinates use (x, y) directly on this grid
- Polar coordinates use (r, θ) over the same grid by circular mapping
- Conversion relies on sine, cosine, and the Pythagorean theorem
- Both systems coexist on the Cartesian Plane, each suited to different problems

The Cartesian Plane is the shared canvas, rectangular at its core, circular in its polar overlay.

Standard Position — The Glyph of Zero

- **Vertex** ⇓ The common endpoint of the angle, fixed at $(0, 0)$
- **Initial Side** ⇓ The ray on the positive x-axis from the origin
- **Terminal Side** ⇓ The rotating ray that sweeps from the initial side
- **Rotation Direction** ⇓ Counterclockwise for positive angles; clockwise for negative
- **Quadrantal Angles** ⇓ Angles whose terminal side lies on an axis (0°, 90°, 180°, 270°)

Angle Anatomy — Standard Position

Component	Description
Vertex	Located at the origin where the two sides meet
Initial Side	Fixed along the positive x-axis as the starting ray
Terminal Side	The ray rotated from the initial side to form the angle
Positive Rotation	Measured counterclockwise from the initial side
Negative Rotation	Measured clockwise from the initial side

An angle is in standard position when its vertex is at the origin and its initial side lies along the positive x-axis. The terminal side rotates about the origin: counterclockwise for positive measures and clockwise for negative. This convention provides a unified framework for defining trigonometric functions and angle measures, from quadrantal angles to arbitrary sweeps.

Key Insights on Standard Position

- Standard Position fixes the vertex at $(0, 0)$ and the initial side on the positive x-axis
- Positive angles rotate counterclockwise; negative rotate clockwise
- Terminal sides on axes define quadrantal angles
- Essential for defining sine, cosine, tangent and more
- Forms the basis for rectangular and polar coordinate linkage

> Standard Position is the cardinal glyph of angle measurement — the origin, the positive x-axis, and the sweep of rotation.

Endpoint — The Termination Glyph

- **Definition** ⇒ A point marking where a geometric object ends: one end of a segment or the single start of a ray
- **Line Segment Context** ⇒ Each segment has two endpoints determining its finite span and boundary
- **Ray Context** ⇒ A ray has exactly one endpoint from which it extends infinitely, defining its origin and direction
- **Angle Context** ⇒ In an angle, the shared endpoint of its two sides is the vertex, serving as the angle's hinge
- **Notation** ⇒ Labeled by a capital letter (e.g., A); segments use $\overline{AB}$, rays use $\overrightarrow{AB}$
- **Applications** ⇒ Determines segment length, constructs polygons, anchors vectors and network nodes

Standard Position — Cardinal Angle Glyph

- **Definition** ⇒ An angle placed with its vertex at $(0, 0)$ and its initial side along the positive x-axis
- **Vertex** ⇒ The origin $(0, 0)$, the common endpoint of both rays
- **Initial Side** ⇒ The fixed ray on the positive x-axis serving as the zero-angle reference
- **Terminal Side** ⇒ The rotating ray that sweeps from the initial side to form the angle
- **Rotation Direction** ⇒ Counterclockwise for positive measures; clockwise for negatives
- **Purpose** ⇒ Establishes a uniform reference for measuring angles in rectangular and polar systems

Initial Side — The Positive Ray Notation

- **Definition** $\leftarrow$ The set of points $(x, 0)$ with $x \geq 0$, i.e. the ray along the positive x-axis
- **Common Notation** $\leftarrow$ Often denoted $[0, \infty)$ on the real line but not a summation $\sum$ or integral $\int$
- **Role** $\leftarrow$ Acts as the zero-angle reference in standard position

Y-Axis — The Fixed Perpendicular Glyph

- **Definition** $\rightarrow$ The line of all points $(0, y)$ for real y, perpendicular to the x-axis
- **Fixed Role** $\rightarrow$ In the standard Cartesian frame it remains orthogonal to the initial side
- **Variants** $\rightarrow$ If you rotate the y-axis independently, you leave the standard rectangular system and enter a rotated or skewed frame

Ordered Pair — Cartesian Glyph

- **Definition** $\Rightarrow$ An ordered pair (x, y) designates a point with horizontal coordinate x and vertical coordinate y
- **Parentheses** $\Rightarrow$ Here the parentheses simply group the two coordinates—they are not interval delimiters and do not imply exclusion
- **Components** $\Rightarrow$ First component x is horizontal displacement; second component y is vertical displacement

Point on the X-Axis

- **Definition** $\Rightarrow$ All points of the form $(x, 0)$ lie exactly on the x-axis because $y = 0$
- **Range** $\Rightarrow$ In pure Cartesian form x may be any real number (including 0)
- **Initial-Side Context** $\Rightarrow$ For the standard-position initial side we further require $x \geq 0$

Interval Notation — Inclusive vs Exclusive

- **Definition** ⇒ Square brackets [ ] include endpoints; parentheses ( ) exclude them in real-number intervals
- **Clarification** ⇒ In $(x, 0)$ the parentheses are not interval notation but part of point notation
- **Implication** ⇒ To describe the ray you'd write $\{(x, 0) \mid x \geq 0\}$ or $[0, \infty)$ for the x-values

Reference Angle — The Acute Measuring Glyph

- **Definition** ⇒ The acute angle between the terminal side of an angle in standard position and the x-axis
- **Computation** ⇒ Subtract the angle from the nearest multiple of $90°$ or $\frac{\pi}{2}$ to yield $0° \leq \theta_{\text{ref}} \leq 90°$
- **Range** ⇒ Always between 0 and $90°$ (or 0 and $\frac{\pi}{2}$ radians) inclusive
- **Use** ⇒ Allows evaluation of trigonometric functions by referencing an acute angle with the same function value
- **Notation** ⇒ Denoted θ_{ref}

Coterminal Angle — The Rotational Glyph

- **Definition** ⇒ Angles that share the same initial and terminal sides when placed in standard position
- **Formula** ⇒ Given θ, coterminal angles are $\theta + 360°k$ or $\theta + 2\pi k$ for any integer k
- **Property** ⇒ They have identical sine, cosine, and tangent values because they land on the same point of the unit circle
- **Notation** ⇒ Written as $\theta \pm 360°n$ or $\theta \pm 2\pi n$
- **Application** ⇒ Used to find equivalent angles within a chosen interval (e.g., $[0, 360°)$ or $[0, 2\pi)$)

Reference Angle — The Acute Angle Glyph

- **Definition** ⇒ The acute angle between an angle's terminal side and the x-axis when in standard position
- **Calculation** ⇒ Take the absolute difference between the angle and the nearest multiple of 180° or π
- **Range** ⇒ Always satisfies $0^\circ \leq \theta_{\text{ref}} \leq 90^\circ$ (or $0 \leq \theta_{\text{ref}} \leq \frac{\pi}{2}$)
- **Purpose** ⇒ Allows evaluation of trig functions by referencing a corresponding acute angle
- **Notation** ⇒ Denoted θ_{ref}

Coterminal Angle — The Rotational Equivalence Glyph

- **Definition** ⇒ Angles sharing the same initial and terminal sides in standard position
- **Formula** ⇒ All coterminal angles are $\theta + 360^\circ k$ or $\theta + 2\pi k$ for integer k
- **Property** ⇒ They yield identical sine, cosine, and tangent values by landing on the same unit-circle point
- **Use Case** ⇒ Finds an equivalent angle within a principal interval (e.g., $[0, 360^\circ)$ or $[0, 2\pi)$)
- **Notation** ⇒ Written as $\theta \pm 360^\circ n$ or $\theta \pm 2\pi n$

Translation — The Shifting Glyph

- **Definition** ⇒ A Euclidean transformation that shifts every point of the triangle by the same vector
- **Operation** ⇒ Subtract the coordinates of the chosen vertex from all vertices so that vertex maps to $(0, 0)$
- **Vector** ⇒ If the vertex is (x_0, y_0), translate by $(-x_0, -y_0)$
- **Notation** ⇒ Denoted $T_{(-x_0, -y_0)}$
- **Purpose** ⇒ Places the triangle's vertex at the origin for standard positioning

Rotation — The Spinning Glyph

- **Definition** ⇒ A rigid transformation that rotates points about the origin by a fixed angle
- **Operation** ⇒ Compute the angle θ between the chosen base side and the positive x-axis and rotate by $-\theta$
- **Formula** ⇒ $(x', y') = (x\cos(-\theta) - y\sin(-\theta),\ x\sin(-\theta) + y\cos(-\theta))$
- **Notation** ⇒ Denoted $R_{-\theta}$
- **Purpose** ⇒ Aligns the chosen triangle side along the positive x-axis

Procedure — Positioning a Triangle in Standard Position

- **Step 1** ⇒ Translate the triangle so the chosen vertex goes to $(0, 0)$
- **Step 2** ⇒ Compute the angle between the chosen base side and the positive x-axis
- **Step 3** ⇒ Rotate the triangle by the negative of that angle to align the side with $x \geq 0$
- **Result** ⇒ The triangle sits with one vertex at the origin and one side on the positive x-axis, ready for trigonometric analysis

Pythagorean Theorem — Right-Triangle Side Calculation

- **Statement** ⇒ In a right triangle with legs a, b and hypotenuse c, $a^2 + b^2 = c^2$
- **Solve for Hypotenuse** ⇒ $c = \sqrt{a^2 + b^2}$
- **Solve for a Leg** ⇒ $a = \sqrt{c^2 - b^2}$ or $b = \sqrt{c^2 - a^2}$
- **Context** ⇒ Applies only when one angle is exactly $90°$

Trigonometric Ratios — Right-Triangle Angles and Sides

- **Definitions** ⇒ $\sin\theta = \frac{\text{opp}}{\text{hyp}}$, $\cos\theta = \frac{\text{adj}}{\text{hyp}}$, $\tan\theta = \frac{\text{opp}}{\text{adj}}$
- **Find an Angle** ⇒ $\theta = \arcsin\frac{\text{opp}}{\text{hyp}}$, or $\theta = \arccos\frac{\text{adj}}{\text{hyp}}$, or $\theta = \arctan\frac{\text{opp}}{\text{adj}}$
- **Context** ⇒ Requires one acute angle and one side known

Law of Cosines — General Triangle Side or Angle

- **Side Formula** $\Rightarrow$ $c^2 = a^2 + b^2 - 2ab\cos C$
- **Angle Formula** $\Rightarrow$ $\cos C = \frac{a^2+b^2-c^2}{2ab}$
- **Solve for Side** $\Rightarrow$ $c = \sqrt{a^2 + b^2 - 2ab\cos C}$
- **Solve for Angle** $\Rightarrow$ $C = \arccos(\frac{a^2+b^2-c^2}{2ab})$
- **Context** $\Rightarrow$ Works for any triangle when two sides and included angle are known (SAS) or three sides (SSS)

Law of Sines — General Triangle Angle or Side

- **Statement** $\Rightarrow$ $\dfrac{a}{\sin A} = \dfrac{b}{\sin B} = \dfrac{c}{\sin C}$
- **Find a Side** $\Rightarrow$ $a = b\frac{\sin A}{\sin B}$ or $c = b\frac{\sin C}{\sin B}$
- **Find an Angle** $\Rightarrow$ $A = \arcsin(\frac{a\sin B}{b})$
- **Context** $\Rightarrow$ Valid when any side–angle opposite pair is known plus another side or angle (ASA, AAS, SSA)

Law of Cosines Term — Interaction Glyph

- **Interaction Term** $\Rightarrow$ $2ab\cos C$ is twice the product of sides a and b multiplied by the cosine of the included angle C
- **Role** $\Rightarrow$ It adjusts the sum $a^2 + b^2$ to account for the tilt between those sides
- **Sign** $\Rightarrow$ The minus sign in $a^2 + b^2 - 2ab\cos C$ reduces the sum when $0° < C < 90°$ and increases it when $90° < C < 180°$
- **Full Formula** $\Rightarrow$ $c^2 = a^2 + b^2 - 2ab\cos C$
- **Special Case** $\Rightarrow$ When $C = 90°$, $\cos C = 0$ and it collapses to $c^2 = a^2 + b^2$ (the Pythagorean theorem)

Interaction Term — Angular Adjustment Glyph

- **Geometric Origin** $\Rightarrow$ Derived from the dot product of two sides of lengths a and b
- **Adjustment Role** $\Rightarrow$ Modifies $a^2 + b^2$ by the projection of one side onto the other
- **Cosine Function** $\Rightarrow$ $\cos C$ measures alignment of sides, not quadrant or 360° normalization
- **Sign Impact** $\Rightarrow$ For $0^\circ < C < 90^\circ$, $\cos C > 0$ subtracts; for $90^\circ < C < 180^\circ$, $\cos C < 0$ subtracting a negative adds
- **Purpose** $\Rightarrow$ Ensures the correct length of the third side for both acute and obtuse interior angles

Alignment Context — The Vector Alignment Glyph

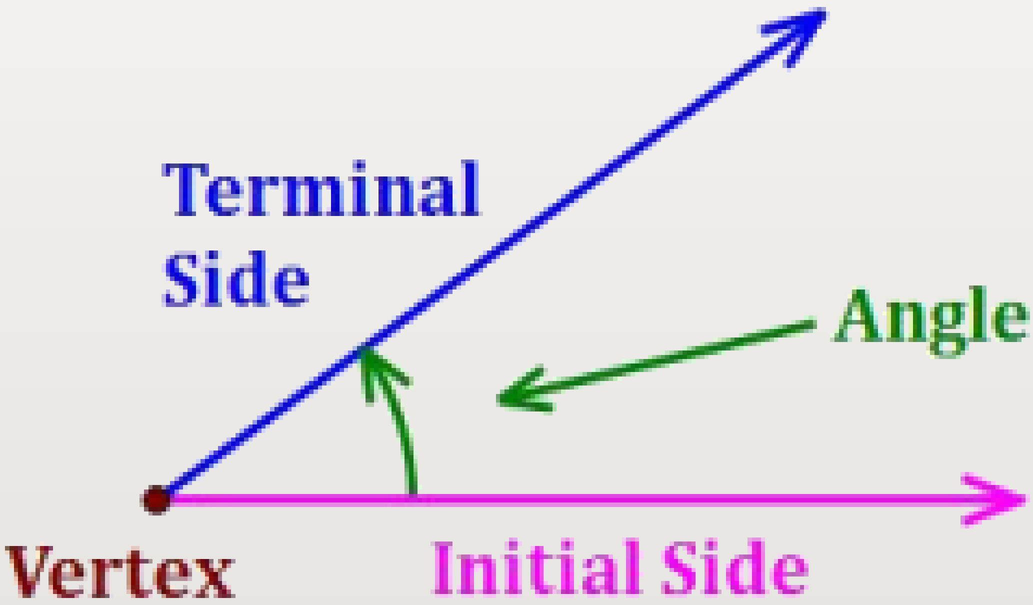
- **Definition** $\Rightarrow$ Refers to the angle between the two side-vectors of lengths a and b at their common vertex
- **Vertex as Origin** $\Rightarrow$ In the dot-product derivation we translate that vertex to $(0, 0)$ so both sides become vectors from the origin
- **Cosine Measure** $\Rightarrow$ $\cos C$ quantifies how much one side "leans" toward the other at that vertex
- **Coordinate Independence** $\Rightarrow$ This alignment is intrinsic to the triangle and doesn't depend on the global x–y axes unless you choose standard position
- **Purpose** $\Rightarrow$ Ensures the third side's length reflects the exact tilt between sides a and b

Labeling Convention — Angle vs Side Glyph

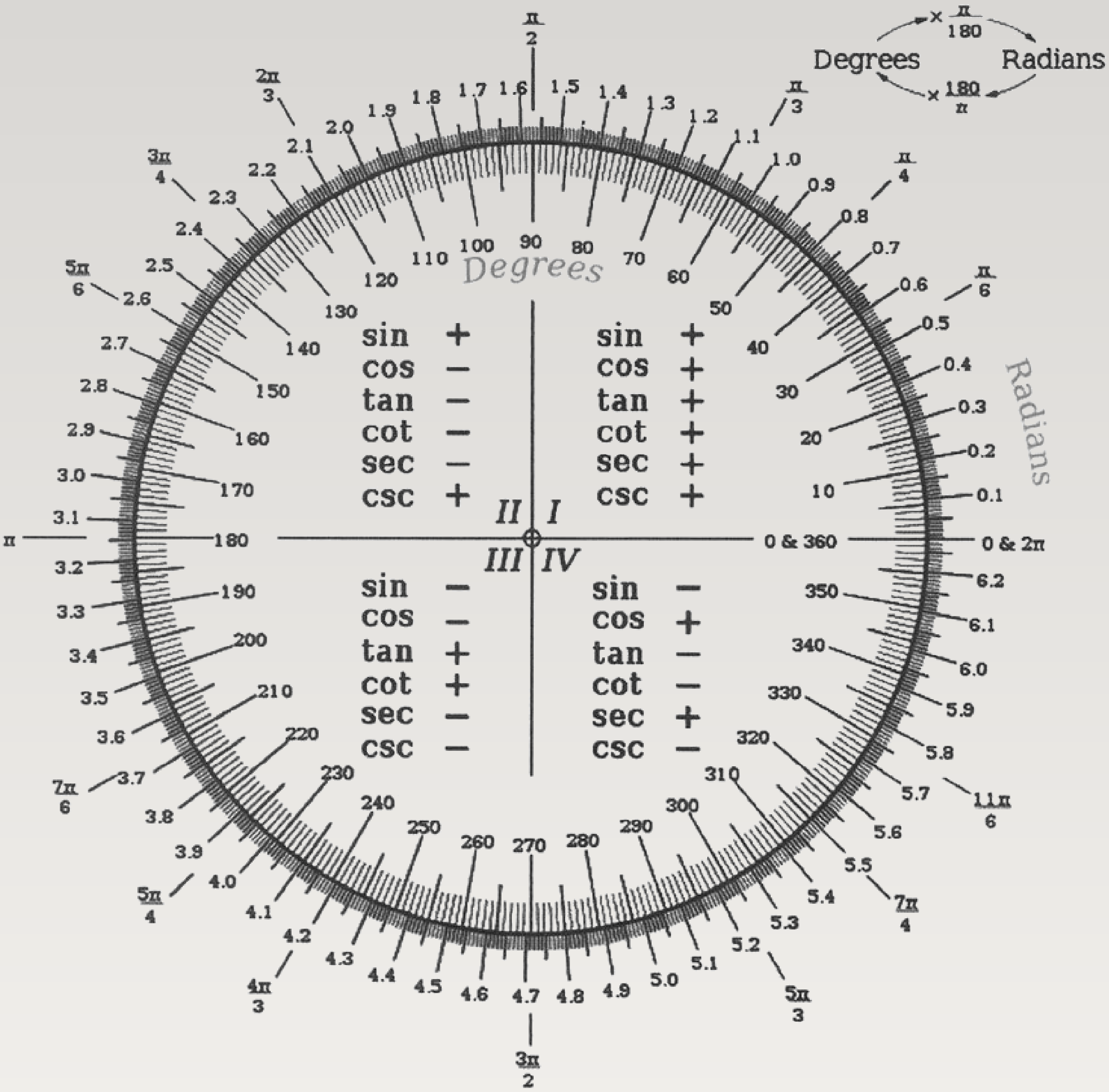
- **Convention** $\Rightarrow$ Angles are denoted by uppercase letters (e.g., A, B, C)
- **Sides** $\Rightarrow$ Sides are denoted by lowercase letters (e.g., a, b, c)
- **Opposition** $\Rightarrow$ Side a is opposite angle A; side b opposite B; side c opposite C
- **Triangle Context** $\Rightarrow$ In $\triangle ABC$, vertices A, B, C label angles; sides BC, CA, AB label a, b, c respectively
- **Usage** $\Rightarrow$ This convention ensures formulas like the Law of Sines and Law of Cosines remain consistent

Polar Axis Polar Angle
coordinate system Rectangular and Polar

V Defining Angles



V Trigonometric Functions



Triangle Definition — The Ritual Glyph

- **Vertices** ⇒ A,B,C
- **Sides** ⇒ a=BC=3, b=CA=4, c=AB=5
- **Right Angle** ⇒ At vertex C since $3^2 + 4^2 = 5^2$
- **Goal** ⇒ Find angles A,B; place in standard position; get coordinates; compute trig ratios

Law of Cosines for Angles — The Audit Glyph

- **Formula A** ⇒ $A = \arccos\left(\frac{b^2+c^2-a^2}{2bc}\right) = \arccos\left(\frac{4^2+5^2-3^2}{2\cdot4\cdot5}\right)$
- **Formula B** ⇒ $B = \arccos\left(\frac{a^2+c^2-b^2}{2ac}\right) = \arccos\left(\frac{3^2+5^2-4^2}{2\cdot3\cdot5}\right)$
- **Numeric A** ⇒ $A = \arccos(0.8) \approx 36.87°$
- **Numeric B** ⇒ $B = \arccos(0.6) \approx 53.13°$
- **Check** ⇒ $A + B + C = 36.87 + 53.13 + 90 = 180°$

Standard Position — The Containment Glyph

- **Step 1** ⇒ Translate A→(0,0)
- **Step 2** ⇒ Rotate AB onto positive x-axis ⟹ B→(5,0)
- **Step 3** ⇒ C becomes $(b\cos A,\ b\sin A) = (4\cos 36.87°,\ 4\sin 36.87°)$
- **Result** ⇒ A=(0,0), B=(5,0), C≈(3.2, 2.4)

Trig Ratios — The Projection Glyph

- **sin A** ⇒ $\frac{\text{opp}}{\text{hyp}} = \frac{3}{5} = 0.6$
- **cos A** ⇒ $\frac{\text{adj}}{\text{hyp}} = \frac{4}{5} = 0.8$
- **tan A** ⇒ $\frac{\text{opp}}{\text{adj}} = \frac{3}{4} = 0.75$
- **sin B** ⇒ $\frac{4}{5} = 0.8$
- **cos B** ⇒ $\frac{3}{5} = 0.6$
- **tan B** ⇒ $\frac{4}{3} \approx 1.3333$

V Pythagorean Helper

Pythagorean Helper — The Hypotenuse Glyph

- **Function** ⇒ `r_nx_pth(x,y)` returns $\sqrt{x^2 + y^2}$
- **Inputs** ⇒ x,y represent the two legs of a right triangle
- **Output** ⇒ Hypotenuse length c
- **Role** ⇒ Provides c for subsequent trig-ratio calculations

r_nx_pth(x, y)

<MINTED>

V SOHCAHTOA Solver

SOHCAHTOA Solver — The Angle Extraction Glyph

- **Definition** ⇒ `nx_solved_sohcahtoa(a,b)` solves for angle A
- **Compute c** ⇒ $c = r\_nx\_pth(a, b)$
- **Build ratios** ⇒ $\sin A = \frac{b}{c}$; $\cos A = \frac{a}{c}$; $\tan A = \frac{b}{a}$
- **Inverse trig** ⇒ Uses `r_nx_ts_asin`, `r_nx_ts_acos`, `r_nx_ts_atan` then `nx_rad2deg`
- **Output** ⇒ Prints angle A in degrees three times from each ratio

nx_solved_sohcahtoa(a, b)

<MINTED>

Inverse Trig Conversion — The Angle Extraction Glyph

- **Purpose** ⇒ Convert a sine, cosine, or tangent ratio back into its corresponding angle
- **asin** ⇒ Arc-sine inverts sin: finds θ such that $\sin\theta = \text{ratio}$
- **acos** ⇒ Arc-cosine inverts cos: finds θ such that $\cos\theta = \text{ratio}$
- **atan** ⇒ Arc-tangent inverts tan: finds θ such that $\tan\theta = \text{ratio}$
- **rad2deg** ⇒ Transforms the radian-output of the inverse function into degrees for readability

Ratio Context — The Side-Ratio Glyph

- **x = b/c** ⇒ Opposite side (BC=b) over hypotenuse (c=AB) for angle A
- **y = a/c** ⇒ Adjacent side (CA=a) over hypotenuse (c=AB) for angle A
- **z = b/a** ⇒ Opposite side (b) over adjacent side (a) for angle A
- **SOHCAHTOA** ⇒ These exactly match $\sin A$, $\cos A$, and $\tan A$ respectively
- **Inverse Trig Input** ⇒ You feed each ratio into arcsin, arccos, or arctan to recover angle A

Ratio Domain & Range - The Dimensionless Glyph

- **What "ratio" means** ⇒ A pure, dimensionless fraction of two side-lengths in a triangle
- $\sin\theta$ ⇒ Opposite side / hypotenuse => always between -1 and $+1$
- $\cos\theta$ ⇒ Adjacent side / hypotenuse => always between -1 and $+1$
- $\tan\theta$ ⇒ Opposite side / adjacent side => any real number (adjacent can approach zero)
- **Not "to π"** ⇒ These ratios are not measured "to π" or any other unit—they are just length / length
- **Inverse-Trig Inputs** ⇒ `arcsin`/`arccos` take inputs in $[-1, 1]$; `arctan` accepts all real ratios

V Arcsin Taylor Series

pr_nx_ts_asin(z, p)

<MINTED>

Domain Checks & Edge Cases — The Purity Glyph

- **Domain Guard** ⇒ Rejects |z|>1 since asin is only valid on [-1,1]
- **Impurity Notice** ⇒ Prints an error if domain is breached
- **Exact Ends** ⇒ If |z|=1 returns $\pm \frac{c\_pi}{2}$ exactly
- **Sign Function** ⇒ Uses $nx\_sign(z)$ to pick + or -
- **Setup p** ⇒ Keeps 'p' as the series depth (initially 128)

Taylor Series Expansion — The Recurrence Glyph

- **Goal** ⇒ Compute $\mathrm{asin}(z) = \sum_{n=0}^{\infty} \frac{(2n)!}{4^n\,(n!)^2\,(2n+1)}\, z^{2n+1}$
- **Accumulator r** ⇒ Starts at r=z (n=0 term)
- **Term t** ⇒ Tracks z^{2n+1}; updated via $t \leftarrow t{\cdot}z^2$
- **Coeff s** ⇒ Tracks $\frac{(2n)!}{4^n\,(n!)^2}$ via recurrence $s \leftarrow s{\cdot}\frac{2n-1}{2n}$
- **Series Add** ⇒ Adds each term $s{\cdot}t/(2n+1)$ to r
- **Loop Count** ⇒ Runs from n=1 to p−1 terms for desired accuracy

Final Return — The Resultant Glyph

- **Output** ⇒ Returns r as the approximate $\mathrm{asin}(z)$ in radians
- **Wrapper** ⇒ r_nx_ts_asin(z) calls pr_nx_ts_asin(z,128) by default
- **Convert to Degrees** ⇒ Use nx_rad2deg on the result if you need degrees

Arcsin Taylor Series — The Coefficient Glyph

- **Series Definition** ⇒ $\arcsin(z) = \sum_{n=0}^{\infty} \frac{(2n)!}{4^n\,(n!)^2\,(2n+1)}\, z^{2n+1}$
- **Coefficient** ⇒ $C_n = \frac{(2n)!}{4^n\,(n!)^2}$
- **Term** ⇒ $T_n = C_n\,\frac{z^{2n+1}}{2n+1}$
- **Goal** ⇒ Compute C_n via recurrence, avoiding fresh factorials each time

Recurrence Derivation — The Pure Ratio Glyph

- **Definition** ⇒ $C_n = \dfrac{(2n)!}{4^n\,(n!)^2}$
- **Ratio C_n/C_{n-1}** ⇒ $\dfrac{C_n}{C_{n-1}} = \dfrac{(2n)!/4^n\,(n!)^2}{(2n-2)!/4^{n-1}\,\left((n-1)!\right)^2} = \dfrac{(2n)(2n-1)}{4\,n^2}$
- **Simplify** ⇒ $\dfrac{(2n)(2n-1)}{4\,n^2} = \dfrac{2n-1}{2n}$
- **Recurrence** ⇒ $C_n = C_{n-1} \times \dfrac{2n-1}{2n}$

Code Mapping — The Implementation Glyph

- **Loop Index** ⇒ `i` corresponds to n
- **Accumulator** ⇒ `s` holds C_{n-1} at loop start
- **Update** ⇒ `s = s * (2*i - 1) / (2*i)` implements $s \leftarrow s \cdot \frac{2n-1}{2n}$
- **Initialization** ⇒ Before loop: `s` $= C_0 = 1$
- **Result** ⇒ After n iterations: `s` $= C_n$

Coefficient Recurrence in Code

<MINTED>

V Arctan Taylor Series

nx_ts_alt(z, i, p, k, s)

<MINTED>

Arctan Taylor Series — The Alternating Glyph

- **Series Definition** ⇒ $\arctan(z) = \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} z^{2n+1}$
- **Accumulator r** ⇒ Starts at z for $n = 0$
- **Power t** ⇒ Tracks z^{2n+1}; updated via $t \leftarrow t \cdot z^2$
- **Sign s** ⇒ Alternates via $s \leftarrow -s$ to implement $(-1)^n$
- **Odd Index i** ⇒ Uses $i = 2n + 1$; increments by $k = 2$ each loop

Series Engine — The Loop Glyph

- **Init Parameters** ⇒ Called with $(z,\ 3,\ p,\ 2,\ -1)$
- **Start i=3** ⇒ First loop adds term for $n = 1$: $-\ z^3/3$
- **Step k=2** ⇒ Moves from $i = 3 \to 5 \to 7$ ...
- **Term Update** ⇒ $r \mathrel{+}= s\,t/i$
- **Loop Count** ⇒ Runs until $i \geq p$ for desired precision

Argument Reduction — The Reciprocal Glyph

- **Large |z|>1** ⇒ Uses $\arctan(z) = \frac{\pi}{2} - \arctan(1/z)$ if $z > 1$
- **Negative / Large** ⇒ Uses $\arctan(z) = -\frac{\pi}{2} - \arctan(1/z)$ if $z < -1$
- **Reduce Magnitude** ⇒ Calls $\texttt{nx\_ts\_alt}(1/z, 3, p, 2, -1)$
- **Ensures Fast Convergence** ⇒ Keeps series input $|1/z| < 1$
- **Assemble** ⇒ Adds or subtracts $\pm\frac{\pi}{2}$ to the reduced-series result

Domain Guard — The Purity Glyph

- **Check** $\Rightarrow$ if $|z| > 1 \longrightarrow$ print impurity and return -1
- **Purpose** $\Rightarrow$ Prevent divergence: the alternating series for $\arctan(z)$ only converges for $|z| \leq 1$
- **Signal** $\Rightarrow$ Returning -1 flags an invalid input before any looping begins

Alternating Series Engine — The Loop Glyph

- **Accumulator** r $\Rightarrow$ Starts at z (the $n = 0$ term of $\sum(-1)^n z^{2n+1}/(2n+1)$)
- **Power** t $\Rightarrow$ Tracks z^{2n+1}; updated via $t \leftarrow t \cdot z^2$
- **Sign** s $\Rightarrow$ Implements $(-1)^n$; flips each iteration ($s \leftarrow -s$)
- **Index** i $\Rightarrow$ Denominator for each term; begins at first odd index passed in, increments by k (usually 2)
- **Loop** $\Rightarrow$ While $i < p$: add $s \cdot t/i$ to r, flip sign, step index
- **Result** $\Rightarrow$ After i reaches p, r approximates $\arctan(z)$ in radians

pr_nx_ts_atan(z, p)

```
&lt;MINTED&gt;
```

V Generic Series Engine

nx_mc_esp(x, y) & nx_ts(n, t, p, k, s)

```
&lt;MINTED&gt;
```

Domain Guard — The Purity Glyph

- **Check** $\Rightarrow$ if $|n| > 1$ prints impurity & returns -1
- **Reason** $\Rightarrow$ Taylor-type series converge only for $|n| \leq 1$
- **Signal** $\Rightarrow$ Early exit prevents divergent summation

Parameter Guard — The Integrity Glyph

- **Check** $\Rightarrow$ if $t < 1$, $k \leq 0$, or $p \leq t$ prints impurity & returns -1
- **Reason** $\Rightarrow$ Ensures valid start index, positive step, and loop limit
- **Signal** $\Rightarrow$ Prevents infinite loops or empty sums

Generic Series Engine — The Paired-Term Glyph

- **Accumulator** r $\Rightarrow$ Initialized to s (first term)
- **Loop** $\Rightarrow$ While $t < p$: subtract term at t, then add term at $t + k$
- **Index Update** $\Rightarrow$ t increases by k twice per iteration
- **Result** $\Rightarrow$ After loop, r holds paired-term sum approximation

Term Generator — The Exponential Glyph

- **nx_esp(x,y)** $\Rightarrow$ Computes $\frac{x^y}{y!}$
- **Purpose** $\Rightarrow$ Yields the generic y-th Maclaurin term
- **Flexibility** $\Rightarrow$ Used for e.g. exp, sin, cos with appropriate k, s

Series Pattern — The Alternation Glyph

- **Sign** s $\Rightarrow$ Initial sign of first term; alternation emerges via subtraction/addition
- **Pairing** $\Rightarrow$ Groups two successive terms per loop: $-\frac{n^t}{t!} + \frac{n^{t+k}}{(t+k)!}$
- **Use Case** $\Rightarrow$ Efficient for expansions with only even/odd or paired exponents

Flexibility & Efficiency — The Duality Glyph

- **Generic vs Specialized** ⇒ nx_ts is universal; nx_ts_alt optimizes a single alternating series
- **Trade-off** ⇒ nx_ts calls nx_mc_esp each term (slower) but handles any factorial-based series
- **Parameterization** ⇒ By choosing (n, t, p, k, s) you tailor convergence rate, term pattern, ± signs

V Arccos Taylor Series

pr_nx_ts_acos(x, y)

<MINTED>

Domain Guard — The Purity Glyph

- **Check** ⇒ if $|x| > 1$ prints impurity & returns -1
- **Reason** ⇒ $\mathrm{acos}(x)$ is only defined for $|x| \leq 1$
- **Signal** ⇒ Early exit prevents invalid series calls

Complementary Identity — The Complementary Angle Glyph

- **Mathematical Identity** ⇒ $\mathrm{acos}(x) = \frac{\pi}{2} - \mathrm{asin}(x)$
- **Implementation** ⇒ Subtracts the output of `pr_nx_ts_asin(x,y)` from $\frac{\pi}{2}$
- **c_pi** ⇒ Constant representing π in the codebase

Wrapper Behavior — The Resultant Glyph

- **Delegation** ⇒ Reuses the Taylor-series engine of asin(x) instead of a new series for acos(x)
- **Efficiency** ⇒ Avoids writing a separate expansion for acos(x)
- **Final Output** ⇒ Returns acos(x) in radians as $c\_pi/2 - \text{asin}(x)$

V Side-Side-Side

r_nx_solve_sss(a, b, c)

<MINTED>

SSS Acronym — The Side-Side-Side Glyph

- **SSS** ⇒ Side-Side-Side triangle: all three side lengths are given
- **Function Meaning** ⇒ `r_nx_solve_sss` signals solving a triangle by SSS
- **Return** ⇒ Angle opposite side c (between sides a and b)

Law of Cosines Implementation — The Cosine-Angle Glyph

- **Cosine Rule** ⇒ $\cos(C) = \dfrac{a^2 + b^2 - c^2}{2ab}$
- **Ratio** ⇒ Computes $(a^2 + b^2 - c^2)/(2ab)$
- **Inverse Cos** ⇒ Calls `r_nx_ts_acos` on that ratio
- **Output** ⇒ Angle C in radians as $\arccos\left(\frac{a^2+b^2-c^2}{2ab}\right)$

nx_solved_sss(a, b, c)

<MINTED>

SSS Triangle Solver — The Full Angle Glyph

- **SSS** ⇒ Side-Side-Side: all three side lengths a, b, c are known
- **Goal** ⇒ Compute all three internal angles A, B, C
- **Method** ⇒ Uses Law of Cosines via `r_nx_solve_sss`
- **Angle A** ⇒ Opposite side a, computed from sides b, c
- **Angle B** ⇒ Opposite side b, computed from sides c, a
- **Angle C** ⇒ Opposite side c, computed from sides a, b

Triangle Inequality Guard — The Purity Glyph

- **Check** ⇒ If any side sum $\leq$ third side, prints impurity and returns -1
- **Purpose** ⇒ Ensures the three sides can form a valid triangle
- **Signal** ⇒ Prevents degenerate or invalid geometry

Angle Summation — The Closure Glyph

- **Sum** ⇒ Adds angles $x + y + z$ in radians and degrees
- **Check** ⇒ Verifies triangle closure: sum should equal π radians or 180°
- **Debug** ⇒ Useful for detecting numerical drift or impurity in series approximations

V Side-Angle-Side

r_nx_solve_sas(a, b, c)

```
<MINTED>
```

SAS Acronym — The Side-Angle-Side Glyph

- **SAS** ⇒ Side-Angle-Side: two sides a, b and included angle c define the triangle
- **Uniqueness** ⇒ Given a, b > 0 and 0 < c < pi, the third side is determined uniquely

Domain Guards — The Purity Glyph

- **Positive Sides** ⇒ Checks a, b > 0; prints impurity if violated
- **Angle Range** ⇒ Ensures 0 < c < c_pi; prints impurity if violated
- **Negative Square** ⇒ Detects x < 0 after law of cosines to avoid sqrt of negative

Law of Cosines — The Cosine-Angle Glyph

- **Formula** ⇒ $c^2 = a^2 + b^2 - 2\,a\,b\,\cos(c)$
- **Implementation** ⇒ Computes x = $a^2 + b^2$ - 2ab·r_nx_ts_cos(c)
- **Inverse Cos** ⇒ Calls the series-based cosine via r_nx_ts_cos

Output & Usefulness — The Utility Glyph

- **Result** ⇒ Returns $\sqrt{x}$ as the third side length
- **Applications** ⇒ Triangle solving in engineering, surveying, graphics, mechanics
- **Efficiency** ⇒ Reuses cosine series routine; single sqrt for final result

V Trig Sign Resolver

nx_sign_trig(x, id)

<MINTED>

Trig Sign Resolver — The Quadrant Glyph

- **Purpose** ⇒ Returns the sign (+/−) of $\sin(x)$, $\cos(x)$, or $\tan(x)$ based on angle x
- **Input** ⇒ Angle x in radians; id = 0 for sin, 1 for cos, 2 for tan
- **Reduction** ⇒ Uses `nx_mod2pi(x)` to fold angle into $[0, 2\pi)$
- **Quadrant** ⇒ Divides circle into 4 zones: Q1, Q2, Q3, Q4
- **Sign Table** ⇒ $\sin(x)$: + in Q1/Q2, – in Q3/Q4;
  $\cos(x)$: + in Q1/Q4, – in Q2/Q3;
  $\tan(x)$: + in Q1/Q3, – in Q2/Q4
- **Use Case** ⇒ Needed when computing signs of trig values without evaluating full series

V Angle Reduction

nx_ang_cos(x) & nx_ang_sin(x) & nx_ang_tan(x)

<MINTED>

Angle Reduction — The Tangent–Sine Glyph

- **nx_ang_sin(x)** ⇒ Reduces angle x to signed acute form for sine: returns value in $[-90^\circ, +90^\circ]$
- **nx_ang_tan(x)** ⇒ Delegates to `nx_ang_sin(x)` because tangent shares sine's sign and symmetry
- **Why?** ⇒ $\tan(x) = \sin(x)/\cos(x)$, so its sign matches sine when cosine is positive
- **Quadrant Behavior** ⇒ Tangent is positive in Q1/Q3, negative in Q2/Q4 — same as sine's sign flip across 180°
- **Use Case** ⇒ This reduction is for sign-aware series or quadrant-aware approximations, not full evaluation

Tangent Symmetry — The Identity Glyph

- **Odd Function** ⇒ $\tan(-x) = -\tan(x)$
- **Periodicity** ⇒ $\tan(x + 180°) = \tan(x)$
- **Reduction** ⇒ Reflecting across 180° preserves tangent's value and sign
- **Acute Mapping** ⇒ Mapping to $[-90°, +90°]$ is sufficient for sign and series convergence
- **Efficiency** ⇒ Avoids duplicating logic — sine's reduction already handles quadrant sign flips

V Cosine and Secant Series Wrappers

Cosine & Secant Series Wrappers

```
<MINTED>
```

V Reciprocal Trig Solver

nx_solved_shoahatao(a, b, c)

```
<MINTED>
```

Reciprocal Trig Solver — The Inverse Glyph

- **Input** ⇒ Legs a, b; hypotenuse c (optional)
- **Auto-Hypotenuse** ⇒ If $c = 0$, computes $c = \sqrt{a^2 + b^2}$
- **csc(A)** ⇒ $\csc(A) = \frac{c}{b} \Rightarrow A = \arcsin\left(\frac{1}{\csc(A)}\right)$
- **sec(A)** ⇒ $\sec(A) = \frac{c}{a} \Rightarrow A = \arccos\left(\frac{1}{\sec(A)}\right)$
- **cot(A)** ⇒ $\cot(A) = \frac{a}{b} \Rightarrow A = \arctan\left(\frac{1}{\cot(A)}\right)$

Purpose — The Reciprocal Recovery Glyph

- **Goal** ⇒ Recover angle A from reciprocal trig ratios
- **Use Case** ⇒ When given triangle sides and needing angle via csc, sec, cot
- **Series Engines** ⇒ Uses `r_nx_ts_asin`, `r_nx_ts_acos`, `r_nx_ts_atan`
- **Output** ⇒ Prints each ratio and recovered angle in degrees

V Core Tangent and Cotangent Wrappers

Core Tangent & Cotangent Wrappers

<MINTED>

Tangent Series Wrapper — The Ratio Glyph

- **Definition** ⇒ $\tan(x) = \frac{\sin(x)}{\cos(x)}$
- **Implementation** ⇒ `pr_nx_tan(x, p)` calls sine and cosine series engines
- **Wrapper** ⇒ `r_nx_tan(x)` uses default precision $p = 128$
- **Quadrant Awareness** ⇒ Sign logic inherited from sine/cosine wrappers
- **Use Case** ⇒ Direct tangent evaluation via series purity

Cotangent Series Wrapper — The Reciprocal Glyph

- **Definition** ⇒ $\cot(x) = \frac{1}{\tan(x)}$
- **Implementation** ⇒ `pr_nx_ts_cot(x, p)` returns reciprocal of tangent
- **Wrapper** ⇒ `r_nx_ts_cot(x)` uses default precision $p = 128$
- **Impurity Risk** ⇒ Caller must ensure $\tan(x) \neq 0$
- **Use Case** ⇒ Cotangent recovery without separate series engine

V Tangent Identity Expansion

Tangent Identity Expansion

<MINTED>

Secant Identity Wrapper — The Pythagorean Glyph

- **Identity** ⇒ $\sec^2(x) = 1 + \tan^2(x)$
- **Implementation** ⇒ `pr_nx_tan_sec(x, p)` computes $\tan(x)$, squares it, adds 1
- **Wrapper** ⇒ `r_nx_tan_sec(x)` uses default precision $p = 128$
- **Use Case** ⇒ Symbolic derivative of $\tan(x)$, identity validation, secant recovery
- **Efficiency** ⇒ Avoids direct cosine evaluation near $\frac{\pi}{2}$

V Sine Series Wrapper

pr_nx_ts_sin(x, p)

<MINTED>

Sine Series Wrapper — The Odd-Power Glyph

- **Function** ⇒ `pr_nx_ts_sin(x, p)`
- **Reduction** ⇒ `nx_rad_sin(x)` reduces angle to acute form
- **Series** ⇒ Calls `nx_ts(x, 3, p, 2, x)` to sum odd powers
- **Pattern** ⇒ $\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \cdots$
- **Wrapper** ⇒ `r_nx_ts_sin(x)` uses default precision $p = 100$

V Cosecant Wrapper

pr_nx_ts_csc(x, y)

<MINTED>

Cosecant Wrapper — The Reciprocal Glyph

- **Function** ⇒ `pr_nx_ts_csc(x, y)` = $1/\sin(x)$
- **Series** ⇒ Delegates to `pr_nx_ts_sin(x, y)`
- **Wrapper** ⇒ `r_nx_ts_csc(x)` uses default precision $y = 128$
- **Impurity Risk** ⇒ Caller must ensure $\sin(x) \neq 0$

pr_nx_sincos(x, y)

<MINTED>

Identity Audit — The Unit Circle Glyph

- **Function** ⇒ `pr_nx_sincos(x, y)`
- **Evaluation** ⇒ Computes $\sin^2(x) + \cos^2(x)$
- **Expected** ⇒ Returns ≈ 1 for all valid x
- **Use Case** ⇒ Purity check, identity validation, numerical drift detection

Trig Identity Overlay — The Ratio Glyph

- $\tan(x)$ ⇒ $\tan(x) = \dfrac{\sin(x)}{\cos(x)}$
- $\cot(x)$ ⇒ $\cot(x) = \dfrac{1}{\tan(x)}$
- $\sec^2(x)$ ⇒ $\sec^2(x) = 1 + \tan^2(x)$
- $\sin^2 + \cos^2$ ⇒ $\sin^2(x) + \cos^2(x) = 1$

VI Formulas

$$\text{floor}(x) = x - (x \mod 1) \quad \text{if } x \geq 0$$

floor(x) = x - (x $\mod 1) - 1 \quad$ if $x < 0$ and $x \mod 1 \neq 0$

VII Series