**Akima 1D Interpolation**

Akima 1D interpolation is a method for performing interpolation on one-dimensional data using the Akima algorithm. The Akima interpolation is a type of piecewise cubic Hermite interpolation, which means it constructs a piecewise polynomial function that passes through each data point while maintaining continuity and smoothness.

The Akima algorithm calculates the derivatives at each data point using a weighted average of the slopes of neighboring data points. These derivatives are then used to construct cubic polynomials within each segment, resulting in a smooth interpolation curve that avoids oscillations.

The mathematical steps involved in Akima 1D interpolation can be summarized as follows:

1. Given a set of data points (x[i], y[i]) for i = 0, 1, ..., N-1, where x[i] represents the independent variable and y[i] represents the dependent variable.

2. Calculate the slopes or derivatives at each data point, denoted as m[i]. The slope at a point is calculated using the neighboring points. Specifically, for a point (x[i], y[i]), the slope m[i] is calculated as:

m[i] = w1 \* (y[i+1] - y[i-1]) / (x[i+1] - x[i-1]) + w2 \* (y[i+2] - y[i]) / (x[i+2] - x[i]) (Eq. 1)

where w1 and w2 are weights that depend on the spacing between the neighboring points.

3. Interpolate between data points using cubic polynomials. Within each segment (x[i], x[i+1]), the cubic polynomial is constructed based on the data points (x[i-1], y[i-1]), (x[i], y[i]), (x[i+1], y[i+1]), and (x[i+2], y[i+2]).

4. The cubic polynomial for the segment is given by:

P(x) = y[i] + m[i] \* (x - x[i]) + A \* (x - x[i])^2 + B \* (x - x[i])^3 (Eq. 2)

where A and B are coefficients specific to the segment and are calculated to ensure smoothness and continuity.

5. Repeat steps 3 and 4 for each segment to construct the complete interpolation curve.

**PchipInterpolation**

PchipInterpolation, short for Piecewise Cubic Hermite Interpolation, is a method for performing interpolation on one-dimensional data using a piecewise cubic Hermite spline. PchipInterpolation constructs a smooth and monotonic curve that passes through the given data points.

The mathematics behind PchipInterpolation involves the following steps:

1. Given a set of data points (x[i], y[i]) for i = 0, 1, ..., N-1, where x[i] represents the independent variable and y[i] represents the dependent variable.

2. Calculate the slopes or derivatives at each data point, denoted as m[i]. The slopes are calculated using a shape-preserving algorithm to ensure monotonicity. It guarantees that if y[i] ≤ y[i+1], then m[i] ≤ m[i+1].

3. Interpolate between data points using cubic polynomials. Within each segment (x[i], x[i+1]), the cubic polynomial is constructed based on the data points (x[i], y[i]), (x[i+1], y[i+1]), and the corresponding slopes (m[i], m[i+1]).

4. The cubic polynomial for the segment is given by:

P(x) = h00(x) \* y[i] + h10(x) \* h \* m[i] + h01(x) \* y[i+1] + h11(x) \* h \* m[i+1] (Eq. 3)

where h = x[i+1] - x[i] is the interval length, and h00(x), h10(x), h01(x), and h11(x) are the basis functions.

The basis functions are defined as follows:

h00(x) = 2 \* t^3 - 3 \* t^2 + 1;

h10(x) = t^3 - 2 \* t^2 + t;

h01(x) = -2 \* t^3 + 3 \* t^2;

h11(x) = t^3 - t^2;

where t = (x - x[i]) / h is the normalized position within the segment.

5. Repeat steps 3 and 4 for each segment to construct the complete interpolation curve.