Interpolation

JrPhy

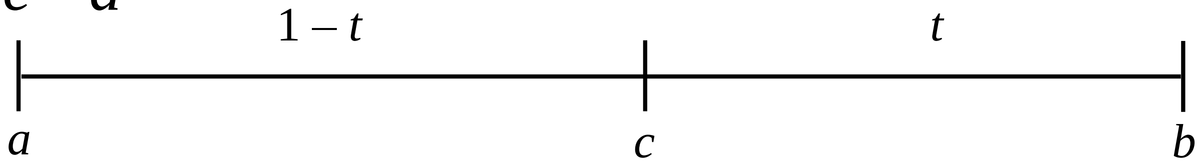
Introduction

- Interpolation is a method of constructing new data points within the range of a discrete set of known data points.
- In this branch, I’ll introduce some method for interpolation:
  - Linear interpolation
  - Lagrange interpolation
  - Spline interpolation
  - Least-square interpolation

Linear Interpolation

- This is the easiest method and everyone has learned it in senior high in Taiwan. Suppose there are two point *a*, *b* in one dimension, then the distance between two points is $|a - b|$, $||$ is absolute value. And *c* is in (a, b), *c* can be determine by *a*, *b* as the figure.
- Suppose $b > c > a$, and $b - \mathrm{c} = t$, $c - a = 1 - t$, then

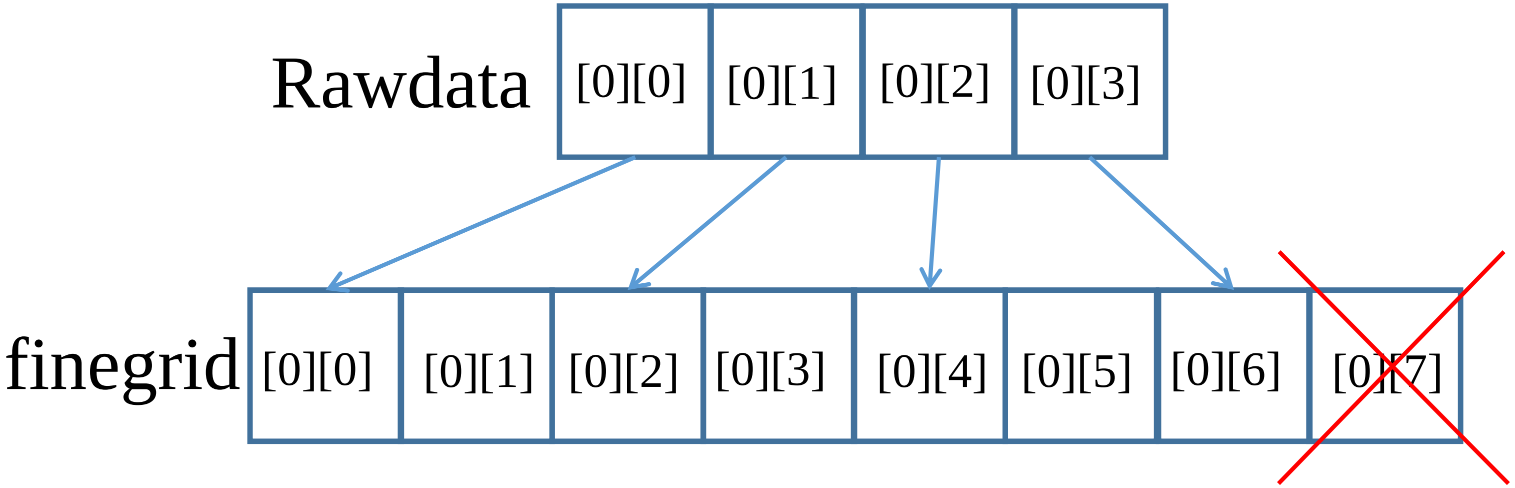
$$\frac{t}{1-t}=\frac{b-c}{c-a}\rightarrow tc-ta=(1-t)b-(1-t)c\rightarrow c=(1-t)b+ta$$



- Note that *a* times $(b - \mathrm{c})$ and *b* times $(c - a)$, it means f c is close to *a*, then the number of *c* is close to *a*, too. So *t* can be seen as the “weight”
- If $t = 0.5$, then $c = 0.5a + 0.5b$, means the mid-point.

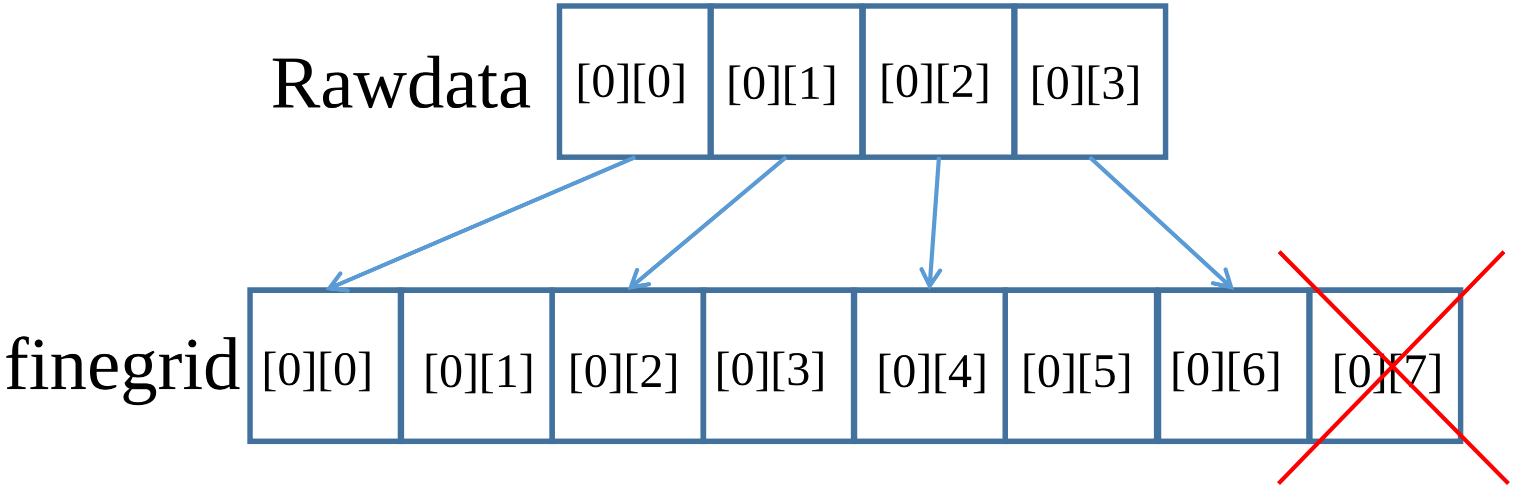
Linear Interpolation

1st we interpolate *x* axis



Suppose there are *m* data in rawdata, there are (*m*–1) interval, and want to insert *n* data in each interval. After inserting the data, the interval increases.

Linear Interpolation



For example, *insert* = 1,

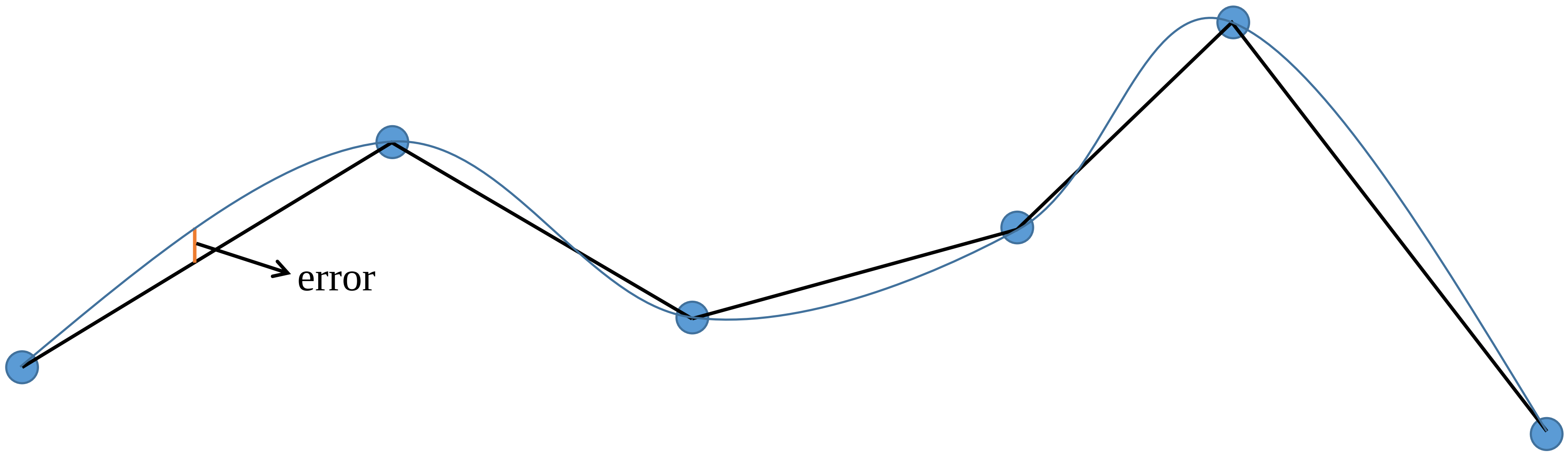
finegrid[0][1] = (Rawdata[0][0] + Rawdata[0][1])/(*insert*+1)

So *insert* will +1 in *linear_intp* function.

And the length of finegrid (*m*–1)*(*insert* +1)+1

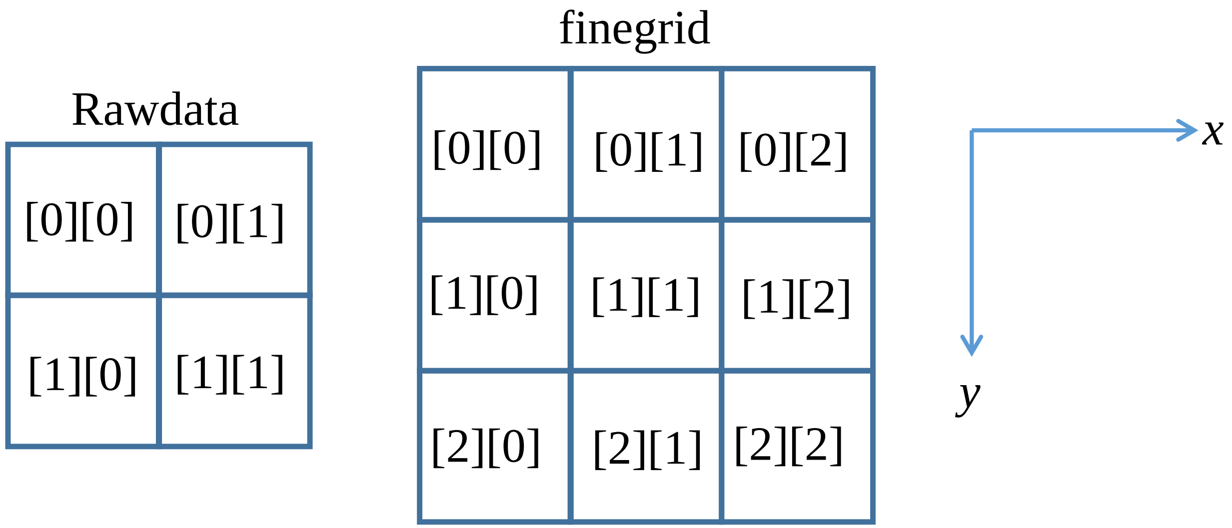
Issue of Linear Interpolation

- Since it uses linear, so if the original data is a curve, it causes the error.



2D linear Interpolation

- If we use linear interpolation, and *x*, *y* are independent, then we can interpolate *x* axis first, then interpolate *y*, or vice versa. For example, we want to refine it to 3 by 3 grid, or it can be seem as insert one grid.



Lagrange interpolation

- Suppose there are $n+1$ data points, then there exists a polynomial with degree of n uniquely, and pass through those points.
- The polynomial can be determined by the general form

$$f(x) = a_n x^n + \ldots + a_0 = \sum_{i=0}^{n} a_i x^i \qquad \deg(f) = n$$

- So that by linear system

$$\begin{bmatrix} 1 & x_0 & \cdots & x_0{}^n \\ 1 & x_1 & \cdots & x_1{}^n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \cdots & x_n{}^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}$$

- The matrix is called “Vandermonde matrix”

Lagrange interpolation

- But Vandermonde matrix is a ill-conditioned matrix, it causes the numerical instability. So we expand the matrix and calculate its determinant.
- For example, $n = 2$, $P_2(x) = a_0x^2 + a_0x^1 + a_0$, and
- $P_2(x_0) = f(x_0)$, $P_2(x_1) = f(x_1)$, $P_2(x_2) = f(x_2)$,

$$\left\|\begin{bmatrix} P_2 & 1 & x & x^2 \\ f(x_0) & 1 & x_0 & x_0^2 \\ f(x_1) & 1 & x_1 & x_1^2 \\ f(x_2) & 1 & x_2 & x_2^2 \end{bmatrix}\right\| = 0 = P_2 \begin{vmatrix} 1 & x_0 & x_0^2 \\ 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \end{vmatrix} - f(x_0) \begin{vmatrix} 1 & x & x^2 \\ 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \end{vmatrix} + f(x_1) \begin{vmatrix} 1 & x & x^2 \\ 1 & x_0 & x_0^2 \\ 1 & x_2 & x_2^2 \end{vmatrix} - f(x_2) \begin{vmatrix} 1 & x & x^2 \\ 1 & x_0 & x_0^2 \\ 1 & x_1 & x_1^2 \end{vmatrix}$$

Lagrange interpolation

$$\begin{vmatrix} 1 & x_0 & x_0^{\,2} \\ 1 & x_1 & x_1^{\,2} \\ 1 & x_2 & x_2^{\,2} \end{vmatrix} = \begin{vmatrix} 1 & x_0 & x_0^{\,2} \\ 0 & x_1 - x_0 & x_1^{\,2} - x_0^{\,2} \\ 0 & x_2 - x_0 & x_2^{\,2} - x_0^{\,2} \end{vmatrix} = \begin{vmatrix} x_1 - x_0 & x_1^{\,2} - x_0^{\,2} \\ x_2 - x_0 & x_2^{\,2} - x_0^{\,2} \end{vmatrix}$$

$$= \left(x_1 - x_0\right)\left(x_2^{\,2} - x_0^{\,2}\right) - \left(x_2 - x_0\right)\left(x_1^{\,2} - x_0^{\,2}\right)$$

$$= \left(x_1 - x_0\right)\left(x_2 - x_0\right)\left[\left(x_2 + x_0\right) - \left(x_1 + x_0\right)\right]$$

$$= \left(x_1 - x_0\right)\left(x_2 - x_0\right)\left(x_2 - x_1\right)$$

$$P_2(x) = f(x_0)\frac{\left(x_1 - x\right)\left(x_2 - x\right)}{\left(x_1 - x_0\right)\left(x_2 - x_0\right)} + f(x_1)\frac{\left(x_0 - x\right)\left(x_2 - x\right)}{\left(x_1 - x_0\right)\left(x_1 - x_2\right)}\ f(x_2)\frac{\left(x_1 - x\right)\left(x_0 - x\right)}{\left(x_2 - x_0\right)\left(x_2 - x_1\right)}$$

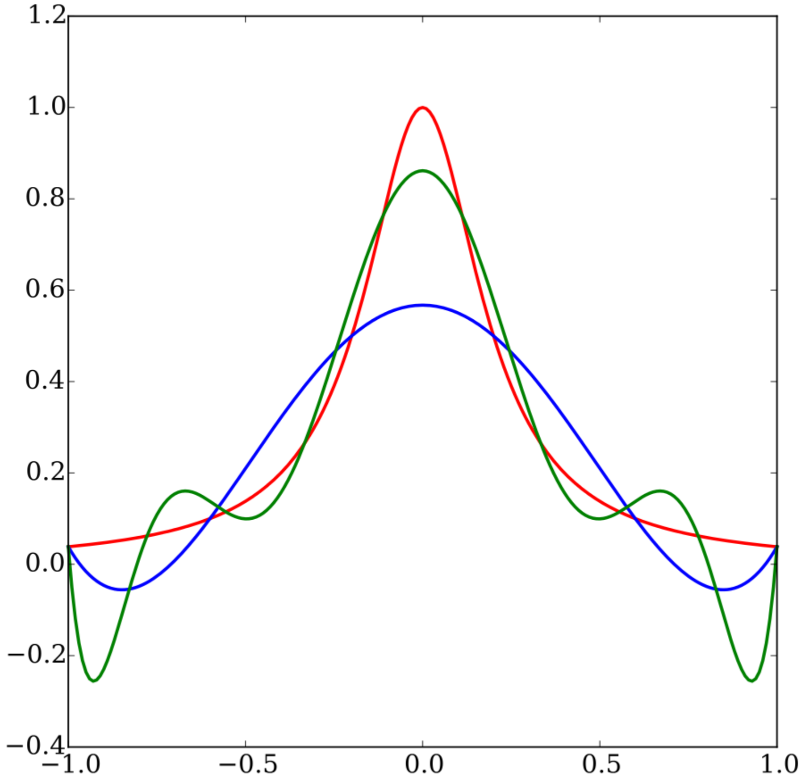
Lagrange interpolation

- It can be written as $$P_2(x) = \sum_{j=0}^{2} f(x_j) \prod_{k=0,k\neq j}^{2} \frac{x - x_k}{x_j - x_k}$$
- Give n+1 distinct points (x_0, y_0), (x_2, y_2), … (x_n, y_n), there is a unique polynomial P with real coefficients satisfying $P(x_i) = y_i$, $i \in Z^+$ such that deg(P) = n. If not distinct points, then deg(P) < n
- Lagrange polynomial:

$$P_n(x) = \sum_{j=0}^{n} f(x_j) \prod_{k=0,k\neq j}^{n} \frac{x - x_k}{x_j - x_k}$$

Runge's phenomenon

Suppose the source is $f(x) = (1+25x^2)^{-1}$
Use polynomial to fit the data points, as the degree is higher, the oscillation at the edges is more obvious.



https://en.wikipedia.org/wiki/Runge%27s_phenomenon

Spline

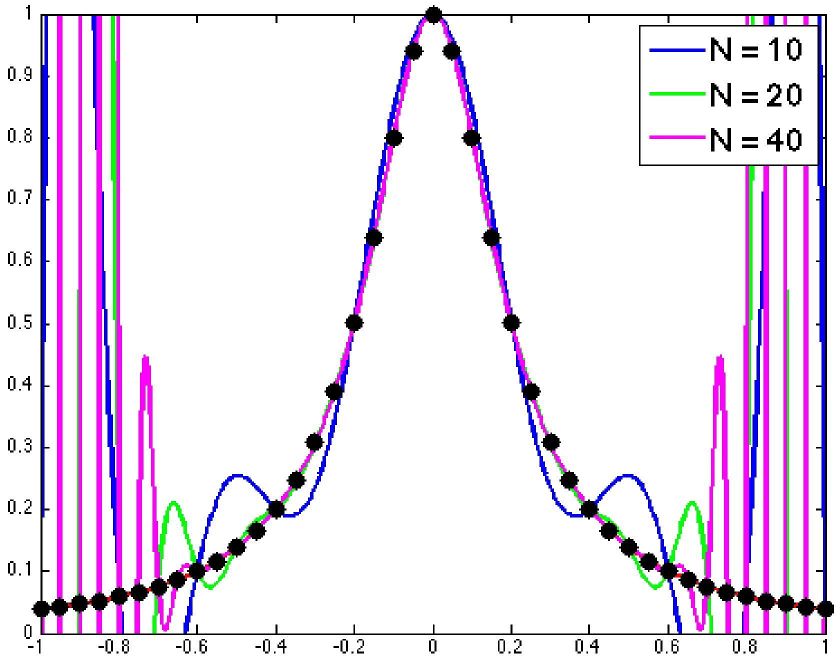
- Spline is a curve segment that connect two distinct points, in word and power point, there is a “Bezier curve”, it’s a kind of spline.
- There are many advantages of spline. Suppose there are n points, it can choose a polynomial f such that $\deg(f) \leq n-1$ to avoid runge's phenomenon, and less error than linear Interpolation.
- For $\deg(f) = 0$, it’s linear spline, it’s the same linear interpolation

Cubic Spline

- The linear spline is like linear interpolation, it’s not smooth at those points. That means the derivative doesn’t exist there.
- So that add more conditions that makes the curve smooth. Suppose there are n+1 points $P_0(x_0, y_0)$, $P_1(x_1, y_1)$, $P_2(x_2, y_2)$, … $P_n(x_n, y_n)$, and curve S_0 connects P_0 and P_1, S_1 connects P_1 and P_2, … and so on.
- By continuous condition:
- $S(x_0^+) = S(x_0^-)$, $S'(x_0^+) = S'(x_0^-)$, $S''(x_0^+) = S''(x_0^-)$
- But the derivative doesn’t exist at the boundary, so that there are one more condition.

Why Spline ?

- Find a curve passes through all points and which is smooth at those points.
- Lagrange interpolate is with runge’s phenomena
- Spline is the best choice
- Choose **cubic** spline



Derivation of cubic spline

- Spline is a curve connect two points, suppose there are n+1 points, then there are n segment spline.
- $(x_0, y_0), (x_1, y_1), (x_2, y_2), \ldots (x_n, y_n)$,

$$S(x)\begin{cases} S_0(x), & x_0 \le x < x_1 \\ S_1(x), & x_1 \le x < x_2 \\ \\ \\ S_n(x), & x_{n-1} \le x < x_n \end{cases}$$

Derivation of spline

- Suppose its forms are

$$S_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i$$

$$S_i'(x) = 3a_i(x - x_i)^2 + 2b_i(x - x_i) + c_i$$

$$S_i''(x) = 6a_i(x - x_i) + 2b_i$$

Derivation of spline

- For segments, there are $4n$ unknown parameters
- All $S_i\,(x_i) = y_i$ and $S_i\,(x_{i+1}) = y_{i+1}$ → $2n$ functions
- $S_i'(x_i) = S_{i+1}'(x_{i+1})$ and $S_i''(x_i) = S_{i+1}''(x_{i+1})$

  → $2n–2$ functions
- Boundary conditions:

  $S_0''(x_0) = b_0 = M$ $\quad$ $S_{n-1}''(x_n) = b_{n\text{-}1} = N$
- So that we can determine $S(x)$ uniquely

Derivation of spline

- All of $S(x)$, $S'(x)$, $S''(x)$ are continuous at x_i, so

$$x_{i+1} - x_i = h_i$$

$$S_{i+1}(x_{i+1}) = S_i(x_{i+1}) \qquad d_{i+1} = a_i h_i^3 + b_i h_i^2 + c_i h_i + d_i \text{ --(1.)}$$

$$S_i'(x_{i+1}) = S_{i+1}'(x_{i+1}) \qquad c_{i+1} = 3a_i h_i^2 + 2b_i h_i + c_i \text{ --(2.)}$$

$$S_i''(x_{i+1}) = S_{i+1}''(x_{i+1}) \qquad b_{i+1} = 3a_i h_i + b_i \text{ --(3.)}$$

Derivation of spline

$$d_i = y_i \quad d_{i+1} = y_{i+1} \qquad 3h_i a_i = b_{i+1} - b_i$$

$$y_{i+1} - y_i = \frac{b_{i+1} - b_i}{3} h_i^{\,2} + b_i h_i^{\,2} + c_i h_i = \frac{b_{i+1} + 2b_i}{3} h_i^{\,2} + c_i h_i$$

$$c_i = \frac{y_{i+1} - y_i}{h_i} - \frac{b_{i+1} + 2b_i}{3} h_i$$

Derivation of spline

$$c_{i+1} = 3a_i h_i + 2b_i h_i + c_i$$

$$= (b_{i+1} - b_i)h_i + 2b_i h_i + \frac{y_{i+1} - y_i}{h_i} - \frac{b_{i+1} + 2b_i}{3} h_i$$

$$= \frac{y_{i+1} - y_i}{h_i} + \frac{2b_{i+1} + b_i}{3} h_i$$

$$= \frac{y_{i+1} - y_i}{h_i} + \frac{2b_{i+1} + b_i}{3} h_i$$

Derivation of spline

$$c_{i+1} = \frac{y_{i+2} - y_{i+1}}{h_{i+1}} - \frac{b_{i+2} + 2b_{i+1}}{3} h_{i+1} = \frac{y_{i+1} - y_i}{h_i} + \frac{2b_{i+1} + b_i}{3} h_i$$

$$b_{i+2}h_{i+1} + 2b_{i+1}(h_{i+1} + h_i) + b_i h_i = 3\left[\frac{y_{i+2} - y_{i+1}}{x_{i+2} - x_{i+1}} - \frac{y_{i+1} - y_i}{x_{i+1} - x_i}\right]$$

- In matrix form, ***Hb* = *y***
- ***H*** is (n-1) by (n-1) matrix, ***b*** and ***y*** are n-1 vector

Derivation of spline

$$\boldsymbol{H} = \begin{bmatrix} 2(h_1+h_2) & h_2 & 0 & 0 & \cdots & 0 \\ h_2 & 2(h_2+h_3) & h_3 & 0 & \cdots & 0 \\ 0 & h_3 & 2(h_3+h_4) & h_4 & \cdots & 0 \\ 0 & 0 & h_4 & 2(h_4+h_5) & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & h_{n-2} \\ 0 & 0 & 0 & 0 & h_{n-2} & 2(h_{n-2}+h_{n-1}) \end{bmatrix}$$

$$\boldsymbol{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \end{bmatrix} \qquad \boldsymbol{y} = 3\begin{bmatrix} \frac{y_2-y_1}{x_2-x_1} - \frac{y_1-y_0}{x_1-x_0} \\ \frac{y_3-y_2}{x_3-x_2} - \frac{y_2-y_1}{x_2-x_1} \\ \vdots \\ \frac{y_{i+1}-y_i}{x_{i+1}-x_i} - \frac{y_i-y_{i-1}}{x_i-x_{i-1}} \end{bmatrix}$$

Solving the eqs.

- Here we choose $b_0 = b_n = 0$, so both of them are not in the vector. It's called "Nature cubic spline"
- After solving all b_i, then substitute them to solve the other parameters.
- The matrix can be solved by gaussian elimination

$$d_i = y_i$$

$$c_i = \frac{y_{i+1} - y_i}{h_i} + \frac{2b_{i+1} + b_i}{3} h_i$$

$$a_i = \frac{b_{i+1} - b_i}{3h_i}$$