CS381: Numerical Computation & Softwares

Interpolation

Jakramate Bootkrajang

Department of Computer Science

Chiang Mai University

Quote of the day

"Nothing in life is to be feared, it is only to be understood. Now is the time to understand more, so that we may fear less. "

- *Marie Curie* -

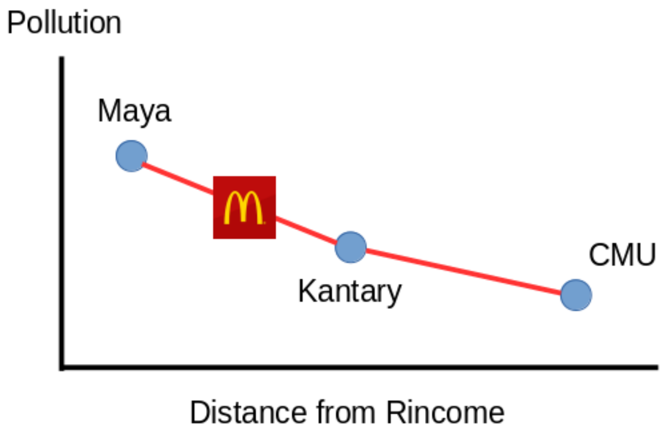
เป้าหมาย

- นิยามของปัญหา
- Polynomial interpolation
- Spline interpolation

Motivation

- สมมุติว่าเราต้องการติดตั้งเครื่องดักจำฝุ่นละอองในอากาศ เพื่อตรวจวัดคุณภาพอากาศ ตลอดแนวถนนนิมมาน
- เครื่องมีราคาสูงมาก เราซื้อได้แค่ 3 เครื่อง โดยติดตั้งไว้ที่ ห้างเมญ่า โรงแรมแคนทารี่ และหอประชุมมช.
- คุณภาพอากาศ ณ 3 ตำแหน่งดังกล่าวสามารถอ่านได้จากเครื่อง แต่ถามว่า คุณภาพอากาศตรง McDonald เป็นอย่างไรเราตอบไม่ได้
- วิธีหนึ่งเพื่อใช้ประมาณค่าคุณภาพอากาศที่ McDonald คือการลากเส้นเชื่อมจุด 3 จุดเข้าด้วยกัน และใช้ค่าบนเส้นเชื่อมดังกล่าวเป็นตัวแทน

Motivation



The Interpolation Problem

สมมุติว่าเรามีข้อมูล $n + 1$ ตัว ที่อยู่ในรูปของ $(x_0, y_0), \ldots (x_n, y_n)$. เราสนใจที่จะหา พหุนาม (polynomial) $Q_n(x)$ ที่มีดีกรีต่ำที่สุด ที่ทำให้

$$Q_n(x_i) = y_i$$

สำหรับ $i \in \{0, \ldots, n\}$

เราเรียกพนุนามที่มีคุณสมบัติข้างต้นว่า Interpolant

Polynomial Interpolation

Can we do that ?

มีทฤษฎีบทบทหนึ่งที่บอกว่า เราสามารถหา $Q_n(x)$ ดังกล่าวได้

Theorem

If $x_0, \ldots x_n$ *are distinct, then for any* $f(x_0), \ldots, f(x_n)$ *there exists a unique polynomial* $Q_n(x)$ *with sequence of coefficients* $a_0, \ldots, a_n$ *such that* $Q_n(x_i) = y_i$ *for all* $i \in \{0, \ldots, n\}$*, where*

$$Q_n(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

Lagrange polynomials

Lagrange เสนอแนวคิดในการสร้างพหุนามดังกล่าวในรูปของ

$$Q_n(x) = \sum_{j=0}^{n} f(x_j)L_j(x) = f(x_0)L_0(x) + f(x_1)L_1(x) + \ldots f(x_n)L_n(x)$$

เพื่อให้เงื่อนไขของ Interpolant เป็นจริง นั่นคือ

$$Q_n(x_i) = f(x_i) \quad , 0 \leq i \leq n$$

สิ่งที่เราต้องทำก็คือหา $L_j(x)$ ที่

$$L_j(x_i) = \begin{cases} 1 & \text{if} \quad i = j \\ 0 & \text{otherwise} \end{cases}$$

A good choice of $L_j(x)$

เราสามารถสร้างพนุนามที่เราต้องการได้ง่ายๆโดยกำหนดให้

$$
\begin{aligned}
L_j(x) &= \frac{(x-x_0)(x-x_1)\dots(x-x_{j-1})(x-x_{j+1})\dots(x-x_n)}{(x_j-x_0)(x_j-x_1)\dots(x_j-x_{j-1})(x_j-x_{j+1})\dots(x_j-x_n)} \\
&= \frac{\prod_{i=0,i\neq j}^{n}(x-x_i)}{\prod_{i=0,i\neq j}^{n}(x_j-x_i)}
\end{aligned} \tag{1}
$$

for $j = 0, \dots, n$

Example: Linear interpolation

สมมุติว่าเรามีข้อมูลสองตัว (x_0, y_0) and (x_1, y_1) ปัญหาของการ interpolate ข้อมูลสองตัวคือการหาเส้นตรงที่เชื่อมจุดทั้งสอง (พหุนามดีกรี 1)

เราพบว่าเราสามารถกำหนด Lagrange polynomial ให้เป็น $L_0(x) = \frac{x-x_1}{x_0-x_1}$ and $L_1(x) = \frac{x-x_0}{x_1-x_0}$ ดังนั้น พนุนามที่เราตามหาซึ่ง interpolate ข้อมูลทั้งสองได้คือ

$$
\begin{aligned}
Q(x) &= f(x_0)L_0(x) + f(x_1)L_1(x) = f(x_0)\frac{x - x_1}{x_0 - x_1} + f(x_1)\frac{x - x_0}{x_1 - x_0} \\
&= \frac{f(x_0) - f(x_1)}{x_0 - x_1}x + \frac{f(x_1)x_0 - f(x_0)x_1}{x_0 - x_1} = a_0 + a_1x
\end{aligned}
$$

Quiz: ลองตรวจสอบดูว่าพนุนามดังกล่าว interpolate ข้อมูลทั้งสองจริงหรือไม่

ข้อจำกัดของ Lagrange Polynomial

ในทางปฏิบัติแล้ว เรามักไม่ค่อยใช้ Lagrange polynomial ในการ interpolate ข้อมูล ด้วยเหตุผลสองประการคือ

- พนุนามในรูปของ $Q(x) = \sum_{i=0}^{n} y_i \prod_{j=0, j\neq i}^{n} \frac{x - x_j}{x_i - x_j}$ มีลักษณะ ill-conditioned: cancellation errors สามารถเกิดขึ้นได้ !
- ในกรณีที่เรามีข้อมูลใหม่ (x_{n+1}, y_{n+1}) เราจำเป็นต้องหา Lagrange polynomial ใหม่หมดตั้งแต่ต้น

Lab: Implement Lagrange polynomial

Newton polynomial

- เราสามารถเริ่มจากการลองหา polynomial อย่างง่ายที่ interpolate ข้อมูล 1 ตัว
  - $Q_0(x) = a_0$ แล้วให้ค่าสัมประสิทธิ์ $a_0 = f(x_0)$
- ในกรณีที่ข้อมูลมีสองตัว เราอาจลองเขียนพหุนามโดย
  - $Q_1(x) = a_0 + a_1 x$
- ถูกต้องไหม ?

Newton polynomial

- เหมือนจะยังไม่ถูกต้องเสียทีเดียว เนื่องจากไม่เป็นไปตามเงื่อนไขของการทำ interpolation
- เราสามารถปรับปรุงนิดหน่อยเพื่อให้ได้ตามเงื่อนไข
  - $Q_1(x) = a_0 + a_1(x - x_0)$
- ถูกต้องไหม ?

Newton polynomial

- สมมุติว่าเราสามารถหา $Q_{n-1}(x)$ ที่เป็นไปตามเงื่อนไขของการทำ interpolation ได้ เราจะหา $Q_n(x)$ ได้อย่างไร
- หากเราสร้าง $Q_n(x) = Q_{n-1}(x) + c(x - x_0)\cdots(x - x_{n-1})$
- เราต้องแน่ใจว่า $Q_n(x)$ ใหม่ที่ได้ต้อง interpolate $f(x_n)$ ด้วย
- ด้วยข้อกำหนดเพิ่มเติมข้างต้น เราสามารถหาค่า c ได้ดังนี้
- $c = \frac{f(x_n) - Q_{n-1}(x_n)}{\prod_{j=0}^{n-1}(x_n - x_j)}$

กล่าวโดยสรุปแล้ว Newton polynomial สามารถสร้างได้โดย

$$
\begin{aligned}
Q_n(x) &= a_0 + a_1(x - x_0) + \ldots + a_n(x - x_0) \cdot \ldots \cdot (x - x_n) \\
&= a_0 + \sum_{j=1}^{n} a_j \prod_{k=0}^{j-1} (x - x_k)
\end{aligned}
$$

โดยที่

$$
\begin{aligned}
a_0 &= f(x_0) \\
a_j &= \frac{f(x_j) - Q_{j-1}(x_j)}{\prod_{k=0}^{j-1}(x_j - x_k)}, \quad 1 \leq j \leq n
\end{aligned}
$$

ตัวอย่าง

ให้หา polynomial ที่ interpolate ข้อมูลสองตัว $(x_0, f(x_0)), (x_1, f(x_1))$

$$
\begin{aligned}
Q_1(x) &= a_0 + a_1(x - x_0) \\
&= f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0)
\end{aligned}
$$

Lab: Implement Newton polynomial

Divided differences

- การเขียนโปรแกรมเพื่อสร้าง Newton polynomial อาจทำได้ง่ายขึ้นโดยอาศัยการคำนวณ a_j ที่ใช้หลักการที่เรียกว่า divided difference
- เราจะเรียก $a_j = \frac{f(x_j) - Q_{j-1}(x_j)}{\prod_{k=0}^{j-1}(x_j - x_k)}$ ว่า j^{th}-order divided difference
- สังเกตว่าการคำนวณค่า a_j ขึ้นอยู่กับจุด $x_0, \ldots x_j$ และค่าของฟังก์ชัน ณ จุดนั้นๆ $f(x_0) \ldots f(x_j)$

Divided differences

- เราจะเขียน $a_j = f[x_0, \ldots, x_j]$ เพื่อเน้นว่าค่าของ a_j ขึ้นอยู่กับจุด $x_0, \ldots x_j$ และค่าของฟังก์ชัน ณ จุดนั้นๆ
- สำหรับ $a_0 = f[x_0] = f(x_0)$
- เมื่อใช้ notation ใหม่ของ divided difference เราสามารถเขียน Newton polynomial ได้ใหม่ว่า

$$Q_n(x) = a_0 + f[x0, x1](x - x_0) + \ldots + f[x_0, \ldots, f_n](x - x_0) \cdot \ldots \cdot (x - x_n)$$

Divided differences

- Divided difference น่าสนใจตรงที่ว่ามีทฤษฎีที่ช่วยให้เราคำนวณ divide difference ของพจน์สูงๆ โดยใช้หลักการ recursive (การเรียกซ้ำ)

$$f[x_0, \ldots, f_n] = \frac{f[x_1, \ldots, f_n] - f[x_0, \ldots, f_{n-1}]}{x_n - x_0}$$

Lab: Implement divided difference algorithm

Exercise

- Find the polynomial of degree ≤ 2 that interpolates (-1,9), (0,5), (1,3)

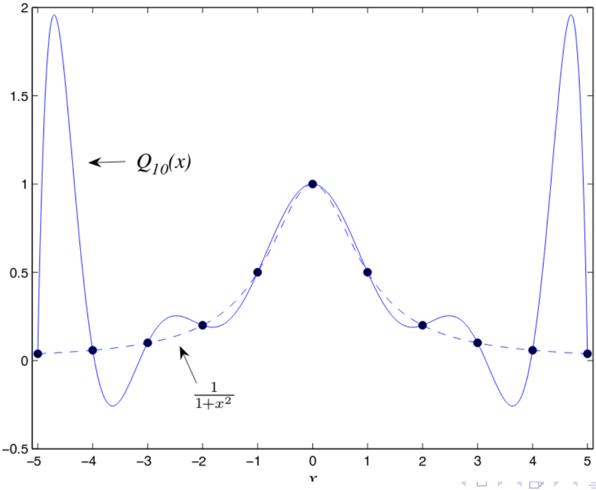
Spline Interpolation

Spline interpolation

- ข้างต้นเราพิจารณาแค่ polynomial interpolation
- คราวนี้เราจะลองพิจารณาการทำ interpolation ที่ประกอบด้วย polynomial หลายๆตัวต่อกัน
- เราเรียกว่า piece-wise polynomial หรือ spline

Why do we need spline interpolation ?

เมื่อเรามีข้อมูลหลายตัว polynomial interpolation มักจะให้พหุนามที่ซับซ้อนเกินไป

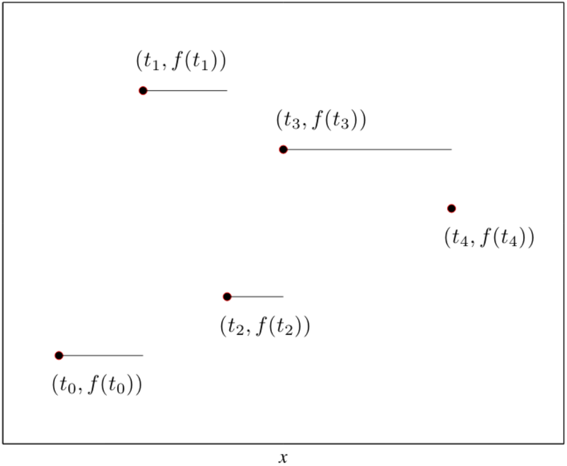


นิยามของ Spline interpolation

- Spline interpolation ประกอบด้วยจุด t_i จำนวน $n+1$ ตัว โดยที่ $t_0 < t_1 < \cdots < t_n$ และเรียกจุดเหล่านั้นว่า knots (แปลว่าปม) และ $t_0 \leq x_0, t_n \geq x_n$
- Spline ที่มีดีกรีเท่ากับ k บน ช่วง $t_0, \ldots, t_n$ คือฟังก์ชัน $s(x)$ ที่มีคุณสมบัติดังต่อไปนี้
  1) ดีกรีของพนุหนาม $s(x)$ บนทุกๆช่วงเปิด $[t_{i-1}, t_i)$ มีค่าน้อยกว่าหรือเท่ากับ k
  2) เราสามารถหาอนุพันธ์ลำดับที่ $k-1$ ของ $s(x)$ บน $[t_0, t_n]$ ได้

ตัวอย่างของ Spline interpolation

ตัวอย่างของ Spline degree 0 หรือที่เรียกว่า piecewise-constant function.



Spline of degree 1

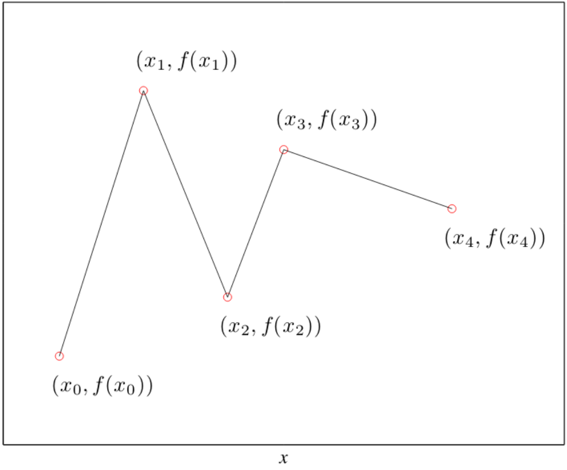
Spline degree 1 สามารถนิยามได้ว่า

$$s(x) = \begin{cases} s_0(x) = a_0 x + b_0 & , \quad x \in [t_o, t_1) \\ s_1(x) = a_1 x + b_1 & , \quad x \in [t_1, t_2) \\ \vdots & \\ s_n(x) = a_n x + b_n & , \quad x \in [t_{n-1}, t_n] \end{cases}$$

ในความเป็นจริง t_i (knot) อาจไม่จำเป็นจะต้องเป็นจุดเดียวกับ x_i (ข้อมูล) เสมอไป อาจเลือกจุดที่ต่างออกไป หรือเลือกให้ knot กับ ข้อมูลเป็นจุดเดียวกันก็ได้

ตัวอย่างของ Linear Spline

ตัวอย่างของ Spline degree 1 หรือที่เรียกว่า piecewise-linear function.



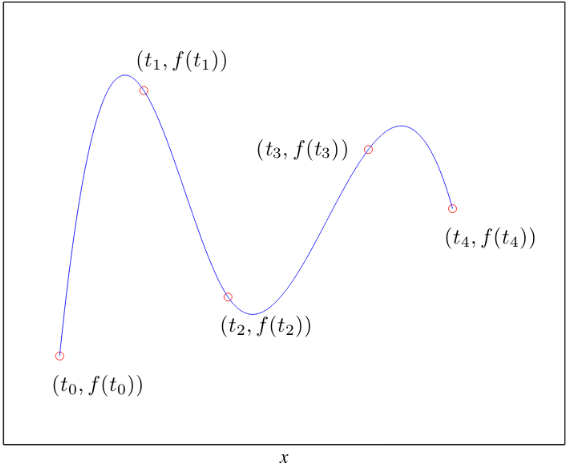
Cubin splines

Spline ที่น่าจะถูกนำมาใช้มากที่สุดอาจเป็น cubic spline ที่เกิดจากการนำพนุนามดีกรี ≤ 3 มาเชื่อมต่อกัน (cubic = กำลังสาม)

$$s(x) = \begin{cases} s_0(x) & , \quad x \in [t_o, t_1) \\ s_1(x) & , \quad x \in [t_1, t_2) \\ & \vdots \\ s_n(x) & , \quad x \in [t_{n-1}, t_n] \end{cases}$$

โดยที่ $\forall i$, ดีกรีของ $s_i(x) \leq 3$ และ ตามข้อกำหนด cubic splines จะต้องเป็นพนุนามที่สามารถหาอนุพันธ์ได้ถึงลำดับที่สอง $(3 - 1 = 2)$

ตัวอย่างของ Cubic Spline



การหา cubin spline

- เพื่อให้ง่ายต่อการทำความเข้าใจ เราจะกำหนดให้ข้อมูลที่เรามี ทำหน้าที่เป็น knots ไปเลยนั่นคือ

$$s(t_i) = y_i, \quad 0 \leq i \leq n$$

- เราสามารถอนุมานจากข้อกำหนดในการทำ interpolation ได้อีกว่า

$$s_{i-1}(t_i) = y_i = s_i(t_i), \quad 1 \leq i \leq n-1$$

- นอกจากนั้นเราก็ต้องการความต่อเนื่อง (continuity) ของอนุพันธ์อันดับ 1 และ 2 ที่จุด knot ต่างๆ

$$s_i'(t_{i+1}) = s_{i+1}'(t_{i+1}), \quad 0 \leq i \leq n-2$$

$$s_i''(t_{i+1}) = s_{i+1}''(t_{i+1}), \quad 0 \leq i \leq n-2$$

Computing the spline

- กำหนดให้ $h_i = t_{i+1} - t_i$ และให้ $z_i = s''(t_i)$
- เนื่องจาก อนุพันธ์ลำดับที่สองของ $s(x)$ ก็คือฟังก์ชันเชิงเส้น เราจะได้ว่า $s_i''(x)$ ก็คือเส้นตรงที่เชื่อม (t_i, z_i) และ (t_{i+1}, z_{i+1})
- หากใช้ Lagrange polynomial เพื่อหาเส้นตรงที่ interpolate สองจุดเราจะได้ว่า

$$s_i''(x) = \frac{x - t_i}{h_i} z_{i+1} - \frac{x - t_{i+1}}{h_i} z_i$$

Computing the spline

จาก $s_i''(x) = \frac{x-t_i}{h_i} z_{i+1} - \frac{x-t_{i+1}}{h_i} z_i$

- อินทิเกรตเส้นตรงหนึ่งครั้งเราจะได้

$$s_i'(x) = \frac{1}{2}(x - t_i)^2 \frac{z_{i+1}}{h_i} - \frac{1}{2}(x - t_{i+1})^2 \frac{z_i}{h_i} + \tilde{c}$$

- อินทิเกรตอีกหนึ่งครั้งเราจะได้

$$s_i(x) = \frac{z_{i+1}}{6h_i}(x - t_i)^3 + \frac{z_i}{6h_i}(t_{i+1} - x)^3 + C(x - t_i) + D(t_{i+1} - x)$$

- How to find C and D ?

Computing the spline

- เงื่อนไขของการทำ interpolation ที่ว่า $s(t_i) = y_i$ ทำให้เราได้ว่า

$$y_i = \frac{z_i}{6h_i} h_i^3 + Dh_i \text{ นั่นคือ } D = \frac{y_i}{h_i} - \frac{z_i h_i}{6}$$

- เช่นเดียวกัน เงื่อนไข $s_i(t_{i+1}) = y_{i+1}$ ทำให้เราได้ว่า

$$y_{i+1} = \frac{z_{i+1}}{6h_i} h_i^3 + Ch_i \text{ นั่นคือ } C = \frac{y_{i+1}}{h_i} - \frac{z_{i+1} h_i}{6}$$

Computing the spline

นั่นหมายความว่าเราสามารถเขียน $s(x)$ ในรูปของ

$$
\begin{aligned}
s_i(x) = &\frac{z_{i+1}}{6h_i}(x - t_i)^3 + \frac{z_i}{6h_i}(t_{i+1} - x)^3 \\
&+ \left(\frac{y_{i+1}}{h_i} - \frac{z_{i+1}h_i}{6}\right)(x - t_i) \\
&+ \left(\frac{y_i}{h_i} - \frac{z_i h_i}{6}\right)(t_{i+1} - x)
\end{aligned}
$$

Computing the spline

$s(x)$ ที่ได้ ตรงเงื่อนไขของ อนุพันธ์ลำดับที่สอง ต่อไปเราจะเพิ่มเงื่อนไขที่กำหนดลักษณะของ $s(x)$ ผ่านอนุพันธ์ลำดับที่ 1 ซึ่งกำหนดไว้ว่า $s_i'(x) = s_{i-1}'(x)$

$$
\begin{aligned}
s_i'(x) = & \frac{z_{i+1}}{2h_i}(x - t_i)^2 - \frac{z_i}{2h_i}(t_{i+1} - x)^2 \\
& + \frac{y_{i+1}}{h_i} - \frac{z_{i+1}h_i}{6} \\
& - \frac{y_i}{h_i} + \frac{z_i h_i}{6}
\end{aligned}
$$

Computing the spline

$$
\begin{aligned}
s'_{i-1}(x) = &\frac{z_i}{2h_{i-1}}(x - t_{i-1})^2 - \frac{z_{i-1}}{2h_{i-1}}(t_i - x)^2 \\
&+ \frac{y_i}{h_{i-1}} - \frac{z_i h_{i-1}}{6} \\
&- \frac{y_{i-1}}{h_{i-1}} + \frac{z_{i-1} h_{i-1}}{6}
\end{aligned}
$$

ถึงจุดนี้เราจะได้ว่า

$$
\begin{aligned}
s_i'(t_i) &= -\frac{z_i}{2h_i}h_i^2 + \frac{y_{i+1}}{h_i} - \frac{z_{i+1}h_i}{6} - \frac{y_i}{h_i} + \frac{z_ih_i}{6} \\
&= -\frac{h_i}{3}z_i - \frac{h_i}{6}z_{i+1} - \frac{y_i}{h_i} + \frac{y_{i+1}}{h_i} \qquad (1) \\
s_{i-1}'(t_i) &= \frac{z_i}{2h_{i-1}}h_{i-1}^2 + \frac{y_i}{h_{i-1}} - \frac{z_ih_{i-1}}{6} - \frac{y_{i-1}}{h_{i-1}} + \frac{z_{i-1}h_{i-1}}{6} \\
&= \frac{h_{i-1}}{6}z_{i-1} + \frac{h_{i-1}}{3}z_i - \frac{y_{i-1}}{h_{i-1}} + \frac{y_i}{h_{i-1}} \qquad (2)
\end{aligned}
$$

สืบเนื่องจากเงื่อนไขว่า สมการ (1) ต้องเท่ากับ สมการ (2) (เงื่อนไขอนุพันธ์อันดับหนึ่งของ spline)

$$s'_i(t_i) = s'_{i-1}(t_i)$$

$$-\frac{h_i}{3}z_i - \frac{h_i}{6}z_{i+1} - \frac{y_i}{h_i} + \frac{y_{i+1}}{h_i} = \frac{h_{i-1}}{6}z_{i-1} + \frac{h_{i-1}}{3}z_i - \frac{y_{i-1}}{h_{i-1}} + \frac{y_i}{h_{i-1}}$$

หรือเมื่อจัดรูปสมการแบบด้านล่าง เราจะได้ระบบสมการ $n-1$ สมการ

$$-\frac{h_i}{6}z_{i-1} - \frac{h_i + h_{i-1}}{3}z_i + \frac{h_i}{6}z_{i+1} = \frac{1}{h_i}(y_{i+1} - y_i) - \frac{1}{h_{i-1}}(y_i - y_{i-1})$$

โดยที่ $1 \leq i \leq n$ และมี ตัวแปรที่ไม่ทราบค่า (unknown) $n+1$ ตัว ได้แก่ $z_0, \ldots, z_n$

System of equations for the spline

ระบบสมการที่เกิดขึ้น มีตัวแปรไม่ทราบค่ามากกว่าจำนวนสมการ ทำให้เราต้องกำจัดตัวแปรไม่ทราบค่า บางตัวไป โดยทั่วไปแล้ว เรามักให้ $z_0 = z_n = 0$ แล้วเขียนระบบสมการในรูปของเมทริกซ์

$$\begin{bmatrix} \frac{h_0+h_1}{3} & \frac{h_1}{6} & & & \\ \frac{h_1}{6} & \frac{h_1+h_2}{3} & \frac{h_2}{6} & & \\ & \ddots & \ddots & \ddots & \\ & & \frac{h_{n-3}}{6} & \frac{h_{n-3}+h_{n-2}}{3} & \frac{h_{n-2}}{6} \\ & & & \frac{h_{n-2}}{6} & \frac{h_{n-2}+h_{n-1}}{3} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_{n-2} \\ z_{n-1} \end{bmatrix} = \begin{bmatrix} \frac{y_2-y_1}{h_1} - \frac{y_1-y_0}{h_0} \\ \frac{y_3-y_2}{h_2} - \frac{y_2-y_1}{h_1} \\ \vdots \\ \frac{y_{n-1}-y_{n-2}}{h_{n-2}} - \frac{y_{n-2}-y_{n-3}}{h_{n-3}} \\ \frac{y_n-y_{n-1}}{h_{n-1}} - \frac{y_{n-1}-y_{n-2}}{h_{n-2}} \end{bmatrix}$$

$$H * Z = Y$$

System of equations for the spline

- เนื่องจาก matrix H เป็น positive definite matrix ($a^T Ha \geq 0$)
- เราสามารถหา inverse ของเมทริกซ์ดังกล่าวได้
- คำตอบของระบบสมการดังกล่าวก็สามารถหาได้โดย $Z = H^{-1}Y$
- เมื่อเราทราบ Z เราสามารถกลับไปหา $s_i(x)$ ได้

แบบฝึกหัด

- Implement Spline Interpolation.
- Derive the system of equations for the case where $h_i = h, \forall i$.

References

- Introduction to numerical analysis by Doron Levy (บทที่ 3)