ULSAN NATIONAL INSTITUTE OF SCIENCE AND TECHNOLOGY

Third Week

Interpolation

ULSAN NATIONAL INSTITUTE OF SCIENCE AND TECHNOLOGY

Interpolation

3.1. Taylor expansion

Theorem 3.1

Suppose $f \in C^n[a,b]$, that $f^{(n+1)}$ exists on $[a,b]$, and $x_0 \in [a,b]$, there exists a number $\xi(x)$ between x_0 and x with

$$f(x) = P_n(x) + R_n(x),$$

where

$$P_n(x) = f(x_0)+f'(x_0)(x-x_0)+\frac{f''(x_0)}{2!}(x-x_0)^2+\cdots+\frac{f^{(n)}(x_0)}{n!}(x-x_0)^n = \sum_{k=0}^{n}\frac{f^{(k)}(x_0)}{k!}(x-x_0)^k$$

and

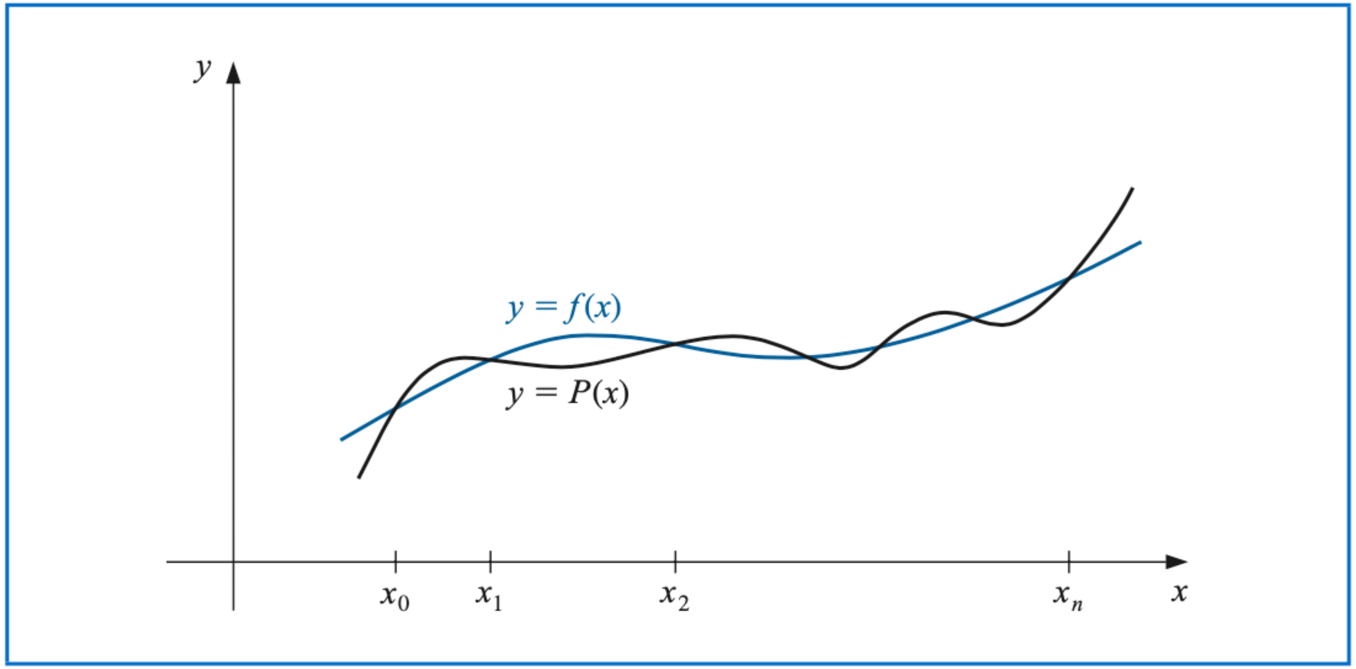
$$R_n(x) = \frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x-x_0)^{n+1}.$$

3.2. Lagrange interpolation polynomial

Theorem 3.2 (Weierstrass approximation theorem)

Suppose that f is defined and continuous on $[a, b]$. For each $\epsilon > 0$, there exists a polynomial $P(x)$, with the property that

$$|f(x) - P(x)| < \epsilon, \text{ for all } x \text{ in } [a, b]$$



3.2. Lagrange interpolation polynomial

Theorem 3.3 (Lagrange interpolation polynomial)

If $x_0, x_1, \cdots, x_n$ are $n+1$ distinct numbers and f is a function whose values are given at these numbers, then a unique polynomial $P(x)$ of degree at most n exists with

$$f(x_k) = P(x_k), \text{ for each } k = 0, 1, \cdots, n.$$

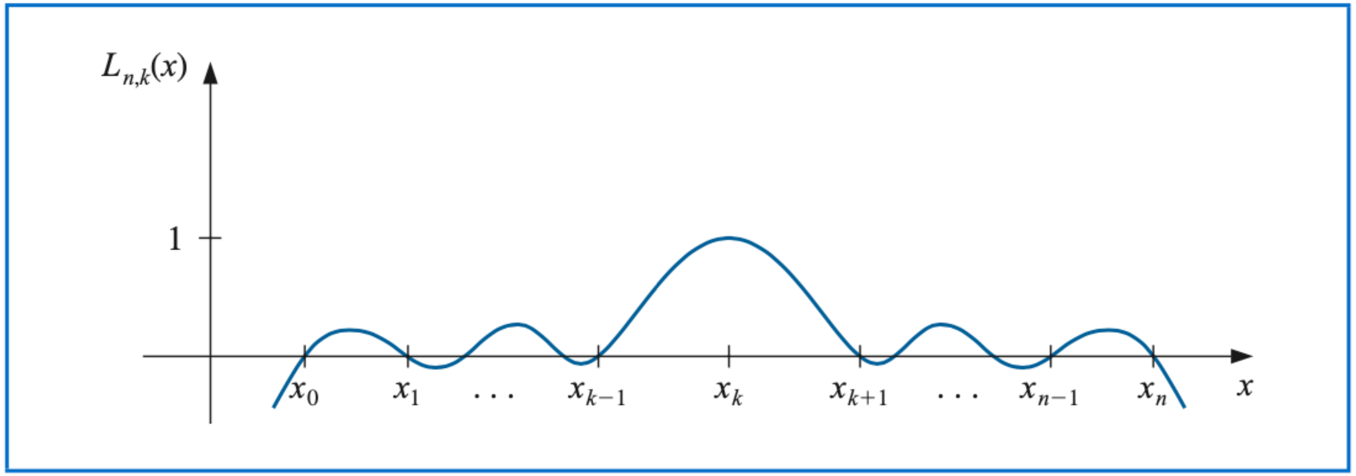
This polynomial is given by

$$P(x) = f(x_0)L_{n,0}(x) + \cdots + f(x_n)L_{n,n}(x) = \sum_{k=0}^{n} f(x_k)L_{n,k}(x),$$

where, for each $k = 0, 1, \cdots, n$,

$$L_{n,k} = \frac{(x-x_0)(x-x_1)\cdots(x-x_{k-1})(x-x_{k+1})\cdots(x-x_n)}{(x_k-x_0)(x_k-x_1)\cdots(x_k-x_{k-1})(x_k-x_{k+1})\cdots(x_k-x_n)} = \prod_{i=0, i\neq k}^{n} \frac{(x-x_i)}{(x_k-x_i)}.$$

3.2. Lagrange interpolation polynomial



Theorem 3.4

Suppose $x_0, x_1, \cdots, x_n$ are distinct numbers in the interval $[a, b]$ and $f \in C^{n+1}[a, b]$. Then, for each x in $[a, b]$, a number $\xi(x)$ between $x_0, x_1, \cdots, x_n$, and hence in (a, b), exist with

$$f(x) = P(x) + \frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x - x_0)(x - x_1)\cdots(x - x_n),$$

where $P(x)$ is the interpolating polynomial given in Thm 2.2.

3.2. Lagrange interpolation polynomial

Example 3.1

Suppose a table is to be prepared for the function $f(x) = e^x$ for x in $[0, 1]$. Assume the number of decimal places to be given per entry is $d \geq 8$ and that the difference between adjacent x-values, the step size, is h. What step size h will ensure that linear interpolation gives an absolute error of at most 10^{-6} for all x in $[0, 1]$?

3.3. Divide difference

Suppose that $P_n(x)$ is the nth order Lagrange polynomial such that $P_n(x_k) = f(x_k),\ k = 0, \cdots, n$
Let $P_n(x) = a_0 + a_1(x - x_0) + \cdots + a_n(x - x_0)\cdots(x - x_{n-1})$. Then,

$$P_n(x_0) = a_0 = f(x_0) = f[x_0]$$

$$P_n(x_1) = a_0 + a_1(x_1 - x_0) = f(x_1) \implies a_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

$$\vdots$$

Define the zeroth divided difference of f with respect to x_i:

$$f[x_i] = f(x_i)$$

The first divided difference of x with respect to x_i and x_{i+1}:

$$f[x_i, x_{i+1}] = \frac{f[x_{i+1}] - f[x_i]}{x_{i+1} - x_i} (= f[x_{i+1}, x_i])$$

$$\vdots$$

3.3. Divide difference

The kth divided difference of x with respect to $x_i,\, x_{i+1}, \cdots, x_{i+k}$:

$$f[x_i,\, x_{i+1}, \cdots, x_{i+k}] = \frac{f[x_{i+1},\, x_{i+2}, \cdots, x_{i+k}] - f[x_i,\, x_{i+1}, \cdots, x_{i+k-1}]}{x_{i+k} - x_i}$$

$\therefore P_n(x) = f[x_0] + \sum_{k=0}^{n} f[x_0,\, x_1, \cdots, x_k](x - x_0)(x - x_1)\cdots(x - x_{k-1})$: Newton's divided difference formula

Let $x = x_0 + sh$ where $h = x_{i+1} - x_i$, then we can express $P_n(x)$ compactly as

$$P_n(x) = P_n(x_0 + sh) = f[x_0] + \sum_{k=1}^{n} \binom{s}{k} \Delta^k f(x_0)$$

$$\left(\binom{s}{k} = \frac{s(s-1)\cdots(s-k+1)}{k!},\ f[x_0,\, x_1, \cdots, x_k] = \frac{1}{k!h^k}\Delta^k f(x_0) \right)$$

3.3. Divide difference

Theorem 3.5

Suppose $f \in C^n(a,b)$ and $x_0\, x_1, \cdots, x_n$ are distinct numbers in $[a,b]$. Then a number exists in (a,b) with

$$f[x_0\, x_1, \cdots, x_n] = \frac{f^{(n)}(\xi)}{n!}$$

3.4. Hermite interpolation

Theorem 3.6

If $f \in C^1[a,b]$ and $x_0, \cdots, x_n \in [a,b]$ are distinct, the unique polynomial of least degree agreeing with f and f' at $x_0, \cdots, x_n$ is the Hermite polynomial of degree at most $2n+1$ given by

$$H_{2n+1}(x) = \sum_{j=0}^{n} f(x_j) H_{n,j}(x) + \sum_{j=0}^{n} f'(x_j) \hat{H}_{n,j}(x),$$

where, for $L_{n,j}(x)$ denoting the jth Lagrange coefficient polynomial of degree n, we have

$$H_{n,j}(x) = [1 - 2(x - x_j) L'_{n,j}(x_j)] L^2_{n,j}(x) \text{ and } \hat{H}_{n,j}(x) = (x - x_j) L^2_{n,j}(x).$$

Moreover, if $f \in C^{2n+2}[a,b]$, then

$$f(x) = H_{2n+1}(x) + \frac{(x - x_0)^2 \cdots (x - x_n)^2}{(2n+2)!} f^{(2n+2)}(\xi(x))$$

3.4. Hermite interpolation

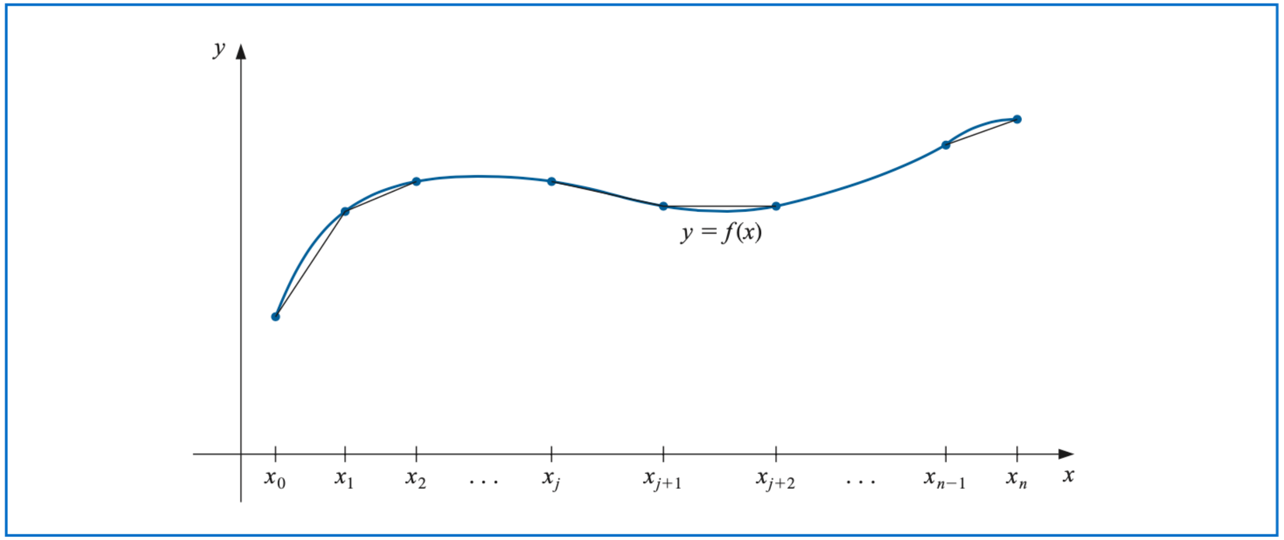
⟨Hermite polynomial using divided difference⟩

Construct $\{z_i\}_{i=0}^{2n+1}$ such that $z_{2i} = z_{2i+1} = x_i$, $i = 0, \cdots, n$.

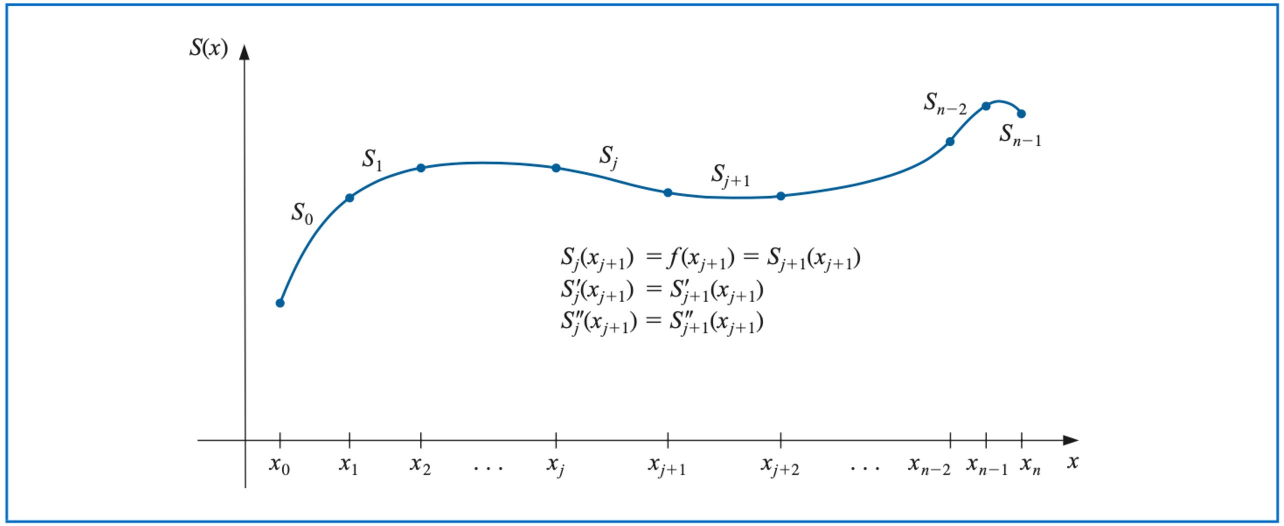
z	$f(z)$	First divided differences	Second divided differences
$z_0 = x_0$	$f[z_0] = f(x_0)$		
		$f[z_0, z_1] = f'(x_0)$	
$z_1 = x_0$	$f[z_1] = f(x_0)$		$f[z_0, z_1, z_2] = \dfrac{f[z_1, z_2] - f[z_0, z_1]}{z_2 - z_0}$
		$f[z_1, z_2] = \dfrac{f[z_2] - f[z_1]}{z_2 - z_1}$	
$z_2 = x_1$	$f[z_2] = f(x_1)$		$f[z_1, z_2, z_3] = \dfrac{f[z_2, z_3] - f[z_1, z_2]}{z_3 - z_1}$
		$f[z_2, z_3] = f'(x_1)$	
$z_3 = x_1$	$f[z_3] = f(x_1)$		$f[z_2, z_3, z_4] = \dfrac{f[z_3, z_4] - f[z_2, z_3]}{z_4 - z_2}$
		$f[z_3, z_4] = \dfrac{f[z_4] - f[z_3]}{z_4 - z_3}$	
$z_4 = x_2$	$f[z_4] = f(x_2)$		$f[z_3, z_4, z_5] = \dfrac{f[z_4, z_5] - f[z_3, z_4]}{z_5 - z_3}$
		$f[z_4, z_5] = f'(x_2)$	
$z_5 = x_2$	$f[z_5] = f(x_2)$		

$$H_{2n+1}(x) = f[z_0] + \sum_{k=1}^{2n+1} f[z_0, \cdots, z_k](x - z_0)(x - z_1)\cdots(x - z_{k-1})$$

3.5. Cubic spline interpolation



Piecewise-polynomial approximation



Piecewise cubic spline

3.5. Cubic spline interpolation

Definition 3.1

Given a function f is defined on $[a, b]$ and a set of nodes $a = x_0 < x_1 < \cdots < x_n = b$, a **cubic spline interpolant** S for f is a function that satisfies the following conditions:

1. $S(x)$ is a cubic polynomial, denoted $S_j(x)$, on the subinterval $[x_j, x_{j+1}]$ for each $j = 0,\ 1, \cdots, n-1$;
2. $S_j(x) = f(x_j)$ and $S_j(x_{j+1}) = f(x_{j+1})$ for each $j = 0,\ 1, \cdots, n-1$;
3. $S_{j+1}(x_{j+1}) = S_j(x_{j+1})$ for each $j = 0,\ 1, \cdots, n-2$;
4. $S'_{j+1}(x_{j+1}) = S'_j(x_{j+1})$ for each $j = 0,\ 1, \cdots, n-2$;
5. $S''_{j+1}(x_{j+1}) = S''_j(x_{j+1})$ for each $j = 0,\ 1, \cdots, n-2$;
6. One of the following sets of boundary conditions is satisfied:
   (a) $S''(x_0) = S''(x_n) = 0$ (natural or free boundary);
   (b) $S'(x_0) = f'(x_0)$ and $S'(x_n) = f'(x_n)$ (clamped boundary).

3.5. Cubic spline interpolation

Let $S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3$ for each $j = 0,\ 1, \cdots, n$.
Since $S_j(x_j) = f(x_j)$,

$$a_j = f(x_j).$$

Define $a_n = f(x_n)$ then

$$a_j = f(x_j) \text{ for } j = 0,\ 1, \cdots, n.$$

Since $S_{j+1}(x_{j+1}) = S_j = (x_{j+1})$ and let $h_j = x_{j+1} - x_j$, then we can get

$$a_{j+1} = a_j + b_j h_j + c_j h_j^2 + d_j h_j^3,\ j = 0,\ 1, \cdots, n-1.$$

Since $S_j'(x) = b_j + 2c_j(x - x_j) + 3d_j(x - x_j)^2$,

$$S_j'(x_j) = b_j,\ j = 0,\ 1, \cdots, n.$$

Since $S_{j+1}'(x_{j+1}) = S_j'(x_{j+1})$,

$$b_{j+1} = b_j + 2c_j h_j + 3d_j h_j^2,\ j = 0,\ 1, \cdots, n-1.$$

3.5. Cubic spline interpolation

Since $S_j''(x) = 2c_j + 6d_j(x - x_j)$,

$$S_j''(x_j) = 2c_j,\ j = 0, \cdots, n-1 \implies c_j = \frac{S_j''(x_j)}{2}$$

Define $c_n = \dfrac{S_n''(x_n)}{2}$, then

$$c_j = \frac{S_j''(x_j)}{2},\ j = 0, \cdots, n$$

Since $S_{j+1}''(x_{j+1}) = S_j''(x_{j+1})$, $c_{j+1} = c_j + 3d_jh_j$, $j = 0, \cdots, n-1$,

$$\therefore d_j = \frac{(c_{j+1} - c_j)}{3h_j},\ j = 0, \cdots, n-1$$

3.5. Cubic spline interpolation

Since $a_{j+1} = a_j + b_j h_j + c_j h_j^2 + d_j h_j^3$, $j = 0, \cdots, n-1$,

$$a_{j+1} = a_j + b_j h_j + \frac{h_j^2}{3}(c_{j+1} + 2c_j) \tag{1}$$

Since $b_{j+1} = b_j + 2c_j h_j + 3d_j h_j^2$, $j = 0, \cdots, n-1$,

$$b_{j+1} = b_j + h_j(c_j + c_{j+1}) \tag{2}$$

Solving (1) for b_j, we have

$$b_j = \frac{1}{h_j}(a_{j+1} - a_j) - \frac{h_j}{3}(c_{j+1} + 2c_j) \tag{3}$$

From (2), we have

$$b_j = b_{j-1} + h_{j-1}(c_{j-1} + c_j) \tag{4}$$

3.5. Cubic spline interpolation

Substituting (3) to (4), we can get

$$h_{j-1}c_{j-1} + 2(h_{j-1} + h_j)c_j + h_j c_{j+1} = \frac{3}{h_j}(a_{j+1} - a_j) - \frac{3}{h_{j-1}}(a_j - a_{j-1})$$

Let $g_j = \frac{3}{h_j}(a_{j+1} - a_j) - \frac{3}{h_{j-1}}(a_j - a_{j-1})$, $j = 0, \cdots, n-1$ and choose $c_0 = 0 = c_n$ then we get,

$$\begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 \\ h_0 & 2(h_0+h_1) & h_1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & h_1 & 2(h_1+h_2) & h_2 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \\ 0 & 0 & 0 & 0 & \cdots & h_{n-1} & 2(h_{n-1}+h_n) & h_n \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{n-1} \\ c_n \end{bmatrix} = \begin{bmatrix} 0 \\ g_1 \\ g_2 \\ \vdots \\ g_{n-1} \\ 0 \end{bmatrix}$$

It can be solved by the Thomas algorithm.

3.5. Cubic spline interpolation

$\langle$Thomas algorithm$\rangle$

$$\begin{pmatrix} b_1 & c_1 & 0 & 0 \\ a_2 & b_2 & c_2 & 0 \\ 0 & a_3 & b_3 & c_3 \\ 0 & 0 & a_4 & b_4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{pmatrix}$$

Row 1: $b_1 x_1 + c_1 x_2 = r_1 \implies x_1 + \frac{c_1}{b_1} x_2 = \frac{r_1}{b_1}$

Choose $\frac{c_1}{b_1} = \gamma_1$ and $\frac{r_1}{b_1} = \rho_1$, we can get

$$x_1 + \gamma_1 x_2 = \rho_1 \tag{1}$$

Row 2: $a_2 x_1 + b_2 x_2 + c_2 x_3 = r_2$

Substituting (1) in this equation we can get $x_2 + \frac{c_2}{b_2 - a_2 x_1} x_3 = \frac{r_2 - a_2 \rho_1}{b_2 - a_2 \gamma_1}$

Choose $\frac{c_2}{b_2 - a_2 x_1} = \gamma_2$, $\frac{r_2 - a_2 \rho_1}{b_2 - a_2 \gamma_1} = \rho_2$, we can get

$$x_2 + \gamma_2 x_3 = \rho_2 \tag{2}$$

3.5. Cubic spline interpolation

⟨Thomas algorithm⟩

$$\begin{pmatrix} b_1 & c_1 & 0 & 0 \\ a_2 & b_2 & c_2 & 0 \\ 0 & a_3 & b_3 & c_3 \\ 0 & 0 & a_4 & b_4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{pmatrix}$$

Row 3: $a_3 x_2 + b_3 x_3 + c_3 x_4 = r_3$

Substituting (2) in this equation we can get $x_3 + \dfrac{c_3}{b_3 - a_3\gamma_2} x_4 = \dfrac{r_3 - a_3\rho_2}{b_3 - a_2\gamma_2}$

Choose $\dfrac{c_3}{b_3 - a_3\gamma_2} = \gamma_3$, $\dfrac{r_3 - a_3\rho_2}{b_3 - a_3\gamma_2} = \rho_3$, we can get

$$x_3 + \gamma_3 x_4 = \rho_3 \tag{3}$$

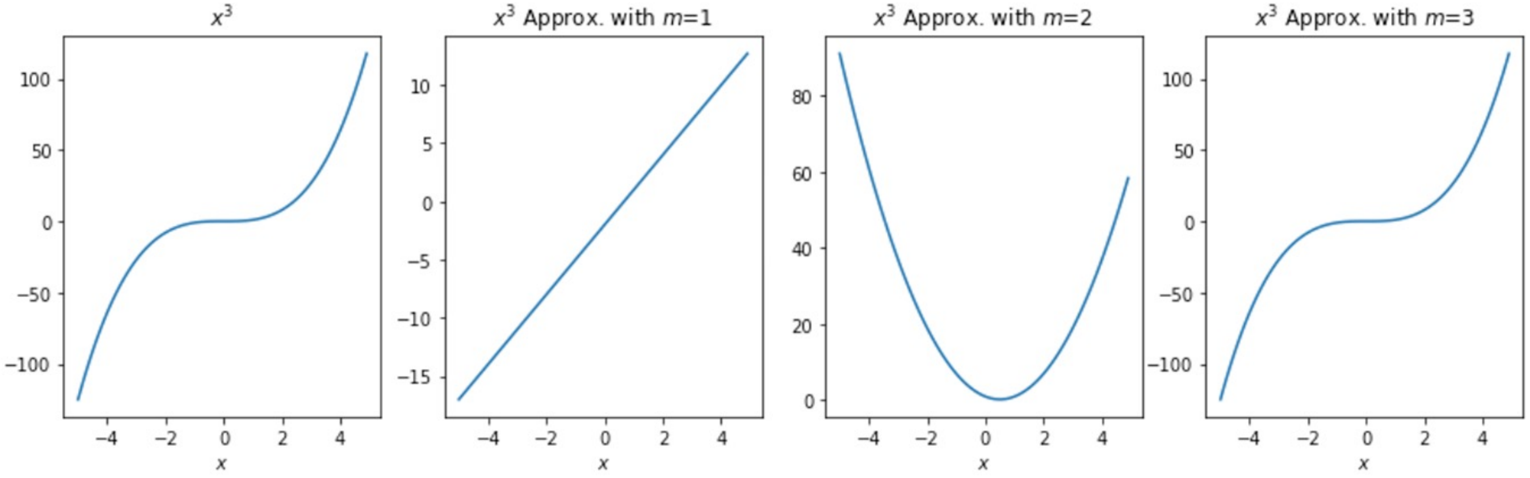
Row 4: $a_4 x_3 + b_4 x_4 = r_4$

Substituting (3) in this equation we can get

$$x_4 = \frac{r_4 - a_4\rho_3}{b_4 - a_4\gamma_3} = \rho_4 \tag{4}$$

3.6. Experiments

- **[노트북] 테일러 시리즈를 이용한 근사함수 계산**

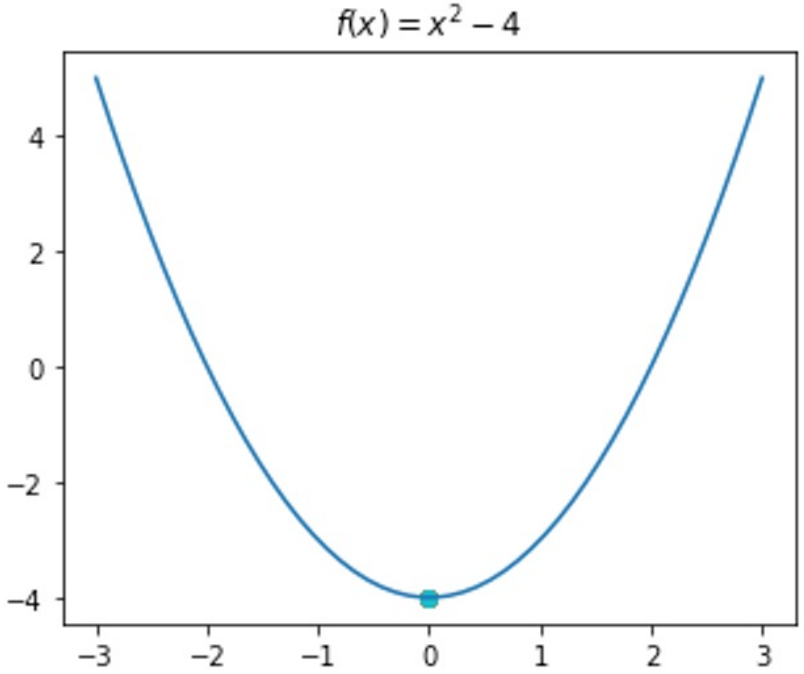


Google Colaboratory

https://colab.research.google.com/github/pp4e-book/pp4e-book....

3.6. Experiments

- **[노트북] scipy를 이용해 뉴턴법으로 최소값 찾기**



- **울산 지역 주택 가격 선형 회귀 예측 모델에서 최소값 찾기**
- **울산 토마토 가장 키우기 좋은 달 선형 회귀 예측 모델의 테일러 전개를 통한 근사 정도 시각화**

3.6. Experiments

Table 3.18

x	0.9	1.3	1.9	2.1	2.6	3.0	3.9	4.4	4.7	5.0	6.0	7.0	8.0	9.2	10.5	11.3	11.6	12.0	12.6	13.0	13.3
$f(x)$	1.3	1.5	1.85	2.1	2.6	2.7	2.4	2.15	2.05	2.1	2.25	2.3	2.25	1.95	1.4	0.9	0.7	0.6	0.5	0.4	0.25

Figure 3.12