Taylor Expansion

One Variable

$f:[a,b]\to R;\ f(x)$, n+1 times differentiable; $f^{(n)}$ continuous.

$$f(x)=\ f(a)+f'(a)(x-a)+\frac{f''(a)}{2!}+\cdots+\frac{f^{(n)}}{n!}(x-a)^n+R_{n+1}(x)$$

$$R_{n+1}(x)=f(x)-\sum_{k=0}^{n}\frac{f^{(k)}}{k!}(x-a)^i=\frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x-a)^{n+1}$$

$$\lim_{x\to a}R_{n+1}=0$$

Two Variable

U convex open set in $R^n;\ f:U\to R$; continuous partial derivatives of all orders up to m+1.

$$D_j^l f:=\frac{\partial^l f}{\partial x_j},\qquad D_{i_1\dots i_k}f:=\frac{\partial^k f}{\partial x_{i_1}\dots\partial x_{i_k}}$$

$$f(\boldsymbol{x})=\sum_{k_1+\cdots+k_n\le m}\frac{(D_1^{k_1}D_2^{k_2}\dots D_n^{k_n}f)(\boldsymbol{x}-\boldsymbol{a})}{k_1!\,k_2!\dots k_n!}(x_1-a_1)^{k_1}(x_2-a_2)^{k_2}\dots(x_n-a_n)^{k_n}+R(\boldsymbol{x})$$

$$f(\boldsymbol{a}+\boldsymbol{x})=\sum_{k_1+\cdots+k_n\le m}\frac{(D_1^{k_1}D_2^{k_2}\dots D_n^{k_n}f)(\boldsymbol{x})}{k_1!\,k_2!\dots k_n!}x_1^{k_1}x_2^{k_2}\dots x_n^{k_n}+R(\boldsymbol{x})$$

Interpolation and Polynomial approximation

Lagrange Interpolation

$$P_n(x)=\sum_{k=0}^{n}f(x_k)L_k(x),\qquad L_k(x)=\prod_{\substack{i=0\\ i\neq k}}^{n}\frac{(x-x_i)}{(x_k-x_i)}$$

$$E_L(x)=f(x)-P_n(x)=\frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x-x_0)\dots(x-x_n)$$

Newton's Divided Difference

$$P_n(x)=f[x_0]+\sum_{k=1}^{n}f[x_0,x_1,\dots x_k](x-x_0)\dots(x-x_{k-1})$$

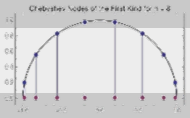
$$f[x_i,x_{i+1}\dots,x_{i+k}]=\frac{f[x_{i+1},x_{i+2},\dots,x_{i+k}]-f[x_i,x_{i+1},\dots,x_{i+k-1}]}{x_{i+k}-x_i}$$

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

x	$f(x)$	First divided differences	Second divided differences	Third divided differences
x_0	$f[x_0]$			
		$f[x_0,x_1]=\frac{f[x_1]-f[x_0]}{x_1-x_0}$		
x_1	$f[x_1]$		$f[x_0,x_1,x_2]=\frac{f[x_1,x_2]-f[x_0,x_1]}{x_2-x_0}$	
		$f[x_1,x_2]=\frac{f[x_2]-f[x_1]}{x_2-x_1}$		$f[x_0,x_1,x_2,x_3]=\frac{f[x_1,x_2,x_3]-f[x_0,x_1,x_2]}{x_3-x_0}$
x_2	$f[x_2]$		$f[x_1,x_2,x_3]=\frac{f[x_2,x_3]-f[x_1,x_2]}{x_3-x_1}$	
		$f[x_2,x_3]=\frac{f[x_3]-f[x_2]}{x_3-x_2}$		$f[x_1,x_2,x_3,x_4]=\frac{f[x_2,x_3,x_4]-f[x_1,x_2,x_3]}{x_4-x_1}$
x_3	$f[x_3]$		$f[x_2,x_3,x_4]=\frac{f[x_3,x_4]-f[x_2,x_3]}{x_4-x_2}$	
		$f[x_3,x_4]=\frac{f[x_4]-f[x_3]}{x_4-x_3}$		$f[x_2,x_3,x_4,x_5]=\frac{f[x_3,x_4,x_5]-f[x_2,x_3,x_4]}{x_5-x_2}$
x_4	$f[x_4]$		$f[x_3,x_4,x_5]=\frac{f[x_4,x_5]-f[x_3,x_4]}{x_5-x_3}$	
		$f[x_4,x_5]=\frac{f[x_5]-f[x_4]}{x_5-x_4}$		
x_5	$f[x_5]$			

Chebyshev nodes

$$x_k=\frac{a+b}{2}+\frac{b-a}{2}\cos\left(\frac{2k+1}{2n}\pi\right),\qquad k=0,\dots,n-1$$



Spline Interpolation

$$S_i(x_{i+1})=\ S_{i+1}(x_{i+1})=\ f(x_{i+1})$$

$$S_i'(x_{i+1})\ =\ S_{i+1}'(x_{i+1}),\ S_i''(x_{i+1})=\ S_{i+1}''(x_{i+1}),\dots,\ S_i^{(k)}(x_{i+1})=\ S_{i+1}^{(k)}(x_{i+1})$$

Natural Cubic Spline

$$S''(x_0)=S''(x_n)=0$$

Clamped Cubic Spline

$$S'(x_0)=f'(x_0),\qquad S'(x_n)=f'(x_n)$$

Linear Spline

$$S_i(x)=f(x_i)+\frac{f(x_{i+1})-f(x_i)}{x_{i+1}-x_i}(x-x_i)$$

Numerical Differentiation

(n+1)-Point Formula

$$f'(x)=\sum_{k=0}^{n}f(x_k)L_k'(x)+D_x\left[\frac{(x-x_0)\dots(x-x_n)}{(n+1)!}\right]f^{(n+1)}(\xi(x))+\frac{(x-x_0)\dots(x-x_n)}{(n+1)!}D_x\left[f^{(n+1)}(\xi(x))\right]$$

$$f'(x_j)=\sum_{k=0}^{n}f(x_k)L_k'(x_j)+\frac{f^{(n+1)}\left(\xi(x_j)\right)}{(n+1)!}\prod_{\substack{k=0\\ k\neq j}}^{n}(x_j-x_k)$$

Three Point Formulas

$$f'(x_0)=\frac{1}{2h}\left[-3f(x_0)+4f(x_0+h)-f(x_0+2h)\right]+\frac{h^2}{3}f'''(\xi_0)$$

$$f'(x_0)=\frac{1}{2h}[f(x_0+h)-f(x_0-h)]-\frac{h^2}{6}f'''(\xi_1)$$

Five Point Formulas

$$f'(x_0)=\frac{1}{12h}\left[f(x_0-2h)-8f(x_0-h)+8f(x_0+h)-f(x_0+2h)\right]+\frac{h^4}{30}f^{(5)}(\xi)$$

$$f'(x_0)=\frac{1}{12h}\left[-25f(x_0)+48f(x_0+h)-36f(x_0+2h)+16f(x_0+3h)-3f(x_0+4h)\right]+\frac{h^4}{5}f^{(5)}(\xi)$$

Second Derivative Formula

$$f(x_0+h)=f(x_0)+f'(x_0)h+\frac{1}{2}f''(x_0)h^2+\frac{1}{6}f'''(x_0)h^3+\frac{1}{24}f^{(4)}(\xi_1)h^4$$

$$f(x_0-h)=f(x_0)-f'(x_0)h+\frac{1}{2}f''(x_0)h^2-\frac{1}{6}f'''(x_0)h^3+\frac{1}{24}f^{(4)}(\xi_{-1})h^4$$

$$f(x_0+h)+f(x_0-h)=2f(x_0)+f''(x_0)h^2+\frac{1}{24}\left[f^{(4)}(\xi_1)+\frac{1}{24}f^{(4)}(\xi_{-1})\right]h^4$$

$$f^{(4)}(\xi)=\tfrac{1}{2}[f^{(4)}(\xi_1)+f^{(4)}(\xi_{-1})]\qquad\text{Intermediate value theorem}$$

$$f''(x_0)=\frac{1}{h^2}[f(x_0-h)-2f(x_0)+f(x_0+h)]-\frac{h^2}{12}f^{(4)}(\xi)$$

Other Derivatives

	Error
First Derivative	
$f'(x_i)=\frac{f(x_{i+1})-f(x_{i-1})}{2h}$	$O(h^2)$
$f'(x_i)=\frac{-f(x_{i+2})+8f(x_{i+1})-8f(x_{i-1})+f(x_{i-2})}{12h}$	$O(h^4)$
Second Derivative	
$f''(x_i)=\frac{f(x_{i+1})-2f(x_i)+f(x_{i-1})}{h^2}$	$O(h^2)$
$f''(x_i)=\frac{-f(x_{i+2})+16f(x_{i+1})-30f(x_i)+16f(x_{i-1})-f(x_{i-2})}{12h^2}$	$O(h^4)$
Third Derivative	
$f'''(x_i)=\frac{f(x_{i+2})-2f(x_{i+1})+2f(x_{i-1})-f(x_{i-2})}{2h^3}$	$O(h^2)$
$f'''(x_i)=\frac{-f(x_{i+3})+8f(x_{i+2})-13f(x_{i+1})+13f(x_{i-1})-8f(x_{i-2})+f(x_{i-3})}{8h^3}$	$O(h^4)$
Fourth Derivative	
$f''''(x_i)=\frac{f(x_{i+2})-4f(x_{i+1})+6f(x_i)-4f(x_{i-1})+f(x_{i-2})}{h^4}$	$O(h^2)$
$f''''(x_i)=\frac{-f(x_{i+3})+12f(x_{i+2})+39f(x_{i+1})+56f(x_i)-39f(x_{i-1})+12f(x_{i-2})+f(x_{i-3})}{6h^4}$	$O(h^4)$

Derivation using Forward Differences

$$P_n(x)=P_n(x_0+sh)=f_0+s\Delta f_0+\frac{s(s-1)}{2}\Delta f_0+\cdots\frac{s(s-1)\dots(s-(n-1))}{n!}\Delta^n f_0=\sum_{k=0}^{n}\binom{s}{k}\Delta^k f_0$$

$$f'(x)\simeq\frac{dp}{dx}=\frac{dp}{ds}\frac{ds}{dx}=\frac{1}{h}\left[\Delta f_0-\frac{1}{2}\Delta^2 f_0+\cdots+\frac{(-1)^{n-1}}{n}\Delta^n f_0\right]$$

$$E_n(x)=\frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x-x_0)\dots(x-x_n)=f^{(n+1)}(\xi(x))\frac{h^{n+1}s(s-1)\dots(s-n)}{(n+1)!}$$

$$E_n'(x)=\frac{1}{h}\left[h^{n+1}\frac{f^{(n+1)}(\xi(x))}{(n+1)!}\sum_{i=0}^{n}\prod_{\substack{j=0\\ j\neq i}}^{n}(s-j)+\frac{h^{n+1}s(s-1)\dots(s-n)}{(n+1)!}\cdot\frac{d}{ds}\left(f^{(n+1)}(\xi(x))\right)\right]$$

$$\to E_{f'}(x)=E_n'(x)=O(h^n)$$

Numerical Integral

$$\int_a^b f(x)dx = \int_a^b \sum_{i=0}^{n} f(x_i)L_i(x)dx + \frac{1}{(n+1)!}\int_a^b \prod_{i=0}^{n}(x-x_i)\,f^{(n+1)}(\xi(x))$$

$$E(f) = \frac{1}{(n+1)!}\int_a^b \prod_{i=0}^{n}(x-x_i)\,f^{(n+1)}(\xi(x)), \qquad a_i = \int_a^b L_i(x)dx$$

$$\int_a^b f(x)dx = \sum_{i=0}^{n} a_i f(x_i) + E(f)$$

Closed Newton-Cotes Formulas

n = 1: Trapezoidal rule

$$\int_{x_0}^{x_1} f(x)dx = \frac{h}{2}[f(x_0)+f(x_1)] - \frac{h^3}{12}f''(\xi), \qquad h = b-a$$

n = 2: Simpson's rule

$$\int_{x_0}^{x_2} f(x)dx = \frac{h}{3}[f(x_0)+4f(x_1)+f(x_2)] - \frac{h^5}{90}f^{(4)}(\xi), \qquad h = \frac{b-a}{2}$$

n = 3: Simpson's Three-Eighths rule

$$\int_{x_0}^{x_3} f(x)dx = \frac{3h}{8}[f(x_0)+3f(x_1)+3f(x_2)+f(x_3)] - \frac{3h^5}{80}f^{(4)}(\xi)$$

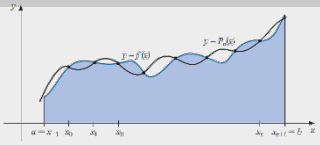
n = 4:

$$\int_{x_0}^{x_4} f(x)dx = \frac{2h}{45}[7f(x_0)+32f(x_1)+12f(x_2)+32f(x_3)+7(x_4)] - \frac{8h^7}{945}f^{(6)}(\xi)$$

Open Newton-Cotes Formulas

n = 0: Midpoint rule

$$\int_{x_{-1}}^{x_1} f(x)dx = 2hf(x_0) + \frac{h^3}{3}f''(\xi)$$



n = 1:

$$\int_{x_{-1}}^{x_2} f(x)dx = \frac{3h}{2}[f(x_0)+f(x_1)] + \frac{3h^3}{4}f''(\xi)$$

n = 2:

$$\int_{x_{-1}}^{x_3} f(x)dx = \frac{4h}{3}[2f(x_0)-f(x_1)+2f(x_2)] + \frac{14h^5}{45}f^{(4)}(\xi)$$

n = 3:

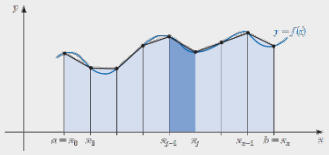
$$\int_{x_{-1}}^{x_4} f(x)dx = \frac{5h}{24}[11f(x_0)+f(x_1)+f(x_2)+11f(x_3)] + \frac{95h^5}{144}f^{(4)}(\xi)$$

Composite Numerical Integration

Composite Trapezoidal Rule

$$\int_a^b f(x)dx = \frac{h}{2}\left[f(a)+2\sum_{j=1}^{n-1} f(x_j)+f(b)\right] + \frac{b-a}{12}h^2 f''(\mu)$$

$$T(n) = h\left[\frac{f_1+f_{n+1}}{2} + \sum_{i=2}^{n} f_i\right]$$

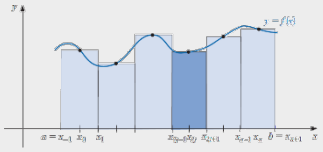


$$T(2n) = \frac{1}{2}[T(n)+M(n)]$$

Composite Midpoint rule

$$\int_a^b f(x)dx = 2h\sum_{j=0}^{\frac{n}{2}} f(x_{2j}) + \frac{b-a}{6}h^2 f''(\mu)$$

$$M(n) = h\sum_{i=1}^{n} f(\bar{x}_i)$$

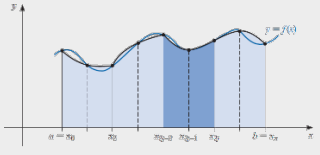


Composite Simpson's Rule

$$\int_a^b f(x)dx = \frac{h}{3}\left[f(a)+2\sum_{j=1}^{\frac{n}{2}-1} f(x_{2j})+4\sum_{j=1}^{\frac{n}{2}} f(x_{2j-1})+f(b)\right] + \frac{b-a}{180}h^4 f^{(4)}(\mu)$$

$$S(n) = \frac{2}{3}h\sum_{i=1}^{n} f(\bar{x}_i) + \frac{h}{3}\left[\frac{f_1+f_{n+1}}{2} + \sum_{i=2}^{n} f(x_i)\right]$$

$$S(n) = \frac{2}{3}M(n) + \frac{1}{3}T(n)$$



Simpson's Rule Proof

$$\int_a^b f(x)\,dx \approx \int_a^b f(x_1)\frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)} + f(x_2)\frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)} + f(x_3)\frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)}\,dx$$

$$\int_a^b p(x)\,dx = hf(x_0)\int_0^2 \frac{(t-1)(t-2)}{(0-1)(0-2)}\,dt + hf(x_1)\int_0^2 \frac{(t-0)(t-2)}{(1-0)(1-2)}\,dt + hf(x_2)\int_0^2 \frac{(t-0)(t-1)}{(2-0)(2-1)}\,dt$$

$$\int_a^b p(x)\,dx = hf(x_0)\frac{1}{3} + hf(x_1)\frac{4}{3} + hf(x_2)\frac{1}{3}$$

Interpolating Polynomial Error Proof

Proof Note first that if $x = x_k$, for any $k = 0, 1, \ldots, n$, then $f(x_k) = P(x_k)$, and choosing $\xi(x_k)$ arbitrarily in (a, b) yields Eq. (3.3).

If $x \neq x_k$, for all $k = 0, 1, \ldots, n$, define the function g for t in $[a, b]$ by

$$g(t) = f(t) - P(t) - [f(x) - P(x)]\frac{(t-x_0)(t-x_1)\cdots(t-x_n)}{(x-x_0)(x-x_1)\cdots(x-x_n)}$$

$$= f(t) - P(t) - [f(x) - P(x)]\prod_{i=0}^{n}\frac{(t-x_i)}{(x-x_i)},$$

Since $f \in C^{n+1}[a, b]$, and $P \in C^{\infty}[a, b]$, it follows that $g \in C^{n+1}[a, b]$. For $t = x_k$, we have

$$g(x_k) = f(x_k) - P(x_k) - [f(x) - P(x)]\prod_{i=0}^{n}\frac{(x_k-x_i)}{(x-x_i)} = 0 - [f(x) - P(x)] \cdot 0 = 0.$$

Moreover,

$$g(x) = f(x) - P(x) - [f(x) - P(x)]\prod_{i=0}^{n}\frac{(x-x_i)}{(x-x_i)} = f(x) - P(x) - [f(x) - P(x)] = 0.$$

Thus $g \in C^{n+1}[a, b]$, and g is zero at the $n + 2$ distinct numbers $x, x_0, x_1, \ldots, x_n$. By Generalized Rolle's Theorem 1.10, there exists a number ξ in (a, b) for which $g^{(n+1)}(\xi) = 0$. So

$$0 = g^{(n+1)}(\xi) = f^{(n+1)}(\xi) - P^{(n+1)}(\xi) - [f(x) - P(x)]\frac{d^{n+1}}{dt^{n+1}}\left[\prod_{i=0}^{n}\frac{(t-x_i)}{(x-x_i)}\right]_{t=\xi}. \qquad (3.4)$$

However $P(x)$ is a polynomial of degree at most n, so the $(n+1)$st derivative, $P^{(n+1)}(x)$, is identically zero. Also, $\prod_{i=0}^{n}[(t-x_i)/(x-x_i)]$ is a polynomial of degree $(n+1)$, so

$$\prod_{i=0}^{n}\frac{(t-x_i)}{(x-x_i)} = \left[\frac{1}{\prod_{i=0}^{n}(x-x_i)}\right]t^{n+1} + (\text{lower-degree terms in } t),$$

and

$$\frac{d^{n+1}}{dt^{n+1}}\prod_{i=0}^{n}\frac{(t-x_i)}{(x-x_i)} = \frac{(n+1)!}{\prod_{i=0}^{n}(x-x_i)}.$$

Equation (3.4) now becomes

$$0 = f^{(n+1)}(\xi) - 0 - [f(x) - P(x)]\frac{(n+1)!}{\prod_{i=0}^{n}(x-x_i)},$$

and, upon solving for $f(x)$, we have

$$f(x) = P(x) + \frac{f^{(n+1)}(\xi)}{(n+1)!}\prod_{i=0}^{n}(x-x_i). \qquad \blacksquare\ \blacksquare\ \blacksquare$$

$f[x_0, x_1, \ldots, x_n] = f^{(n)}(\xi)/n!$ Proof

Proof Let

$$g(x) = f(x) - P_n(x).$$

Since $f(x_i) = P_n(x_i)$ for each $i = 0, 1, \ldots, n$, the function g has $n+1$ distinct zeros in $[a, b]$. Generalized Rolle's Theorem 1.10 implies that a number ξ in (a, b) exists with $g^{(n)}(\xi) = 0$, so

$$0 = f^{(n)}(\xi) - P_n^{(n)}(\xi).$$

Since $P_n(x)$ is a polynomial of degree n whose leading coefficient is $f[x_0, x_1, \ldots, x_n]$,

$$P_n^{(n)}(x) = n!f[x_0, x_1, \ldots, x_n],$$

for all values of x. As a consequence,

$$f[x_0, x_1, \ldots, x_n] = \frac{f^{(n)}(\xi)}{n!}. \qquad \blacksquare\ \blacksquare\ \blacksquare$$

Natural Spline Algorithm

To construct the cubic spline interpolant S for the function f, defined at the numbers $x_0 < x_1 < \cdots < x_n$, satisfying $S''(x_0) = S''(x_n) = 0$:

INPUT $n; x_0, x_1, \ldots, x_n; a_0 = f(x_0), a_1 = f(x_1), \ldots, a_n = f(x_n)$.

OUTPUT a_j, b_j, c_j, d_j for $j = 0, 1, \ldots, n-1$.

(*Note:* $S(x) = S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3$ *for* $x_j \le x \le x_{j+1}$.)

Step 1 For $i = 0, 1, \ldots, n-1$ set $h_i = x_{i+1} - x_i$.

Step 2 For $i = 1, 2, \ldots, n-1$ set

$$\alpha_i = \frac{3}{h_i}(a_{i+1} - a_i) - \frac{3}{h_{i-1}}(a_i - a_{i-1}).$$

Step 3 Set $l_0 = 1$; (*Steps 3, 4, 5, and part of Step 6 solve a tridiagonal linear system using a method described in Algorithm 6.7.*)
$\mu_0 = 0$;
$z_0 = 0$.

Step 4 For $i = 1, 2, \ldots, n-1$
set $l_i = 2(x_{i+1} - x_{i-1}) - h_{i-1}\mu_{i-1}$;
$\mu_i = h_i / l_i$;
$z_i = (\alpha_i - h_{i-1}z_{i-1})/l_i$.

Step 5 Set $l_n = 1$;
$z_n = 0$;
$c_n = 0$.

Step 6 For $j = n-1, n-2, \ldots, 0$
set $c_j = z_j - \mu_j c_{j+1}$;
$b_j = (a_{j+1} - a_j)/h_j - h_j(c_{j+1} + 2c_j)/3$;
$d_j = (c_{j+1} - c_j)/(3h_j)$.

Step 7 OUTPUT $(a_j, b_j, c_j, d_j$ for $j = 0, 1, \ldots, n-1)$;
STOP.

$$A = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ h_0 & 2(h_0 + h_1) & h_1 & & \vdots \\ 0 & h_1 & 2(h_1 + h_2) & \ddots & 0 \\ \vdots & & \ddots & \ddots & h_{n-1} \\ 0 & \cdots & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 0 \\ \frac{3}{h_1}(a_2 - a_1) - \frac{3}{h_0}(a_1 - a_0) \\ \vdots \\ \frac{3}{h_{n-1}}(a_n - a_{n-1}) - \frac{3}{h_{n-2}}(a_{n-1} - a_{n-2}) \\ 0 \end{bmatrix} \quad \text{and} \quad \mathbf{x} = \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{bmatrix}$$

Clamped Spline Algorithm

To construct the cubic spline interpolant S for the function f defined at the numbers $x_0 < x_1 < \cdots < x_n$, satisfying $S'(x_0) = f'(x_0)$ and $S'(x_n) = f'(x_n)$:

INPUT $n; x_0, x_1, \ldots, x_n$; $a_0 = f(x_0), a_1 = f(x_1), \ldots, a_n = f(x_n)$; $FPO = f'(x_0)$; $FPN = f'(x_n)$.

OUTPUT a_j, b_j, c_j, d_j for $j = 0, 1, \ldots, n-1$.

(*Note:* $S(x) = S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3$ *for* $x_j \le x \le x_{j+1}$.)

Step 1 For $i = 0, 1, \ldots, n-1$ set $h_i = x_{i+1} - x_i$.

Step 2 Set $\alpha_0 = 3(a_1 - a_0)/h_0 - 3FPO$;
$\alpha_n = 3FPN - 3(a_n - a_{n-1})/h_{n-1}$.

Step 3 For $i = 1, 2, \ldots, n-1$
set $\alpha_i = \frac{3}{h_i}(a_{i+1} - a_i) - \frac{3}{h_{i-1}}(a_i - a_{i-1})$.

Step 4 Set $l_0 = 2h_0$; (*Steps 4,5,6, and part of Step 7 solve a tridiagonal linear system using a method described in Algorithm 6.7.*)
$\mu_0 = 0.5$;
$z_0 = \alpha_0 / l_0$.

Step 5 For $i = 1, 2, \ldots, n-1$
set $l_i = 2(x_{i+1} - x_{i-1}) - h_{i-1}\mu_{i-1}$;
$\mu_i = h_i / l_i$;
$z_i = (\alpha_i - h_{i-1}z_{i-1})/l_i$.

Step 6 Set $l_n = h_{n-1}(2 - \mu_{n-1})$;
$z_n = (\alpha_n - h_{n-1}z_{n-1})/l_n$;
$c_n = z_n$.

Step 7 For $j = n-1, n-2, \ldots, 0$
set $c_j = z_j - \mu_j c_{j+1}$;
$b_j = (a_{j+1} - a_j)/h_j - h_j(c_{j+1} + 2c_j)/3$;
$d_j = (c_{j+1} - c_j)/(3h_j)$.

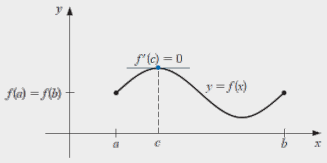
Step 8 OUTPUT $(a_j, b_j, c_j, d_j$ for $j = 0, 1, \ldots, n-1)$;
STOP.

$$A = \begin{bmatrix} 2h_0 & h_0 & 0 & \cdots & 0 \\ h_0 & 2(h_0 + h_1) & h_1 & & \vdots \\ 0 & h_1 & 2(h_1 + h_2) & \ddots & 0 \\ \vdots & & \ddots & \ddots & h_{n-1} \\ 0 & \cdots & 0 & h_{n-1} & 2h_{n-1} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} \frac{3}{h_0}(a_1 - a_0) - 3f'(a) \\ \frac{3}{h_1}(a_2 - a_1) - \frac{3}{h_0}(a_1 - a_0) \\ \vdots \\ \frac{3}{h_{n-1}}(a_n - a_{n-1}) - \frac{3}{h_{n-2}}(a_{n-1} - a_{n-2}) \\ 3f'(b) - \frac{3}{h_{n-1}}(a_n - a_{n-1}) \end{bmatrix}, \quad \text{and} \quad \mathbf{x} = \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{bmatrix}$$

Useful Theorems

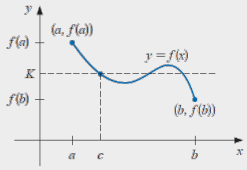
Generalized Rolle's Theorem

Suppose $f \in C[a, b]$ is n times differentiable on (a, b). If $f(x) = 0$ at the $n + 1$ distinct numbers $a \le x_0 < x_1 < \ldots < x_n \le b$, then a number c in (x_0, x_n), and hence in (a, b), exists with $f^{(n)}(c) = 0$.



Intermediate Value Theorem

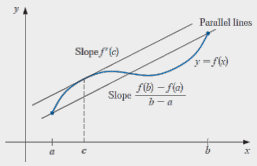
If $f \in C[a, b]$ and K is any number between $f(a)$ and $f(b)$, then there exists a number c in (a, b) for which $f(c) = K$.



Mean Value Theorem

If $f \in C[a, b]$ and f is differentiable on (a, b), then a number c in (a, b) exists with (See Figure 1.4.)

$$f'(c) = \frac{f(b) - f(a)}{b - a}.$$



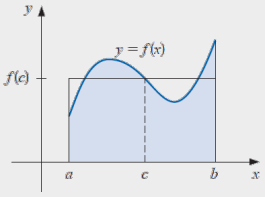
Weighted Mean Value Theorem for Integral

Suppose $f \in C[a, b]$, the Riemann integral of g exists on $[a, b]$, and $g(x)$ does not change sign on $[a, b]$. Then there exists a number c in (a, b) with

$$\int_a^b f(x)g(x)\,dx = f(c)\int_a^b g(x)\,dx.$$

When $g(x) \equiv 1$, Theorem 1.13 is the usual Mean Value Theorem for Integrals. It gives the **average value** of the function f over the interval $[a, b]$ as (See Figure 1.9.)

$$f(c) = \frac{1}{b - a}\int_a^b f(x)\,dx.$$



Initial-Value Problems for Ordinary Differential Equations

Problem

$$y' = f(t,y), \quad a \le t \le b, \quad y(a) = \alpha,$$

Local truncation error definition

The difference method

$$w_0 = \alpha$$
$$w_{i+1} = w_i + h\phi(t_i, w_i), \quad \text{for each } i = 0, 1, \ldots, N-1,$$

has **local truncation error**

$$\tau_{i+1}(h) = \frac{y_{i+1} - (y_i + h\phi(t_i, y_i))}{h} = \frac{y_{i+1} - y_i}{h} - \phi(t_i, y_i),$$

Higher-Order Taylor Method

$$w_0 = \alpha,$$
$$w_{i+1} = w_i + hT^{(n)}(t_i, w_i), \quad \text{for each } i = 0, 1, \ldots, N-1, \tag{5.17}$$

where

$$T^{(n)}(t_i, w_i) = f(t_i, w_i) + \frac{h}{2}f'(t_i, w_i) + \cdots + \frac{h^{n-1}}{n!}f^{(n-1)}(t_i, w_i).$$

Euler's method is Taylor's method of order one.

Euler's method constructs $w_i \approx y(t_i)$, for each $i = 1, 2, \ldots, N$, by deleting the remainder term. Thus Euler's method is

$$w_0 = \alpha,$$
$$w_{i+1} = w_i + hf(t_i, w_i), \quad \text{for each } i = 0, 1, \ldots, N-1. \tag{5.8}$$

Euler's LTE

For example, Euler's method has local truncation error at the ith step

$$\tau_{i+1}(h) = \frac{y_{i+1} - y_i}{h} - f(t_i, y_i), \quad \text{for each } i = 0, 1, \ldots, N-1.$$

This error is a *local error* because it measures the accuracy of the method at a specific step, assuming that the method was exact at the previous step. As such, it depends on the differential equation, the step size, and the particular step in the approximation.

By considering Eq. (5.7) in the previous section, we see that Euler's method has

$$\tau_{i+1}(h) = \frac{h}{2}y''(\xi_i), \quad \text{for some } \xi_i \text{ in } (t_i, t_{i+1}).$$

When $y''(t)$ is known to be bounded by a constant M on $[a, b]$, this implies

$$|\tau_{i+1}(h)| \le \frac{h}{2}M,$$

so the local truncation error in Euler's method is $O(h)$.

Taylor's LTE

If Taylor's method of order n is used to approximate the solution to

$$y'(t) = f(t, y(t)), \quad a \le t \le b, \quad y(a) = \alpha,$$

with step size h and if $y \in C^{n+1}[a, b]$, then the local truncation error is $O(h^n)$. ■

Proof Note that Eq. (5.16) on page 277 can be rewritten

$$y_{i+1} - y_i - hf(t_i, y_i) - \frac{h^2}{2}f'(t_i, y_i) - \cdots - \frac{h^n}{n!}f^{(n-1)}(t_i, y_i) = \frac{h^{n+1}}{(n+1)!}f^{(n)}(\xi_i, y(\xi_i)),$$

for some ξ_i in (t_i, t_{i+1}). So the local truncation error is

$$\tau_{i+1}(h) = \frac{y_{i+1} - y_i}{h} - T^{(n)}(t_i, y_i) = \frac{h^n}{(n+1)!}f^{(n)}(\xi_i, y(\xi_i)),$$

for each $i = 0, 1, \ldots, N-1$. Since $y \in C^{n+1}[a, b]$, we have $y^{(n+1)}(t) = f^{(n)}(t, y(t))$ bounded on $[a, b]$ and $\tau_i(h) = O(h^n)$, for each $i = 1, 2, \ldots, N$. ■ ■ ■

خطای موضعی بسط تیلور تا مرتبه k از $O(h^{k+1})$ است زیرا :

$$R_k(x) = \frac{h^{k+1}}{(k+1)!}y^{(k+1)}(\mu_x) \quad , \quad \mu_x \in [x, x+h]$$

خطای سراسری تقریب که روی بازه $[a, b]$ تعریف میشود برابر است با :

$$\sum_{i=0}^{n-1}\frac{h^{k+1}}{(k+1)!}y^{(k+1)}(\mu_i) = n\frac{h^{k+1}}{(k+1)!}y^{(k+1)}(\mu) = \frac{b-a}{h}\frac{h^{k+1}}{(k+1)!}y^{(k+1)}(\mu_x) \quad , \quad \mu \in [a, b]$$

Heun's Method (Modified Euler's Method)

$$w_0 = \alpha,$$
$$w_{i+1} = w_i + \frac{h}{2}[f(t_i, w_i) + f(t_{i+1}, w_i + hf(t_i, w_i))], \quad \text{for} \quad i = 0, 1, \ldots, N-1.$$

Runge-Kutta Second Order Proof

Writing out the first three terms of Taylor series are

$$y_{i+1} = y_i + \left.\frac{dy}{dx}\right|_{x_i,y_i} h + \frac{1}{2!}\left.\frac{d^2y}{dx^2}\right|_{x_i,y_i} h^2 + O(h^3) \tag{A.5}$$

where

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

Since

$$\frac{dy}{dx} = f(x, y)$$

we can rewrite the Taylor series as

$$y_{i+1} = y_i + f(x_i, y_i)h + \frac{1}{2!}f'(x_i, y_i)h^2 + O(h^3) \tag{A.6}$$

Now

$$f'(x,y) = \frac{\partial f(x,y)}{\partial x} + \frac{\partial f(x,y)}{\partial y}\frac{dy}{dx}. \tag{A.7}$$

Hence

$$y_{i+1} = y_i + f(x_i, y_i)h + \frac{1}{2!}\left(\left.\frac{\partial f}{\partial x}\right|_{x_i,y_i} + \left.\frac{\partial f}{\partial y}\right|_{x_i,y_i} \times \left.\frac{dy}{dx}\right|_{x_i,y_i}\right)h^2 + O(h^3)$$
$$= y_i + f(x_i, y_i)h + \frac{1}{2}\left.\frac{\partial f}{\partial x}\right|_{x_i,y_i} h^2 + \frac{1}{2}\left.\frac{\partial f}{\partial y}\right|_{x_i,y_i} f(x_i, y_i)h^2 + O(h^3) \tag{A.8}$$

Now the term used in the Runge-Kutta 2nd order method for k_2 can be written as a Taylor series of two variables with the first three terms as

$$k_2 = f(x_i + p_i h, y_i + q_{11}k_1 h)$$
$$= f(x_i, y_i) + p_i h\left.\frac{\partial f}{\partial x}\right|_{x_i,y_i} + q_{11}k_1 h\left.\frac{\partial f}{\partial y}\right|_{x_i,y_i} + O(h^2) \tag{A.9}$$

Hence

$$y_{i+1} = y_i + (a_1k_1 + a_2k_2)h$$
$$= y_i + \left(a_1 f(x_i, y_i) + a_2\left\{f(x_i, y_i) + p_i h\left.\frac{\partial f}{\partial x}\right|_{x_i,y_i} + q_{11}k_1 h\left.\frac{\partial f}{\partial y}\right|_{x_i,y_i} + O(h^2)\right\}\right)h$$
$$= y_i + (a_1 + a_2)hf(x_i, y_i) + a_2 p_i h^2\left.\frac{\partial f}{\partial x}\right|_{x_i,y_i} + a_2 q_{11}f(x_i, y_i)h^2\left.\frac{\partial f}{\partial y}\right|_{x_i,y_i} + O(h^3)$$

Second-Order

$$\begin{cases} K_1 = hf(x_n, y_n) \\ K_2 = hf(x_n + \frac{h}{2}, y_n + \frac{K_1}{2}) \\ y_{n+1} = y_n + K_2 \end{cases} \qquad \begin{cases} K_1 = hf(x_n, y_n) \\ K_2 = hf(x_{n+1}, y_n + K_1) \\ y_{n+1} = y_n + \frac{1}{2}(K_1 + K_2) \end{cases}$$

Third and Fourth-Order

$$\begin{cases} K_1 = hf(x_n, y_n) \\ K_2 = hf(x_n + \frac{h}{2}, y_n + \frac{K_1}{2}) \\ K_3 = hf(x_{n+1}, y_n + 2K_2 - K_1) \\ y_{n+1} = y_n + \frac{1}{6}(K_1 + 4K_2 + K_3) \end{cases} \qquad \begin{cases} K_1 = hf(x_n, y_n) \\ K_2 = hf(x_n + \frac{h}{2}, y_n + \frac{K_1}{2}) \\ K_3 = hf(x_n + \frac{h}{2}, y_n + \frac{K_2}{2}) \\ K_4 = hf(x_{n+1}, y_n + K_3) \\ y_{n+1} = y_n + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4) \end{cases}$$

Higher-Order Differential Equations

$$y^{(m)}(t) = f(t, y, y', \ldots, y^{(m-1)}), \quad a \le t \le b,$$

with initial conditions $y(a) = \alpha_1, y'(a) = \alpha_2, \ldots, y^{(m-1)}(a) = \alpha_m$ can be converted into a system of equations in the form (5.45) and (5.46).

Let $u_1(t) = y(t), u_2(t) = y'(t), \ldots,$ and $u_m(t) = y^{(m-1)}(t)$. This produces the first-order system

$$\frac{du_1}{dt} = \frac{dy}{dt} = u_2, \quad \frac{du_2}{dt} = \frac{dy'}{dt} = u_3, \quad \cdots, \quad \frac{du_{m-1}}{dt} = \frac{dy^{(m-2)}}{dt} = u_m,$$

and

$$\frac{du_m}{dt} = \frac{dy^{(m-1)}}{dt} = y^{(m)} = f(t, y, y', \ldots, y^{(m-1)}) = f(t, u_1, u_2, \ldots, u_m),$$

with initial conditions

$$u_1(a) = y(a) = \alpha_1, \quad u_2(a) = y'(a) = \alpha_2, \quad \ldots, \quad u_m(a) = y^{(m-1)}(a) = \alpha_m.$$

Solutions of Equations in One Variable

Convergence Order

Suppose $\{p_n\}_{n=0}^{\infty}$ is a sequence that converges to p, with $p_n \neq p$ for all n. If positive constants λ and α exist with

$$\lim_{n\to\infty} \frac{|p_{n+1}-p|}{|p_n-p|^{\alpha}} = \lambda,$$

then $\{p_n\}_{n=0}^{\infty}$ **converges to p of order α, with asymptotic error constant λ.** ■

An iterative technique of the form $p_n = g(p_{n-1})$ is said to be of *order* α if the sequence $\{p_n\}_{n=0}^{\infty}$ converges to the solution $p = g(p)$ of order α.

In general, a sequence with a high order of convergence converges more rapidly than a sequence with a lower order. The asymptotic constant affects the speed of convergence but not to the extent of the order. Two cases of order are given special attention.

(i) If $\alpha = 1$ (and $\lambda < 1$), the sequence is **linearly convergent**.

(ii) If $\alpha = 2$, the sequence is **quadratically convergent**.

Bisection Technique

Suppose f is a continuous function defined on the interval $[a,b]$, with $f(a)$ and $f(b)$ of opposite sign. The Intermediate Value Theorem implies that a number p exists in (a,b) with $f(p) = 0$. Although the procedure will work when there is more than one root in the interval (a,b), we assume for simplicity that the root in this interval is unique. The method calls for a repeated halving (or bisecting) of subintervals of $[a,b]$ and, at each step, locating the half containing p.

To begin, set $a_1 = a$ and $b_1 = b$, and let p_1 be the midpoint of $[a,b]$; that is,

$$p_1 = a_1 + \frac{b_1 - a_1}{2} = \frac{a_1 + b_1}{2}.$$

- If $f(p_1) = 0$, then $p = p_1$, and we are done.
- If $f(p_1) \neq 0$, then $f(p_1)$ has the same sign as either $f(a_1)$ or $f(b_1)$.
  - If $f(p_1)$ and $f(a_1)$ have the same sign, $p \in (p_1, b_1)$. Set $a_2 = p_1$ and $b_2 = b_1$.
  - If $f(p_1)$ and $f(a_1)$ have opposite signs, $p \in (a_1, p_1)$. Set $a_2 = a_1$ and $b_2 = p_1$.

Convergence Order Proof

Suppose that $f \in C[a,b]$ and $f(a)\cdot f(b) < 0$. The Bisection method generates a sequence $\{p_n\}_{n=1}^{\infty}$ approximating a zero p of f with

$$|p_n - p| \leq \frac{b-a}{2^n}, \quad \text{when} \quad n \geq 1.$$ ■

Proof For each $n \geq 1$, we have

$$b_n - a_n = \frac{1}{2^{n-1}}(b-a) \quad \text{and} \quad p \in (a_n, b_n).$$

Since $p_n = \frac{1}{2}(a_n + b_n)$ for all $n \geq 1$, it follows that

$$|p_n - p| \leq \frac{1}{2}(b_n - a_n) = \frac{b-a}{2^n}.$$ ■ ■ ■

Because

$$|p_n - p| \leq (b-a)\frac{1}{2^n},$$

the sequence $\{p_n\}_{n=1}^{\infty}$ converges to p with rate of convergence $O\left(\frac{1}{2^n}\right)$; that is,

$$p_n = p + O\left(\frac{1}{2^n}\right).$$

Fixed-Point Iteration

Definition

The number p is a **fixed point** for a given function g if $g(p) = p$. ■

Problems

- Given a root-finding problem $f(p) = 0$, we can define functions g with a fixed point at p in a number of ways, for example, as

$$g(x) = x - f(x) \quad \text{or as} \quad g(x) = x + 3f(x).$$

- Conversely, if the function g has a fixed point at p, then the function defined by

$$f(x) = x - g(x)$$

$$g(x) = x \pm af(x)$$

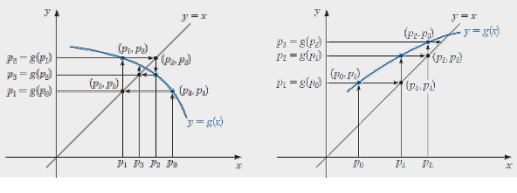
Sufficient Conditions for The Existence and Uniqueness of a Fixed Point

(i) If $g \in C[a,b]$ and $g(x) \in [a,b]$ for all $x \in [a,b]$, then g has at least one fixed point in $[a,b]$.

(ii) If, in addition, $g'(x)$ exists on (a,b) and a positive constant $k < 1$ exists with

$$|g'(x)| \leq k, \quad \text{for all } x \in (a,b),$$

then there is exactly one fixed point in $[a,b]$. (See Figure 2.4.) ■



Proof

(i) If $g(a) = a$ or $g(b) = b$, then g has a fixed point at an endpoint. If not, then $g(a) > a$ and $g(b) < b$. The function $h(x) = g(x) - x$ is continuous on $[a,b]$, with

$$h(a) = g(a) - a > 0 \quad \text{and} \quad h(b) = g(b) - b < 0.$$

The Intermediate Value Theorem implies that there exists $p \in (a,b)$ for which $h(p) = 0$. This number p is a fixed point for g because

$$0 = h(p) = g(p) - p \quad \text{implies that} \quad g(p) = p.$$

(ii) Suppose, in addition, that $|g'(x)| \leq k < 1$ and that p and q are both fixed points in $[a,b]$. If $p \neq q$, then the Mean Value Theorem implies that a number ξ exists between p and q, and hence in $[a,b]$, with

$$\frac{g(p) - g(q)}{p - q} = g'(\xi).$$

Thus

$$|p - q| = |g(p) - g(q)| = |g'(\xi)||p - q| \leq k|p - q| < |p - q|,$$

which is a contradiction. This contradiction must come from the only supposition, $p \neq q$. Hence, $p = q$ and the fixed point in $[a,b]$ is unique. ■ ■ ■

Fixed-Point Theorem

Let $g \in C[a,b]$ be such that $g(x) \in [a,b]$, for all x in $[a,b]$. Suppose, in addition, that g' exists on (a,b) and that a constant $0 < k < 1$ exists with

$$|g'(x)| \leq k, \quad \text{for all } x \in (a,b).$$

Then for any number p_0 in $[a,b]$, the sequence defined by

$$p_n = g(p_{n-1}), \quad n \geq 1,$$

converges to the unique fixed point p in $[a,b]$. ■

Proof Theorem 2.3 implies that a unique point p exists in $[a,b]$ with $g(p) = p$. Since g maps $[a,b]$ into itself, the sequence $\{p_n\}_{n=0}^{\infty}$ is defined for all $n \geq 0$, and $p_n \in [a,b]$ for all n. Using the fact that $|g'(x)| \leq k$ and the Mean Value Theorem 1.8, we have, for each n,

$$|p_n - p| = |g(p_{n-1}) - g(p)| = |g'(\xi_n)||p_{n-1} - p| \leq k|p_{n-1} - p|,$$

where $\xi_n \in (a,b)$. Applying this inequality inductively gives

$$|p_n - p| \leq k|p_{n-1} - p| \leq k^2|p_{n-2} - p| \leq \cdots \leq k^n|p_0 - p|. \tag{2.4}$$

Since $0 < k < 1$, we have $\lim_{n\to\infty} k^n = 0$ and

$$\lim_{n\to\infty} |p_n - p| \leq \lim_{n\to\infty} k^n|p_0 - p| = 0.$$

Hence $\{p_n\}_{n=0}^{\infty}$ converges to p. ■ ■ ■

Bound for Error

If g satisfies the hypotheses of Theorem 2.4, then bounds for the error involved in using p_n to approximate p are given by

$$|p_n - p| \leq k^n \max\{p_0 - a, b - p_0\} \tag{2.5}$$

and

$$|p_n - p| \leq \frac{k^n}{1-k}|p_1 - p_0|, \quad \text{for all} \quad n \geq 1. \tag{2.6}$$

■

Proof Because $p \in [a,b]$, the first bound follows from Inequality (2.4):

$$|p_n - p| \leq k^n|p_0 - p| \leq k^n \max\{p_0 - a, b - p_0\}.$$

For $n \geq 1$, the procedure used in the proof of Theorem 2.4 implies that

$$|p_{n+1} - p_n| = |g(p_n) - g(p_{n-1})| \leq k|p_n - p_{n-1}| \leq \cdots \leq k^n|p_1 - p_0|.$$

Thus for $m > n \geq 1$,

$$\begin{aligned} |p_m - p_n| &= |p_m - p_{m-1} + p_{m-1} - \cdots + p_{n+1} - p_n| \\ &\leq |p_m - p_{m-1}| + |p_{m-1} - p_{m-2}| + \cdots + |p_{n+1} - p_n| \\ &\leq k^{m-1}|p_1 - p_0| + k^{m-2}|p_1 - p_0| + \cdots + k^n|p_1 - p_0| \\ &= k^n|p_1 - p_0|\left(1 + k + k^2 + \cdots + k^{m-n-1}\right). \end{aligned}$$

By Theorem 2.3, $\lim_{m\to\infty} p_m = p$, so

$$|p - p_n| = \lim_{m\to\infty} |p_m - p_n| \leq \lim_{m\to\infty} k^n|p_1 - p_0| \sum_{i=0}^{m-n-1} k^i \leq k^n|p_1 - p_0| \sum_{i=0}^{\infty} k^i.$$

But $\sum_{i=0}^{\infty} k^i$ is a geometric series with ratio k and $0 < k < 1$. This sequence converges to $1/(1-k)$, which gives the second bound:

$$|p - p_n| \leq \frac{k^n}{1-k}|p_1 - p_0|.$$ ■ ■ ■

Convergence Order

- اگر $g'(p) \neq 0$ یعنی p ریشه ساده f باشد، آنگاه مرتبه هم گرایی $\{x_n\}$ به ریشه واقعی، خطی است.
- اگر $g'(p) = g''(p) = \cdots = g^{(k-1)}(p) = 0$ و $g^{(k)}(p) \neq 0$ آنگاه مرتبه هم گرایی $\{x_n\}$ به ریشه معادله، دقیقا برابر k است.

Proof

$$g(x_n) = g(p) + g'(p)(x_n - p) + g''(p)\frac{(x_n-p)^2}{2!} + \cdots + g^{(k)}\frac{(x_n-p)^k}{k!} + O((x_n-p)^{k+1})$$

$$g(x_n) = x_{n+1}, \qquad e_n := x_n - p, \qquad g(p) = p$$

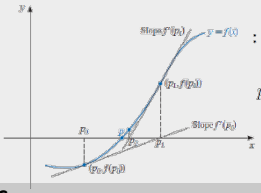
$$\rightarrow g(x_n) = x_{n+1} = p + g^{(k)}\frac{(e_n)^k}{k!} + O(e_n^{k+1})$$

$$\rightarrow e_{n+1} = x_{n+1} - p = g^{(k)}\frac{(e_n)^k}{k!} + O(e_n^{k+1}) \rightarrow \lim_{n\to\infty}\frac{e_{n+1}}{e_n^k} = \frac{g^{(k)}}{k!} .:$$

Newton-Raphson's Method

این روش عددی تکراری، یک روش غیرقطعی برای یافتن ریشه با تکیه بر مشتق تابع است. فرض کنیم f روی $[a,b]$ دو بار مشتق پذیر باشد و $p_0 \in [a,b]$ تقریبی از p باشد به طوری که $f'(p) \neq 0$ آنگاه با فرض کوچک بودن مقدار $|p-p_0|$ داریم :

$$f(p) = f(p_0)+(p-p_0)f'(p_0)+\frac{(p-p_0)^2}{2!}f''(\mu) \implies f(p) \approx f(p_0)+(p-p_0)f'(p_0) \approx 0 \quad (1)$$



این بیان میدارد برای یافتن ریشه با تقریب اولیه p_0 در هر مرحله داریم :

$$p_n = p_{n-1} - \frac{f(p_{n-1})}{f'(p_{n-1})} \quad , \quad n = 1, 2, \ldots \qquad (2)$$

روش معرفی شده، روش نیوتن رفسون نامیده میشود.

Convergence

Let $f \in C^2[a,b]$. If $p \in (a,b)$ is such that $f(p) = 0$ and $f'(p) \neq 0$, then there exists a $\delta > 0$ such that Newton's method generates a sequence $\{p_n\}_{n=1}^{\infty}$ converging to p for any initial approximation $p_0 \in [p-\delta, p+\delta]$. ■

Proof The proof is based on analyzing Newton's method as the functional iteration scheme $p_n = g(p_{n-1})$, for $n \geq 1$, with

$$g(x) = x - \frac{f(x)}{f'(x)}.$$

Let k be in $(0,1)$. We first find an interval $[p-\delta, p+\delta]$ that g maps into itself and for which $|g'(x)| \leq k$, for all $x \in (p-\delta, p+\delta)$.

Since f' is continuous and $f'(p) \neq 0$, part (**a**) of Exercise 29 in Section 1.1 implies that there exists a $\delta_1 > 0$, such that $f'(x) \neq 0$ for $x \in [p-\delta_1, p+\delta_1] \subseteq [a,b]$. Thus g is defined and continuous on $[p-\delta_1, p+\delta_1]$. Also

$$g'(x) = 1 - \frac{f'(x)f'(x) - f(x)f''(x)}{[f'(x)]^2} = \frac{f(x)f''(x)}{[f'(x)]^2},$$

for $x \in [p-\delta_1, p+\delta_1]$, and, since $f \in C^2[a,b]$, we have $g \in C^1[p-\delta_1, p+\delta_1]$.

By assumption, $f(p) = 0$, so

$$g'(p) = \frac{f(p)f''(p)}{[f'(p)]^2} = 0.$$

Since g' is continuous and $0 < k < 1$, part (**b**) of Exercise 29 in Section 1.1 implies that there exists a δ, with $0 < \delta < \delta_1$, and

$$|g'(x)| \leq k, \quad \text{for all } x \in [p-\delta, p+\delta].$$

It remains to show that g maps $[p-\delta, p+\delta]$ into $[p-\delta, p+\delta]$. If $x \in [p-\delta, p+\delta]$, the Mean Value Theorem implies that for some number ξ between x and p, $|g(x)-g(p)| = |g'(\xi)||x-p|$. So

$$|g(x)-p| = |g(x)-g(p)| = |g'(\xi)||x-p| \leq k|x-p| < |x-p|.$$

Since $x \in [p-\delta, p+\delta]$, it follows that $|x-p| < \delta$ and that $|g(x)-p| < \delta$. Hence, g maps $[p-\delta, p+\delta]$ into $[p-\delta, p+\delta]$.

All the hypotheses of the Fixed-Point Theorem 2.4 are now satisfied, so the sequence $\{p_n\}_{n=1}^{\infty}$, defined by

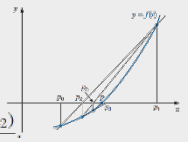
$$p_n = g(p_{n-1}) = p_{n-1} - \frac{f(p_{n-1})}{f'(p_{n-1})}, \quad \text{for } n \geq 1,$$

converges to p for any $p_0 \in [p-\delta, p+\delta]$. ■ ■ ■

Secant Method

To circumvent the problem of the derivative evaluation in Newton's method, we introduce a slight variation. By definition,

$$f'(p_{n-1}) = \lim_{x \to p_{n-1}} \frac{f(x) - f(p_{n-1})}{x - p_{n-1}}.$$



If p_{n-2} is close to p_{n-1}, then

$$f'(p_{n-1}) \approx \frac{f(p_{n-2}) - f(p_{n-1})}{p_{n-2} - p_{n-1}} = \frac{f(p_{n-1}) - f(p_{n-2})}{p_{n-1} - p_{n-2}}.$$

Using this approximation for $f'(p_{n-1})$ in Newton's formula gives

$$p_n = p_{n-1} - \frac{f(p_{n-1})(p_{n-1} - p_{n-2})}{f(p_{n-1}) - f(p_{n-2})}. \qquad (2.12)$$

Aitken's Δ^2 Method

Suppose $\{p_n\}_{n=0}^{\infty}$ is a linearly convergent sequence with limit p. To motivate the construction of a sequence $\{\hat{p}_n\}_{n=0}^{\infty}$ that converges more rapidly to p than does $\{p_n\}_{n=0}^{\infty}$, let us first assume that the signs of $p_n - p$, $p_{n+1} - p$, and $p_{n+2} - p$ agree and that n is sufficiently large that

$$\frac{p_{n+1} - p}{p_n - p} \approx \frac{p_{n+2} - p}{p_{n+1} - p}.$$

Then

$$(p_{n+1} - p)^2 \approx (p_{n+2} - p)(p_n - p),$$

so

$$p_{n+1}^2 - 2p_{n+1}p + p^2 \approx p_{n+2}p_n - (p_n + p_{n+2})p + p^2$$

and

$$(p_{n+2} + p_n - 2p_{n+1})p \approx p_{n+2}p_n - p_{n+1}^2.$$

Solving for p gives

$$p \approx \frac{p_{n+2}p_n - p_{n+1}^2}{p_{n+2} - 2p_{n+1} + p_n}.$$

Adding and subtracting the terms p_n^2 and $2p_np_{n+1}$ in the numerator and grouping terms appropriately gives

$$p \approx \frac{p_np_{n+2} - 2p_np_{n+1} + p_n^2 - p_{n+1}^2 + 2p_np_{n+1} - p_n^2}{p_{n+2} - 2p_{n+1} + p_n}$$

$$= \frac{p_n(p_{n+2} - 2p_{n+1} + p_n) - (p_{n+1}^2 - 2p_np_{n+1} + p_n^2)}{p_{n+2} - 2p_{n+1} + p_n}$$

$$= p_n - \frac{(p_{n+1} - p_n)^2}{p_{n+2} - 2p_{n+1} + p_n}.$$

Aitken's Δ^2 method is based on the assumption that the sequence $\{\hat{p}_n\}_{n=0}^{\infty}$, defined by

$$\hat{p}_n = p_n - \frac{(p_{n+1} - p_n)^2}{p_{n+2} - 2p_{n+1} + p_n}, \qquad (2.14)$$

Horner's Method

Let

$$P(x) = a_nx^n + a_{n-1}x^{n-1} + \cdots + a_1x + a_0.$$

Define $b_n = a_n$ and

$$b_k = a_k + b_{k+1}x_0, \quad \text{for } k = n-1, n-2, \ldots, 1, 0.$$

Then $b_0 = P(x_0)$. Moreover, if

$$Q(x) = b_nx^{n-1} + b_{n-1}x^{n-2} + \cdots + b_2x + b_1,$$

then

$$P(x) = (x - x_0)Q(x) + b_0.$$ ■

Proof By the definition of $Q(x)$,

$$\begin{aligned}(x-x_0)Q(x) + b_0 &= (x-x_0)(b_nx^{n-1} + \cdots + b_2x + b_1) + b_0 \\ &= (b_nx^n + b_{n-1}x^{n-1} + \cdots + b_2x^2 + b_1x) \\ &\quad - (b_nx_0x^{n-1} + \cdots + b_2x_0x + b_1x_0) + b_0 \\ &= b_nx^n + (b_{n-1} - b_nx_0)x^{n-1} + \cdots + (b_1 - b_2x_0)x + (b_0 - b_1x_0).\end{aligned}$$

By the hypothesis, $b_n = a_n$ and $b_k - b_{k+1}x_0 = a_k$, so

$$(x-x_0)Q(x) + b_0 = P(x) \quad \text{and} \quad b_0 = P(x_0).$$ ■ ■ ■

Direct Methods for Solving Linear Systems

Triangular System of Equations

فرض کنید دستگاه معادلات زیر مفروض باشد :

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ 0 & a_{22} & \dots & a_{2n} \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & a_{nn} \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \qquad (۱)$$

به شرط ناصفر بودن مقادیر روی قطر، این دستگاه جواب یکتا دارد :

$$x_n = \frac{b_n}{a_{nn}} \implies \forall i = n-1, n-2, ..., 1 \ : \ x_i = \frac{1}{a_{ii}} \left(b_i - \sum_{j=i+1}^{n} a_{ij} x_j \right) \qquad (۲)$$

محاسبه جواب های این دستگاه بسیار آسان است. بنابراین تلاش میکنیم تا دستگاه معادلات خطی را به فرم بالامثلثی تبدیل کنیم.

Gauss-Jordan Elimination

اول الگوریتم به صورت زیر است :

$$\begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \dots & a_{1n}^{(1)} \\ a_{21}^{(1)} & a_{22}^{(1)} & \dots & a_{2n}^{(1)} \\ \vdots & \vdots & & \vdots \\ a_{n1}^{(1)} & a_{n2}^{(1)} & \dots & a_{nn}^{(1)} \end{bmatrix} \xrightarrow{a_{11}^{(1)} \neq 0} \forall i = 2, 3, ..., n \ : \ Row_i = Row_i - \frac{a_{i1}^{(1)}}{a_{11}^{(1)}} Row_1 \qquad (۳)$$

در الگوریتم بالا، هر سطر با ضریبی از سطری که باید زیر آن صفر شود جمع میشود. این الگوریتم را آنقدر ادامه میدهیم تا ماتریس بالامثلثی حاصل شود. بدین ترتیب برای حل دستگاه معادلات $Ax = b$ الگوریتم حذف گاوس جردن روی ماتریس افزوده $[A|b]$ انجام میشود تا بخش A بالامثلثی شود. در ادامه نیز دستگاه صرفا معادل حل یک دستگاه بالامثلثی مانند قسمت قبل خواهد بود.

هرگاه یکی از ضرایب $a_{kk}^{(k)}$ برابر ۰ باشد، الگوریتم بالا با جا به جا کردن این سطر kام با یک سطر دیگر، $a_{jk}^{(k)}$ ناصفر را به کار میگیرد. هرگاه هیچ سطر دیگری نبود که با جا به جا کردن آن الگوریتم ادامه یابد، الگوریتم متوقف میشود.

Pivoting

در روش های معرفی شده تا به اینجا، برای صفر کردن ضرایب x_j در معادلات $j+1$ ام تا nام، $l_{ij} = -\frac{a_{ij}}{a_{ii}}$ را محاسبه میکنیم و سطرهای $j+1$ ام تا nام را با $l_{ij} Row_j$ جمع میکنیم. هرگاه $|l_{ij}| > 1$ آنگاه خطای محاسبه باعث میشود جواب دستگاه، با مقدار واقعی تفاوت داشته باشد. پس از روش محورگیری [1] برای رفع این مشکل استفاده میکنیم.

- **محورگیری:** هرگاه بخواهیم ضریب x_j در سطرهای $j+1$ام تا nام را صفر کنیم، معادله ای را در سطر j ام قرار میدهیم که ضریب x_j آن نسبت به سایر معادلات، بزرگترین مقدار (قدر مطلق) را داشته باشد. این باعث میشود حداقل خطای محاسبه را داشته باشیم.

The simplest strategy is to select an element in the same column that is below the diagonal and has the largest absolute value; specifically, we determine the smallest $p \geq k$ such that

$$|a_{pk}^{(k)}| = \max_{k \leq i \leq n} |a_{ik}^{(k)}|$$

and perform $(E_k) \leftrightarrow (E_p)$. In this case no interchange of columns is used.

Solve Using Matrix Inverse

هرگاه ماتریس ضرایب دستگاه معادلات خطی وارون پذیر باشد، برای محاسبه پاسخ دستگاه کافی است $x = A^{-1}b$ را محاسبه کنیم. برای یافتن وارون ماتریس A نیز کافی است ماتریس افزوده $[A|I]$ را توسط حذف گاوس جردن به فرم $[I|A^{-1}]$ تبدیل کنیم.

Cramer's Rule

برای حل دستگاه $Ax = b$ که در آن ماتریس ضرایب وارون پذیر است، برای هر x_i داریم :

$$x_i = \frac{1}{|A|} \begin{vmatrix} a_{11} & a_{12} & \dots & b_1 & a_{1,i+1} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & b_2 & a_{2,i+1} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \dots & b_n & a_{n,i+1} & \dots & a_{nn} \end{vmatrix} \qquad (۴)$$

به عبارت دیگر، هر x_i از جایگذاری بردار b در ستون iام ماتریس ضرایب و محاسبه دترمینان حاصل میشود.

Iterative Techniques in Matrix Algebra

Jacobi's Method

The **Jacobi iterative method** is obtained by solving the ith equation in $A\mathbf{x} = \mathbf{b}$ for x_i to obtain (provided $a_{ii} \neq 0$)

$$x_i = \sum_{\substack{j=1 \\ j \neq i}}^{n} \left(-\frac{a_{ij} x_j}{a_{ii}} \right) + \frac{b_i}{a_{ii}}, \qquad \text{for } i = 1, 2, \dots, n.$$

For each $k \geq 1$, generate the components $x_i^{(k)}$ of $\mathbf{x}^{(k)}$ from the components of $\mathbf{x}^{(k-1)}$ by

$$x_i^{(k)} = \frac{1}{a_{ii}} \left[\sum_{\substack{j=1 \\ j \neq i}}^{n} \left(-a_{ij} x_j^{(k-1)} \right) + b_i \right], \qquad \text{for } i = 1, 2, \dots, n. \qquad (7.5)$$

Matrix Form

The Jacobi method can be written in the form $\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}$ by splitting A into its diagonal and off-diagonal parts. To see this, let D be the diagonal matrix whose diagonal entries are those of A, $-L$ be the strictly lower-triangular part of A, and $-U$ be the strictly upper-triangular part of A. With this notation,

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

is split into

$$A = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & a_{nn} \end{bmatrix} - \begin{bmatrix} 0 & \cdots & \cdots & 0 \\ -a_{21} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ -a_{n1} & \cdots & -a_{n,n-1} & 0 \end{bmatrix} - \begin{bmatrix} 0 & -a_{12} & \cdots & -a_{1n} \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & -a_{n-1,n} \\ 0 & \cdots & \cdots & 0 \end{bmatrix}$$

$$= D - L - U.$$

The equation $A\mathbf{x} = \mathbf{b}$, or $(D - L - U)\mathbf{x} = \mathbf{b}$, is then transformed into

$$D\mathbf{x} = (L + U)\mathbf{x} + \mathbf{b},$$

and, if D^{-1} exists, that is, if $a_{ii} \neq 0$ for each i, then

$$\mathbf{x} = D^{-1}(L + U)\mathbf{x} + D^{-1}\mathbf{b}.$$

This results in the matrix form of the Jacobi iterative technique:

$$\mathbf{x}^{(k)} = D^{-1}(L + U)\mathbf{x}^{(k-1)} + D^{-1}\mathbf{b}, \quad k = 1, 2, \dots. \qquad (7.6)$$

Introducing the notation $T_j = D^{-1}(L + U)$ and $\mathbf{c}_j = D^{-1}\mathbf{b}$ gives the Jacobi technique the form

$$\mathbf{x}^{(k)} = T_j \mathbf{x}^{(k-1)} + \mathbf{c}_j. \qquad (7.7)$$

Gauss-Seidel Method

A possible improvement in Algorithm 7.1 can be seen by reconsidering Eq. (7.5). The components of $\mathbf{x}^{(k-1)}$ are used to compute all the components $x_i^{(k)}$ of $\mathbf{x}^{(k)}$. But, for $i > 1$, the components $x_1^{(k)}, \dots, x_{i-1}^{(k)}$ of $\mathbf{x}^{(k)}$ have already been computed and are expected to be better approximations to the actual solutions $x_1, \dots, x_{i-1}$ than are $x_1^{(k-1)}, \dots, x_{i-1}^{(k-1)}$. It seems reasonable, then, to compute $x_i^{(k)}$ using these most recently calculated values. That is, to use

$$x_i^{(k)} = \frac{1}{a_{ii}} \left[-\sum_{j=1}^{i-1} (a_{ij} x_j^{(k)}) - \sum_{j=i+1}^{n} (a_{ij} x_j^{(k-1)}) + b_i \right], \qquad (7.8)$$

for each $i = 1, 2, \dots, n$, instead of Eq. (7.5). This modification is called the **Gauss-Seidel iterative technique** and is illustrated in the following example.

Matrix Form

with the definitions of D, L, and U given previously, we have the Gauss-Seidel method represented by

$$(D - L)\mathbf{x}^{(k)} = U\mathbf{x}^{(k-1)} + \mathbf{b}$$

and

$$\mathbf{x}^{(k)} = (D - L)^{-1}U\mathbf{x}^{(k-1)} + (D - L)^{-1}\mathbf{b}, \quad \text{for each } k = 1, 2, \dots. \qquad (7.9)$$

Letting $T_g = (D - L)^{-1}U$ and $\mathbf{c}_g = (D - L)^{-1}\mathbf{b}$, gives the Gauss-Seidel technique the form

$$\mathbf{x}^{(k)} = T_g \mathbf{x}^{(k-1)} + \mathbf{c}_g. \qquad (7.10)$$

For the lower-triangular matrix $D - L$ to be nonsingular, it is necessary and sufficient that $a_{ii} \neq 0$, for each $i = 1, 2, \dots, n$.

Algorithm 7.2 implements the Gauss-Seidel method.

General Iteration Methods

Convergence of General Iteration Techniques

if $\rho(T) < 1 \rightarrow (I - T)^{-1} = \sum_{j=0}^{\infty} T^j$

If the spectral radius satisfies $\rho(T) < 1$, then $(I - T)^{-1}$ exists, and

$$(I - T)^{-1} = I + T + T^2 + \cdots = \sum_{j=0}^{\infty} T^j. \quad \blacksquare$$

Proof Because $T\mathbf{x} = \lambda\mathbf{x}$ is true precisely when $(I - T)\mathbf{x} = (1 - \lambda)\mathbf{x}$, we have λ as an eigenvalue of T precisely when $1 - \lambda$ is an eigenvalue of $I - T$. But $|\lambda| \leq \rho(T) < 1$, so $\lambda = 1$ is not an eigenvalue of T, and 0 cannot be an eigenvalue of $I - T$. Hence, $(I - T)^{-1}$ exists.

Let $S_m = I + T + T^2 + \cdots + T^m$. Then

$$(I - T)S_m = (1 + T + T^2 + \cdots + T^m) - (T + T^2 + \cdots + T^{m+1}) = I - T^{m+1},$$

and, since T is convergent, Theorem 7.17 implies that

$$\lim_{m \to \infty} (I - T)S_m = \lim_{m \to \infty} (I - T^{m+1}) = I.$$

Thus, $(I - T)^{-1} = \lim_{m \to \infty} S_m = I + T + T^2 + \cdots = \sum_{j=0}^{\infty} T^j$. $\blacksquare \blacksquare \blacksquare$

Convergence Proof iff $\rho(T) < 1$

For any $\mathbf{x}^{(0)} \in \mathbb{R}^n$, the sequence $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ defined by

$$\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}, \quad \text{for each } k \geq 1, \tag{7.11}$$

converges to the unique solution of $\mathbf{x} = T\mathbf{x} + \mathbf{c}$ if and only if $\rho(T) < 1$. ■

Proof First assume that $\rho(T) < 1$. Then,

$$\begin{aligned}\mathbf{x}^{(k)} &= T\mathbf{x}^{(k-1)} + \mathbf{c} \\ &= T(T\mathbf{x}^{(k-2)} + \mathbf{c}) + \mathbf{c} \\ &= T^2\mathbf{x}^{(k-2)} + (T + I)\mathbf{c} \\ &\vdots \\ &= T^k\mathbf{x}^{(0)} + (T^{k-1} + \cdots + T + I)\mathbf{c}.\end{aligned}$$

Because $\rho(T) < 1$, Theorem 7.17 implies that T is convergent, and

$$\lim_{k\to\infty} T^k\mathbf{x}^{(0)} = \mathbf{0}.$$

Lemma 7.18 implies that

$$\lim_{k\to\infty} \mathbf{x}^{(k)} = \lim_{k\to\infty} T^k\mathbf{x}^{(0)} + \left(\sum_{j=0}^{\infty} T^j\right)\mathbf{c} = \mathbf{0} + (I - T)^{-1}\mathbf{c} = (I - T)^{-1}\mathbf{c}.$$

Hence, the sequence $\{\mathbf{x}^{(k)}\}$ converges to the vector $\mathbf{x} \equiv (I - T)^{-1}\mathbf{c}$ and $\mathbf{x} = T\mathbf{x} + \mathbf{c}$.

To prove the converse, we will show that for any $\mathbf{z} \in \mathbb{R}^n$, we have $\lim_{k\to\infty} T^k\mathbf{z} = \mathbf{0}$. By Theorem 7.17, this is equivalent to $\rho(T) < 1$.

Let $\mathbf{z}$ be an arbitrary vector, and $\mathbf{x}$ be the unique solution to $\mathbf{x} = T\mathbf{x} + \mathbf{c}$. Define $\mathbf{x}^{(0)} = \mathbf{x} - \mathbf{z}$, and, for $k \geq 1$, $\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}$. Then $\{\mathbf{x}^{(k)}\}$ converges to $\mathbf{x}$. Also,

$$\mathbf{x} - \mathbf{x}^{(k)} = (T\mathbf{x} + \mathbf{c}) - \left(T\mathbf{x}^{(k-1)} + \mathbf{c}\right) = T\left(\mathbf{x} - \mathbf{x}^{(k-1)}\right),$$

so

$$\mathbf{x} - \mathbf{x}^{(k)} = T\left(\mathbf{x} - \mathbf{x}^{(k-1)}\right) = T^2\left(\mathbf{x} - \mathbf{x}^{(k-2)}\right) = \cdots = T^k\left(\mathbf{x} - \mathbf{x}^{(0)}\right) = T^k\mathbf{z}.$$

Hence $\lim_{k\to\infty} T^k\mathbf{z} = \lim_{k\to\infty} T^k\left(\mathbf{x} - \mathbf{x}^{(0)}\right) = \lim_{k\to\infty}\left(\mathbf{x} - \mathbf{x}^{(k)}\right) = \mathbf{0}$.

But $\mathbf{z} \in \mathbb{R}^n$ was arbitrary, so by Theorem 7.17, T is convergent and $\rho(T) < 1$. ■ ■ ■

If $\|T\| < 1$ *and* $\mathbf{c}$ *is a given vector*

If $\|T\| < 1$ for any natural matrix norm and $\mathbf{c}$ is a given vector, then the sequence $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ defined by $\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}$ converges, for any $\mathbf{x}^{(0)} \in \mathbb{R}^n$, to a vector $\mathbf{x} \in \mathbb{R}^n$, with $\mathbf{x} = T\mathbf{x} + \mathbf{c}$, and the following error bounds hold:

(i) $\|\mathbf{x} - \mathbf{x}^{(k)}\| \leq \|T\|^k\|\mathbf{x}^{(0)} - \mathbf{x}\|$; **(ii)** $\|\mathbf{x} - \mathbf{x}^{(k)}\| \leq \frac{\|T\|^k}{1-\|T\|}\|\mathbf{x}^{(1)} - \mathbf{x}^{(0)}\|$. ■

We have seen that the Jacobi and Gauss-Seidel iterative techniques can be written

$$\mathbf{x}^{(k)} = T_j\mathbf{x}^{(k-1)} + \mathbf{c}_j \quad \text{and} \quad \mathbf{x}^{(k)} = T_g\mathbf{x}^{(k-1)} + \mathbf{c}_g,$$

using the matrices

$$T_j = D^{-1}(L + U) \quad \text{and} \quad T_g = (D - L)^{-1}U.$$

If $\rho(T_j)$ or $\rho(T_g)$ is less than 1, then the corresponding sequence $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ will converge to the solution $\mathbf{x}$ of $A\mathbf{x} = \mathbf{b}$. For example, the Jacobi scheme has

$$\mathbf{x}^{(k)} = D^{-1}(L + U)\mathbf{x}^{(k-1)} + D^{-1}\mathbf{b},$$

and, if $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ converges to $\mathbf{x}$, then

$$\mathbf{x} = D^{-1}(L + U)\mathbf{x} + D^{-1}\mathbf{b}.$$

This implies that

$$D\mathbf{x} = (L + U)\mathbf{x} + \mathbf{b} \quad \text{and} \quad (D - L - U)\mathbf{x} = \mathbf{b}.$$

Since $D - L - U = A$, the solution $\mathbf{x}$ satisfies $A\mathbf{x} = \mathbf{b}$.

We can now give easily verified sufficiency conditions for convergence of the Jacobi and Gauss-Seidel methods. (To prove convergence for the Jacobi scheme see Exercise 14, and for the Gauss-Seidel scheme see [Or2], p. 120.)

Convergence If A is Strictly Diagonally Dominant

If A is strictly diagonally dominant, then for any choice of $\mathbf{x}^{(0)}$, both the Jacobi and Gauss-Seidel methods give sequences $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ that converge to the unique solution of $A\mathbf{x} = \mathbf{b}$. ■

The relationship of the rapidity of convergence to the spectral radius of the iteration matrix T can be seen from Corollary 7.20. The inequalities hold for any natural matrix norm, so it follows from the statement after Theorem 7.15 on page 446 that

$$\|\mathbf{x}^{(k)} - \mathbf{x}\| \approx \rho(T)^k\|\mathbf{x}^{(0)} - \mathbf{x}\|. \tag{7.12}$$

Thus we would like to select the iterative technique with minimal $\rho(T) < 1$ for a particular system $A\mathbf{x} = \mathbf{b}$. No general results exist to tell which of the two techniques, Jacobi or Gauss-Seidel, will be most successful for an arbitrary linear system. In special cases, however, the answer is known, as is demonstrated in the following theorem. The proof of this result can be found in [Y], pp. 120–127.

Stein-Rosenberg

If $a_{ij} \leq 0$, for each $i \neq j$ and $a_{ii} > 0$, for each $i = 1, 2, \ldots, n$, then one and only one of the following statements holds:

(i) $0 \leq \rho(T_g) < \rho(T_j) < 1$; **(ii)** $1 < \rho(T_j) < \rho(T_g)$;
(iii) $\rho(T_j) = \rho(T_g) = 0$; **(iv)** $\rho(T_j) = \rho(T_g) = 1$. ■

For the special case described in Theorem 7.22, we see from part (i) that when one method gives convergence, then both give convergence, and the Gauss-Seidel method converges faster than the Jacobi method. Part (ii) indicates that when one method diverges then both diverge, and the divergence is more pronounced for the Gauss-Seidel method.

Non-Linear System of Equations

فرض کنیم دستگاه معادلات غیرخطی زیر مفروض باشد :

$$\begin{cases} f(x,y) = 0 \\ g(x,y) = 0 \end{cases} \tag{۱۲}$$

فرض کنید (α,β) جواب دستگاه فوق باشند. تخمین (x_s,y_s) از جواب را در نظر بگیریم. داریم :

$$\begin{cases} \alpha = x_s + h_s \\ \beta = y_s + k_s \end{cases}$$

هرگاه (x_s,y_s) تقریب خوبی از جواب باشد، در بسط تیلور f و g ، جملات با ضرایب h_s^2 و k_s^2 حذف میشوند و داریم :

$$\begin{cases} f(x_s,y_s) + h_s\frac{\partial f(x_s,y_s)}{\partial x} + k_s\frac{\partial f(x_s,y_s)}{\partial y} \approx 0 \\ g(x_s,y_s) + h_s\frac{\partial g(x_s,y_s)}{\partial x} + k_s\frac{\partial g(x_s,y_s)}{\partial y} \approx 0 \end{cases} \tag{۱۳}$$

در حالت کلی این دستگاه معادله به صورت زیر نوشته میشود :

$$\begin{bmatrix} \frac{\partial f(x_n,y_n)}{\partial x} & \frac{\partial f(x_n,y_n)}{\partial y} \\ \frac{\partial g(x_n,y_n)}{\partial x} & \frac{\partial g(x_n,y_n)}{\partial y} \end{bmatrix} \times \begin{bmatrix} h_n \\ k_n \end{bmatrix} = \begin{bmatrix} -f(x_n,y_n) \\ -g(x_n,y_n) \end{bmatrix} \tag{۱۴}$$

که در آن، ماتریس ژاکوبی در هر تکرار به عنوان ماتریس ضرایب در نظر گرفته میشود. توجه داریم که در این روش تکراری

$$\begin{cases} x_n = x_{n-1} + h_{n-1} \\ y_n = y_{n-1} + k_{n-1} \\ \alpha = x_n + h_n \\ \beta = y_n + k_n \end{cases} \tag{۱۵}$$

پس هرگاه دترمینان ماتریس ژاکوبی ناصفر باشد، در هر مرحله مقادیر h_{n-1}, k_{n-1} یافت میشوند و تقریب بعدی از جواب، محاسبه میشود. روش فوق، روش نیوتن برای حل دستگاه معادلات غیرخطی نامیده میشود.

To approximate the solution of the nonlinear system $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ given an initial approximation $\mathbf{x}$:

INPUT number n of equations and unknowns; initial approximation $\mathbf{x} = (x_1, \ldots, x_n)^t$, tolerance TOL; maximum number of iterations N.
OUTPUT approximate solution $\mathbf{x} = (x_1, \ldots, x_n)^t$ or a message that the number of iterations was exceeded.

Step 1 Set $k = 1$.

Step 2 While $(k \leq N)$ do Steps 3–7.

Step 3 Calculate $\mathbf{F}(\mathbf{x})$ and $J(\mathbf{x})$, where $J(\mathbf{x})_{ij} = (\partial f_i(\mathbf{x})/\partial x_j)$ for $1 \leq i, j \leq n$.

Step 4 Solve the $n \times n$ linear system $J(\mathbf{x})\mathbf{y} = -\mathbf{F}(\mathbf{x})$.

Step 5 Set $\mathbf{x} = \mathbf{x} + \mathbf{y}$.

Step 6 If $||\mathbf{y}|| < TOL$ then OUTPUT $(\mathbf{x})$;
(The procedure was successful.)
STOP.

Step 7 Set $k = k + 1$.

Step 8 OUTPUT ('Maximum number of iterations exceeded');
(The procedure was unsuccessful.)
STOP. ■

Jacobian Matrix

Define the matrix $J(\mathbf{x})$ by

$$J(\mathbf{x}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(\mathbf{x}) & \frac{\partial f_1}{\partial x_2}(\mathbf{x}) & \cdots & \frac{\partial f_1}{\partial x_n}(\mathbf{x}) \\ \frac{\partial f_2}{\partial x_1}(\mathbf{x}) & \frac{\partial f_2}{\partial x_2}(\mathbf{x}) & \cdots & \frac{\partial f_2}{\partial x_n}(\mathbf{x}) \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1}(\mathbf{x}) & \frac{\partial f_n}{\partial x_2}(\mathbf{x}) & \cdots & \frac{\partial f_n}{\partial x_n}(\mathbf{x}) \end{bmatrix}. \tag{10.8}$$

Overdetermined System of Equations

The method of ordinary least squares can be used to find an approximate solution to overdetermined systems. For the system $\boldsymbol{A\mathbf{x} = \mathbf{b}}$, the least squares formula is obtained from the problem

$$\min_{\mathbf{x}} \|A\mathbf{x} - \mathbf{b}\|,$$

the solution of which can be written with the normal equations,[3]

$$\mathbf{x} = (A^{\mathsf{T}}A)^{-1}A^{\mathsf{T}}\mathbf{b},$$

where $^{\mathsf{T}}$ indicates a matrix transpose, *provided* $(A^{\mathsf{T}}A)^{-1}$ exists (that is, provided A has full column rank). With this formula an approximate solution is found when no exact solution exists, and it gives an exact solution when one does exist. However, to achieve good numerical accuracy, using the QR factorization of A to solve the least squares problem is preferred.[4]

Curve Fitting

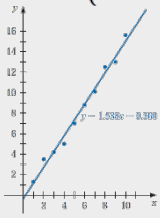
Linear Least Squares

فرض کنیم مجموعه دادگان $\{(x_i, f_i)\}_{i=1}^{N}$ مفروض باشد. به دنبال یافتن بهترین خط به فرم $F(x) = a + bx$ هستیم به طوریکه کم ترین مجموع مربع فاصله با f_i در نقاط x_i را داشته باشد. به عبارت دیگر به دنبال حل کردن مسئله بهینه سازی زیر هستیم :

$$SSE = \sum_{i=1}^{N}(f_i - F(x_i))^2 \implies a^*, b^* = \min_{a,b}\left\{\sum_{i=1}^{N}(f_i - F(x_i))^2\right\} = \min_{a,b}\left\{\sum_{i=1}^{N}(f_i - a - bx_i)^2\right\} \quad (1)$$

ماتریس هسیان بر حسب متغیر های a و b همواره نیمه مثبت معین است پس کافی است شرط صفر بودن مشتقات جزئی را بررسی کنیم :

$$\begin{cases} \frac{\partial SSE}{\partial a} = -2\sum_{i=1}^{N}(f_i - F(x_i)) = 0 \implies aN + b\sum_{i=1}^{N} x_i = \sum_{i=1}^{N} f_i \\ \frac{\partial SSE}{\partial b} = -2\sum_{i=1}^{N}(f_i - F(x_i))(x_i) = 0 \implies a\sum_{i=1}^{N} x_i + b\sum_{i=1}^{N} x_i^2 = \sum_{i=1}^{N} f_i x_i \end{cases} \quad (2)$$



این دستگاه معادلات به صورت زیر هم نوشته میشود :

$$\begin{bmatrix} N & \sum_{i=1}^{N} x_i \\ \sum_{i=1}^{N} x_i & \sum_{i=1}^{N} x_i^2 \end{bmatrix} \times \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N} f_i \\ \sum_{i=1}^{N} f_i x_i \end{bmatrix} \quad (3)$$

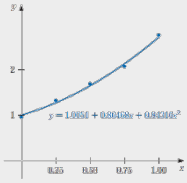
Polynomial Least Squares

این بار فرض کنیم به دنبال برازش کردن داده ها به وسیله چند جمله ای $P(x) = a_0 + a_1 x + ... + a_m x^m$ هستیم. مجددا از معیار کم ترین مجموع مربعات خطا [2] استفاده میکنیم. داریم :

$$SSE = \sum_{i=1}^{N}(f_i - P(x_i))^2 \implies \{a_0^*, ..., a_m^*\} = \min_{a_0^*, ..., a_m^*}\left\{\sum_{i=1}^{N}(f_i - P(x_i))^2\right\} \quad (4)$$

مجددا با صفر قرار دادن مشتقات جزئی میتوان نقطه بهینه و در نتیجه بهترین چندجمله ای را پیدا کرد. برای یک k دلخواه داریم :

$$\frac{\partial SSE}{\partial a_k} = 0 \implies \sum_{i=1}^{N}\sum_{j=0}^{m} a_j x_i^{j+k} = \sum_{i=1}^{k} x_i^k f_i$$



این معادلات به صورت زیر نیز نوشته میشوند :

$$\begin{bmatrix} N & \sum_{i=1}^{N} x_i & \dots & \sum_{i=1}^{N} x_i^m \\ \sum_{i=1}^{N} x_i & \sum_{i=1}^{N} x_i^2 & \dots & \sum_{i=1}^{N} x_i^{m+1} \\ \vdots & \vdots & & \vdots \\ \sum_{i=1}^{N} x_i^m & \sum_{i=1}^{N} x_i^{m+1} & \dots & \sum_{i=1}^{N} x_i^{2m} \end{bmatrix} \times \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N} f_i \\ \sum_{i=1}^{N} x_i f_i \\ \vdots \\ \sum_{i=1}^{N} x_i^m f_i \end{bmatrix} \quad (5)$$

در صورتی که همه نقاط x_i متمایز باشند، ماتریس ضرایب وارون پذیر است و به شکل زیر نوشته میشود :

$$A = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^m \\ 1 & x_2 & x_2^2 & \dots & x_2^m \\ \vdots & 1 & x_N & x_N^2 & \dots & x_N^m \end{bmatrix} \implies A^T \times A = \begin{bmatrix} N & \sum_{i=1}^{N} x_i & \dots & \sum_{i=1}^{N} x_i^m \\ \sum_{i=1}^{N} x_i & \sum_{i=1}^{N} x_i^2 & \dots & \sum_{i=1}^{N} x_i^{m+1} \\ \vdots & \vdots & & \vdots \\ \sum_{i=1}^{N} x_i^m & \sum_{i=1}^{N} x_i^{m+1} & \dots & \sum_{i=1}^{N} x_i^{2m} \end{bmatrix} \quad (6)$$

چون A وارون پذیر است پس $A^T A$ نیز وارون پذیر است و محاسبات آسان تر میشود. اما در مواقعی که A وارون ناپذیر باشد، یعنی دادگان وابسته خطی باشند، انتخاب توابع پایه $\{1, x, x^2, ..., x^m\}$ لزوما مناسب نیست. در این مواقع با تغییر پایه، ماتریس A جدید را میسازیم که ستون های آن، متعامد باشند.

Exponential Least Squares

- فرض کنیم تابع برازش به صورت $y = be^{ax}$ مطلوب باشد. آنگاه داریم :

$$\begin{cases} \frac{\partial SSE}{\partial b} = -2\sum_{i=1}^{N}(f_i - be^{ax_i})(e^{ax_i}) = 0 \\ \frac{\partial SSE}{\partial a} = -2\sum_{i=1}^{N}(f_i - be^{ax_i})(be^{ax_i}) = 0 \end{cases} \quad (15)$$

حل این دستگاه معادلات، معادله خطی به صورت $\ln y = \ln b + ax$ تولید میکند. به عبارت دیگر، نقاط برازش شونده به صورت $(x_i, \ln y_i)$ هستند و $\ln b$ ثابت خط خواهد بود. با یافتن این خط، برازش نمایی اولیه نیز به دست خواهد آمد.

- فرض کنیم تابع برازش به صورت $y = bx^a$ مطلوب باشد. آنگاه داریم :

$$\begin{cases} \frac{\partial SSE}{\partial b} = -2\sum_{i=1}^{N}(f_i - bx_i^a)(x_i^a) = 0 \\ \frac{\partial SSE}{\partial a} = -2\sum_{i=1}^{N}(f_i - bx_i^a)(b(\ln x_i)a) = 0 \end{cases} \quad (16)$$

حل این دستگاه معادلات، معادله خطی به صورت $\ln y = \ln b + a \ln x$ تولید میکند. به عبارت دیگر، نقاط برازش شونده به صورت $(\ln x_i, \ln y_i)$ هستند. با یافتن این خط، برازش نمایی اولیه نیز به دست خواهد آمد.

General Basis Least Squares

فرض کنیم تابع برازش، یک ترکیب خطی از توابع پایه $\{\Phi_1, \Phi_2, ..., \Phi_m\}$ باشد. آنگاه مسئله یافتن بهترین تابع برازش به صورت زیر نوشته میشود :

$$fitting-function : \sum_{i=1}^{m} c_m \Phi_m(x) \implies min_{c_1, c_2, ..., c_m} SSE = min_{c_1, c_2, ..., c_m}\left\{\sum_{i=1}^{N}(f_i - \sum_{j=1}^{m} c_j \Phi_j(x_i))^2\right\} \quad (7)$$

مجددا با قرار دادن همه مشتقات جزئی برابر با صفر داریم :

$$\frac{\partial SSE}{\partial c_k} = -2\sum_{i=1}^{N}(f_i - \sum_{j=1}^{m} c_j \Phi_j(x_i))(\Phi_k(x_i)) = 0 \implies \sum_{i=1}^{N}\sum_{j=1}^{m} c_j \Phi_j(x_i)\Phi_k(x_i) = \sum_{i=1}^{N} f_i \Phi_k(x_i) \quad (8)$$

با انتخاب نوتیشن جدید

$$f = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{bmatrix}, \quad \Phi_k = \begin{bmatrix} \Phi_k(x_1) \\ \Phi_k(x_2) \\ \vdots \\ \Phi_k(x_N) \end{bmatrix} \quad (9)$$

میتوان روابط بالا را به صورت زیر بازنویسی کرد :

$$\sum_{j=1}^{m} c_j < \Phi_j, \Phi_k > = < f_i, \Phi_k >$$

و معادلات به صورت زیر نیز نوشته میشوند :

$$\begin{bmatrix} < \Phi_1, \Phi_1 > & < \Phi_1, \Phi_2 > & \dots & < \Phi_1, \Phi_m > \\ < \Phi_2, \Phi_1 > & < \Phi_2, \Phi_2 > & \dots & < \Phi_2, \Phi_m > \\ \vdots & \vdots & & \vdots \\ < \Phi_m, \Phi_1 > & < \Phi_m, \Phi_2 > & \dots & < \Phi_m, \Phi_m > \end{bmatrix} \times \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix} = \begin{bmatrix} < f, \Phi_1 > \\ < f, \Phi_2 > \\ \vdots \\ < f, \Phi_m > \end{bmatrix} \quad (10)$$

مجددا تعریف میکنیم

$$A = \begin{bmatrix} \Phi_1(x_1) & \Phi_2(x_1) & \dots & \Phi_m(x_1) \\ \Phi_1(x_2) & \Phi_2(x_2) & \dots & \Phi_m(x_2) \\ \vdots & \vdots & & \vdots \\ \Phi_1(x_N) & \Phi_2(x_N) & \dots & \Phi_m(x_N) \end{bmatrix} \quad (11)$$

با این تعریف اگر A وارون پذیر باشد و ستون های آن مستقل خطی باشند، $A^T A$ قطری میشود. این باعث میشود محاسبه ضرایب c_i به مراتب آسان تر شود :

$$\forall i, j : < \Phi_i, \Phi_j > = \begin{cases} ||\Phi_i||_2^2 & i = j \\ 0 & i \neq j \end{cases} \implies \forall i : c_i = \frac{< f, \Phi_i >}{< \Phi_i, \Phi_i >} = \frac{< f, \Phi_i >}{||\Phi_i||_2^2} \quad (12)$$

همانطور که مشاهده کردیم، انتخاب پایه های متعامد، میتواند محاسبات را به مراتب ساده تر کند. در ادامه روش گرام اشمیت را مرور میکنیم که برای متعامدسازی پایه ها به کار میرود.

Gram–Schmidt Process

فرض کنیم مجموعه پایه های مستقل خطی $\{g_1(x), g_2(x), ..., g_m(x)\}$ در اختیار داریم. هدف، تبدیل کردن این پایه ها به مجموعه پایه های متعامد $\{\Phi_1(x), \Phi_2(x), ..., \Phi_m(x)\}$ است. در ادامه الگوریتم گرام اشمیت را معرفی میکنیم :

۱. تعریف کنید $\Phi_1(x) = g_1(x)$

۲. به ازای هر k، مقدار Φ_k را به وسیله سایر پایه های مشخص شده تا به این لحظه، از رابطه زیر محاسبه کنید :

$$\Phi_k(x) = g_k(x) - \sum_{i=1}^{k-1} \frac{< g_k, \Phi_i >}{< \Phi_i, \Phi_i >} \Phi_i(x)$$

الگوریتم گرام اشمیت با دو به دو نرمال کردن توابع پایه، مجموعه توابع پایه متعامد را تولید میکند. همچنین میتوان هر تابع پایه به دست آمده را به صورت $e_k = \frac{\Phi_k(x)}{||\Phi_k||_2^2}$ نرمال سازی کرد.

Change of Basis

فرض کنیم مجموعه توابع $\{g_1(x), g_2(x), ..., g_m(x)\}$ پایه های مستقل خطی برازش اولیه داده ها باشند یعنی

$$F(x) = \sum_{i=1}^{m} c_i g_i(x)$$

و همچنین $\{\Phi_1(x), \Phi_2(x), ..., \Phi_m(x)\}$ مجموعه پایه های متعامد حاصل از اجرای الگوریتم گرام اشمیت باشند یعنی

$$F(x) = \sum_{i=1}^{m} b_i \Phi_i(x)$$

آنگاه برای یافتن ضرایب جدید از روی وزن های قدیمی (وزن های c_i) از تجزیه QR استفاده میکنیم. به عبارت دیگر داریم $A = QR$ یعنی :

$$\begin{bmatrix} g_1(x_1) & g_1(x_2) & \dots & g_1(x_N) \\ g_2(x_1) & g_2(x_2) & \dots & g_2(x_N) \\ \vdots & \vdots & & \vdots \\ g_m(x_1) & g_m(x_2) & \dots & g_m(x_N) \end{bmatrix} = \begin{bmatrix} \Phi_1(x_1) & \Phi_1(x_2) & \dots & \Phi_1(x_N) \\ \Phi_2(x_1) & \Phi_2(x_2) & \dots & \Phi_2(x_N) \\ \vdots & \vdots & & \vdots \\ \Phi_m(x_1) & \Phi_m(x_2) & \dots & \Phi_m(x_N) \end{bmatrix} \times \begin{bmatrix} 1 & \alpha_{12} & \alpha_{13} & \dots & \alpha_{1N} \\ 0 & 1 & \alpha_{23} & \dots & \alpha_{2N} \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix} \quad (13)$$

که در آن داریم

$$\alpha_{ij} = \frac{< g_j, \Phi_i >}{< \Phi_i, \Phi_i >}$$

با این تعاریف، برای یافتن وزن های جدید کافی است از ماتریس بالامثلثی R استفاده کنیم یعنی :

$$R \times C = B \implies R \times \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} \quad (14)$$

Linear Algebra

Vector Norm

A **vector norm** on $\mathbb{R}^n$ is a function, $\|\cdot\|$, from $\mathbb{R}^n$ into $\mathbb{R}$ with the following properties:

(i) $\|\mathbf{x}\| \geq 0$ for all $\mathbf{x} \in \mathbb{R}^n$,

(ii) $\|\mathbf{x}\| = 0$ if and only if $\mathbf{x} = \mathbf{0}$,

(iii) $\|\alpha\mathbf{x}\| = |\alpha|\|\mathbf{x}\|$ for all $\alpha \in \mathbb{R}$ and $\mathbf{x} \in \mathbb{R}^n$,

(iv) $\|\mathbf{x}+\mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$. ■

Examples

The l_2 and l_∞ norms for the vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)^t$ are defined by

$$\|\mathbf{x}\|_2 = \left\{\sum_{i=1}^{n} x_i^2\right\}^{1/2} \quad \text{and} \quad \|\mathbf{x}\|_\infty = \max_{1\leq i\leq n} |x_i|.$$ ■

Cauchy-Schwarz Inequality

For each $\mathbf{x} = (x_1, x_2, \ldots, x_n)^t$ and $\mathbf{y} = (y_1, y_2, \ldots, y_n)^t$ in $\mathbb{R}^n$,

$$\mathbf{x}^t\mathbf{y} = \sum_{i=1}^{n} x_i y_i \leq \left\{\sum_{i=1}^{n} x_i^2\right\}^{1/2} \left\{\sum_{i=1}^{n} y_i^2\right\}^{1/2} = \|\mathbf{x}\|_2 \cdot \|\mathbf{y}\|_2. \tag{7.1}$$

■

Distance

If $\mathbf{x} = (x_1, x_2, \ldots, x_n)^t$ and $\mathbf{y} = (y_1, y_2, \ldots, y_n)^t$ are vectors in $\mathbb{R}^n$, the l_2 and l_∞ distances between $\mathbf{x}$ and $\mathbf{y}$ are defined by

$$\|\mathbf{x}-\mathbf{y}\|_2 = \left\{\sum_{i=1}^{n} (x_i - y_i)^2\right\}^{1/2} \quad \text{and} \quad \|\mathbf{x}-\mathbf{y}\|_\infty = \max_{1\leq i\leq n} |x_i - y_i|.$$ ■

Vector Sequence Converge Definition

A sequence $\{\mathbf{x}^{(k)}\}_{k=1}^{\infty}$ of vectors in $\mathbb{R}^n$ is said to **converge** to $\mathbf{x}$ with respect to the norm $\|\cdot\|$ if, given any $\varepsilon > 0$, there exists an integer $N(\varepsilon)$ such that

$$\|\mathbf{x}^{(k)} - \mathbf{x}\| < \varepsilon, \quad \text{for all } k \geq N(\varepsilon).$$ ■

The sequence of vectors $\{\mathbf{x}^{(k)}\}$ converges to $\mathbf{x}$ in $\mathbb{R}^n$ with respect to the l_∞ norm if and only if $\lim_{k\to\infty} x_i^{(k)} = x_i$, for each $i = 1, 2, \ldots, n$. ■

Proof Suppose $\{\mathbf{x}^{(k)}\}$ converges to $\mathbf{x}$ with respect to the l_∞ norm. Given any $\varepsilon > 0$, there exists an integer $N(\varepsilon)$ such that for all $k \geq N(\varepsilon)$,

$$\max_{i=1,2,\ldots,n} |x_i^{(k)} - x_i| = \|\mathbf{x}^{(k)} - \mathbf{x}\|_\infty < \varepsilon.$$

This result implies that $|x_i^{(k)} - x_i| < \varepsilon$, for each $i = 1, 2, \ldots, n$, so $\lim_{k\to\infty} x_i^{(k)} = x_i$ for each i.

Conversely, suppose that $\lim_{k\to\infty} x_i^{(k)} = x_i$, for every $i = 1, 2, \ldots, n$. For a given $\varepsilon > 0$, let $N_i(\varepsilon)$ for each i represent an integer with the property that

$$|x_i^{(k)} - x_i| < \varepsilon,$$

whenever $k \geq N_i(\varepsilon)$.

Define $N(\varepsilon) = \max_{i=1,2,\ldots,n} N_i(\varepsilon)$. If $k \geq N(\varepsilon)$, then

$$\max_{i=1,2,\ldots,n} |x_i^{(k)} - x_i| = \|\mathbf{x}^{(k)} - \mathbf{x}\|_\infty < \varepsilon.$$

This implies that $\{\mathbf{x}^{(k)}\}$ converges to $\mathbf{x}$ with respect to the l_∞ norm. ■ ■ ■

$\|\mathbf{x}\|_\infty \leq \|\mathbf{x}\|_2 \leq \sqrt{n}\|\mathbf{x}\|_\infty$

Proof Let x_j be a coordinate of $\mathbf{x}$ such that $\|\mathbf{x}\|_\infty = \max_{1\leq i\leq n} |x_i| = |x_j|$. Then

$$\|\mathbf{x}\|_\infty^2 = |x_j|^2 = x_j^2 \leq \sum_{i=1}^{n} x_i^2 = \|\mathbf{x}\|_2^2,$$

and

$$\|\mathbf{x}\|_\infty \leq \|\mathbf{x}\|_2.$$

So

$$\|\mathbf{x}\|_2^2 = \sum_{i=1}^{n} x_i^2 \leq \sum_{i=1}^{n} x_j^2 = nx_j^2 = n\|\mathbf{x}\|_\infty^2,$$

and $\|\mathbf{x}\|_2 \leq \sqrt{n}\|\mathbf{x}\|_\infty$. ■ ■ ■

Matrix Norm

A **matrix norm** on the set of all $n \times n$ matrices is a real-valued function, $\|\cdot\|$, defined on this set, satisfying for all $n \times n$ matrices A and B and all real numbers α:

(i) $\|A\| \geq 0$;

(ii) $\|A\| = 0$, if and only if A is O, the matrix with all 0 entries;

(iii) $\|\alpha A\| = |\alpha|\|A\|$;

(iv) $\|A + B\| \leq \|A\| + \|B\|$;

(v) $\|AB\| \leq \|A\|\|B\|$. ■

Examples

$$\|A\| = \max_{\|\mathbf{x}\|=1} \|A\mathbf{x}\| \qquad \max_{\|\mathbf{x}\|=1} \|A\mathbf{x}\| = \max_{\mathbf{z}\neq\mathbf{0}} \left\|A\left(\frac{\mathbf{z}}{\|\mathbf{z}\|}\right)\right\| = \max_{\mathbf{z}\neq\mathbf{0}} \frac{\|A\mathbf{z}\|}{\|\mathbf{z}\|},$$

$\|A\|_\infty = \max_{\|\mathbf{x}\|_\infty=1} \|A\mathbf{x}\|_\infty$, the l_∞ norm,

$\|A\|_2 = \max_{\|\mathbf{x}\|_2=1} \|A\mathbf{x}\|_2$, the l_2 norm.

$\|A\|_\infty$ For n × n Matrix

If $A = (a_{ij})$ is an $n \times n$ matrix, then

$$\|A\|_\infty = \max_{1\leq i\leq n} \sum_{j=1}^{n} |a_{ij}|.$$ ■

Eigenvalue and Eigenvector

If A is a square matrix, the **characteristic polynomial** of A is defined by

$$p(\lambda) = \det(A - \lambda I).$$ ■

If p is the characteristic polynomial of the matrix A, the zeros of p are **eigenvalues**, or characteristic values, of the matrix A. If λ is an eigenvalue of A and $\mathbf{x} \neq \mathbf{0}$ satisfies $(A - \lambda I)\mathbf{x} = \mathbf{0}$, then $\mathbf{x}$ is an **eigenvector**, or characteristic vector, of A corresponding to the eigenvalue λ. ■

To determine the eigenvalues of a matrix, we can use the fact that

- λ is an eigenvalue of A if and only if $\det(A - \lambda I) = 0$.

Once an eigenvalue λ has been found a corresponding eigenvector $\mathbf{x} \neq \mathbf{0}$ is determined by solving the system

- $(A - \lambda I)\mathbf{x} = \mathbf{0}$.

Spectral Radius

The **spectral radius** $\rho(A)$ of a matrix A is defined by

$$\rho(A) = \max |\lambda|, \quad \text{where } \lambda \text{ is an eigenvalue of } A.$$

(For complex $\lambda = \alpha + \beta i$, we define $|\lambda| = (\alpha^2 + \beta^2)^{1/2}$.) ■

n × n Matrix Spectral Radius Theorems

If A is an $n \times n$ matrix, then

(i) $\|A\|_2 = [\rho(A^tA)]^{1/2}$,

(ii) $\rho(A) \leq \|A\|$, for any natural norm $\|\cdot\|$. ■

Proof The proof of part (i) requires more information concerning eigenvalues than we presently have available. For the details involved in the proof, see [Or2], p. 21.

To prove part (ii), suppose λ is an eigenvalue of A with eigenvector $\mathbf{x}$ and $\|\mathbf{x}\| = 1$. Then $A\mathbf{x} = \lambda\mathbf{x}$ and

$$|\lambda| = |\lambda| \cdot \|\mathbf{x}\| = \|\lambda\mathbf{x}\| = \|A\mathbf{x}\| \leq \|A\|\|\mathbf{x}\| = \|A\|.$$

Thus

$$\rho(A) = \max |\lambda| \leq \|A\|.$$ ■ ■ ■

Positive Definite Matrix

A matrix A is **positive definite** if it is symmetric and if $\mathbf{x}^tA\mathbf{x} > 0$ for every n-dimensional vector $\mathbf{x} \neq \mathbf{0}$. ■

If A is an $n \times n$ positive definite matrix, then

(i) A has an inverse;

(ii) $a_{ii} > 0$, for each $i = 1, 2, \ldots, n$;

(iii) $\max_{1\leq k,j\leq n} |a_{kj}| \leq \max_{1\leq i\leq n} |a_{ii}|$;

(iv) $(a_{ij})^2 < a_{ii}a_{jj}$, for each $i \neq j$. ■

Diagonally Dominant Matrix

The $n \times n$ matrix A is said to be **diagonally dominant** when

$$|a_{ii}| \geq \sum_{\substack{j=1,\\ j\neq i}}^{n} |a_{ij}| \quad \text{holds for each } i = 1, 2, \cdots, n. \tag{6.10}$$

A diagonally dominant matrix is said to be **strictly diagonally dominant** when the inequality in (6.10) is strict for each n, that is, when

$$|a_{ii}| > \sum_{\substack{j=1,\\ j\neq i}}^{n} |a_{ij}| \quad \text{holds for each } i = 1, 2, \cdots, n.$$ ■

Made by: Sahand Es