12. Iterative Methods

HoHee Kim

Gauss-Seidel Method : 선형 대수 방정식을 풀기 위한 반복법

[A]{x}={b} 에서 [A] 가 3x3이고 대각선 entry가 모두 nonzero 일 때

반복 j 번째 의 x_1 값

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3 \end{cases}$$

$$x_1^j = \frac{b_1 - a_{12}x_2^{j-1} - a_{13}x_3^{j-1}}{a_{11}}$$

$$x_2^j = \frac{b_2 - a_{21}x_1^j - a_{23}x_3^{j-1}}{a_{22}}$$

$$x_3^j = \frac{b_3 - a_{31}x_1^j - a_{32}x_2^j}{a_{33}}$$

Ex) Use the Gauss-Seidel method to obtain the solution for

$$\begin{cases} 3x_1 - 0.1x_2 - 0.2x_3 = 7.85 \\ 0.1x_1 + 7x_2 - 0.3x_3 = -19.3 \\ 0.3x_1 - 0.2x_2 + 10x_3 = 71.4 \end{cases}$$

$$x_1 = \frac{7.85 + 0.1x_2 + 0.2x_3}{3}$$

$$x_2 = \frac{-19.3 - 0.1x_1 + 0.3x_3}{7}$$

$$x_3 = \frac{71.4 - 0.3x_1 + 0.2x_2}{10}$$

- $$\begin{aligned} x_1 &= \frac{7.85+0.1(0)+0.2(0)}{3} = 2.616667 \\ x_2 &= \frac{-19.3-0.1(2.616667)+0.3(0)}{7} = -2.794524 \\ x_3 &= \frac{71.4-0.3(2.616667)+0.2(-2.794524)}{10} = 7.005610 \end{aligned}$$

x_2와 x_3 를 0 로 가정

- $$\begin{aligned} x_1 &= \frac{7.85+0.1(-2.794524)+0.2(7.005610)}{3} = 2.990557 \\ x_2 &= \frac{-19.3-0.1(2.990557)+0.3(7.005610)}{7} = -2.499625 \\ x_3 &= \frac{71.4-0.3(2.990557)+0.2(-2.499625)}{10} = 7.000291 \end{aligned}$$

x_1에 대한 근사상대오차 : $\varepsilon_{a,1} = \left| \dfrac{2.990557-2.616667}{2.990557} \right| \times 100\% = 12.5\%$

$\varepsilon_{a,2} = 11.8\%,$

$\varepsilon_{a,3} = 0.076\%$

☞ 반복하면 참 값 $x_1=3$, $x_2=-2.5$, $x_3=7$ 에 수렴

Jacobi iterative method 과 비교: Jacobi 방법은 새로운 값이 얻더라도 바로 사용하지 않고, 다음 반복과정에 사용하기 위해 저장해 둠

First iteration

$$x_1 = (b_1 - a_{12}x_2 - a_{13}x_3)/a_{11} \qquad \boxed{x_1} = (b_1 - a_{12}x_2 - a_{13}x_3)/a_{11}$$

$$x_2 = (b_2 - a_{21}x_1 - a_{23}x_3)/a_{22} \qquad \boxed{x_2} = (b_2 - a_{21}x_1 - a_{23}x_3)/a_{22}$$

$$x_3 = (b_3 - a_{31}x_1 - a_{32}x_2)/a_{33} \qquad \boxed{x_3} = (b_3 - a_{31}x_1 - a_{32}x_2)/a_{33}$$

Second iteration

$$x_1 = (b_1 - a_{12}x_2 - a_{13}x_3)/a_{11} \qquad x_1 = (b_1 - a_{12}x_2 - a_{13}x_3)/a_{11}$$

$$x_2 = (b_2 - a_{21}x_1 - a_{23}x_3)/a_{22} \qquad x_2 = (b_2 - a_{21}x_1 - a_{23}x_3)/a_{22}$$

$$x_3 = (b_3 - a_{31}x_1 - a_{32}x_2)/a_{33} \qquad x_3 = (b_3 - a_{31}x_1 - a_{32}x_2)/a_{33}$$

Gauss-Seidel **Jacobi**

Convergence & Diagonal Dominance

Gauss-Seidel method 은 종종 발산하거나 수렴속도가 느린 경우도 있음

다음 조건을 만족하면 수렴보장

$$|a_{ii}| > \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}|$$

☞ 각각의 방정식에서 대각선 요소의 계수는 대각선 요소를 제외한 다른 요소들의 절대값의 합보다 커야 함

☞ 이 조건을 만족하는 system을 diagonally dominant 라고 함

(수렴을 위한 필요조건은 아니므로 이 조건을 만족하지 않아도 수렴하는 경우 있음)

simple fixed point iteration x=g(x) 에서 수렴조건은 |g'(x)|<1 처럼,

- 2개의 nonlinear equations u(x,y) 와 v(x,y) 의 충분 수렴조건은

$$\left|\frac{\partial u}{\partial x}\right| + \left|\frac{\partial u}{\partial y}\right| < 1 \quad and \quad \left|\frac{\partial v}{\partial x}\right| + \left|\frac{\partial v}{\partial y}\right| < 1$$

→ 이 기준을 linear equation 에도 적용가능

- 2개의 linear equations 경우, Gauss-Seidel 알고리즘은 다음처럼 표현

$$\begin{cases} a_{11}x_1 + a_{12}x_2 = b_1 \\ a_{21}x_1 + a_{22}x_2 = b_2 \end{cases} \quad \rightarrow \quad x_1 = \frac{b_1 - a_{12}x_2}{a_{11}} = u(x_1, x_2), \quad x_2 = \frac{b_2 - a_{21}x_1}{a_{22}} = v(x_1, x_2)$$

$$\left[\begin{array}{ll} \dfrac{\partial u}{\partial x_1} = 0, & \dfrac{\partial u}{\partial x_2} = -\dfrac{a_{12}}{a_{11}} \\ \dfrac{\partial v}{\partial x_1} = -\dfrac{a_{21}}{a_{22}}, & \dfrac{\partial v}{\partial x_2} = 0 \end{array}\right.$$

위의 수렴조건에 대입

$$\Rightarrow \left|\frac{a_{12}}{a_{11}}\right| < 1, \qquad \left|\frac{a_{21}}{a_{22}}\right| < 1 \quad \Rightarrow |a_{12}| < |a_{11}|, \qquad |a_{21}| < |a_{22}|$$

Relaxation : Gauss-Seidel method 의 수렴을 향상시키려고
Gauss-Seidel method 에 약간의 수정을 한 방법

$$\boxed{x_i^{\text{new}} = \lambda x_i^{\text{new}} + (1-\lambda)x_i^{\text{old}}}$$

이전과 현재 반복과정에서 얻은 결과값의 가중평균값

λ : weighting factor (between 0 and 2)

- λ=1 : no relaxation , 수정 안됨
- 0<λ<1 : underrelaxation, 수렴 안 되는 시스템을 수렴되게 하거나 진동을 약화시켜 수렴을 서두르게 함
- 1<λ≤2 : overrelaxation, 현재 값에 더 가중치를 두어 참값에 가까이 가지만 속도는 느림

successive overrelaxation (SOR) : 이미 수렴되는 시스템의 수렴을 가속

☞ λ 은 경험에 의해 결정

Ex) Solve the system Gauss-Seidel using overrelaxation (λ=1.2, $\varepsilon_s = 10\%$)

$$\begin{cases} -3x_1 + 12x_2 = 9 \\ 10x_1 - 2x_2 = 8 \end{cases} \quad \begin{matrix} 10x_1 - 2x_2 = 8 \\ -3x_1 + 12x_2 = 9 \end{matrix} \quad \begin{cases} x_1 = \dfrac{8+2x_2}{10} = 0.8 + 0.2x_2 \\ x_2 = \dfrac{9+3x_1}{12} = 0.75 + 0.25x_1 \end{cases}$$

diagonally dominant

초기값 x_1=x_2=0 로 설정

- $$\begin{aligned} x_1 &= 0.8 + 0.2(0) = 0.8 \\ x_{1,r} &= 1.2(0.8) - 0.2(0) = 0.96 \\ x_2 &= 0.75 + 0.25(0.96) = 0.99 \\ x_{2,r} &= 1.2(0.99) - 0.2(0) = 1.188 \end{aligned}$$

- $$\begin{aligned} x_1 &= 0.8 + 0.2(1.188) = 1.0376 \\ x_{1,r} &= 1.2(1.0376) - 0.2(0.96) = 1.05312 \\ x_2 &= 0.75 + 0.25(1.05312) = 1.01328 \\ x_{2,r} &= 1.2(1.01328) - 0.2(1.188) = 0.978336 \end{aligned}$$

$$\varepsilon_{a,1} = \left| \frac{1.05312 - 0.96}{1.05312} \right| \times 100\% = 8.84\%$$

$$\varepsilon_{a,2} = \left| \frac{0.978336 - 1.188}{0.978336} \right| \times 100\% = 21.43\%$$

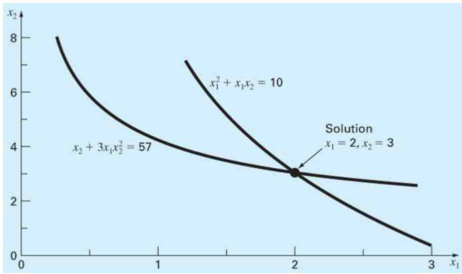
☞허용오차 이내로 반복 → x_1=0.984177, x_2=0.999586 으로 x_1=x_2=1 에 근접

Nonlinear Systems : 반복법을 이용하여 solution 을 구함

① Successive substitution

Ex) Use successive substitution

$$\begin{cases} x_1^2 + x_1 x_2 = 10 \\ x_2 + 3x_1 x_2^2 = 57 \end{cases}$$



X $\left[\begin{aligned} x_1 &= \frac{10 - x_1^2}{x_2} \\ x_2 &= 57 - 3x_1x_2^2 \end{aligned}\right.$ 발산

O $\left[\begin{aligned} x_1 &= \sqrt{10 - x_1x_2} \\ x_2 &= \sqrt{\frac{57 - x_2}{3x_1}} \end{aligned}\right.$ 수렴

초기 추정값 x_1=1.5, x_2=3.5 설정

- $\left[\begin{aligned} x_1 &= \sqrt{10 - 1.5(3.5)} = 2.17945 \\ x_2 &= \sqrt{\frac{57 - 3.5}{3(2.17945)}} = 2.86051 \end{aligned}\right.$
- $\left[\begin{aligned} x_1 &= \sqrt{10 - 2.17945(2.86051)} = 1.94053 \\ x_2 &= \sqrt{\frac{57 - 2.86051}{3(1.94053)}} = 3.04955 \end{aligned}\right.$

☞반복하면 참 값 x_1=2, x_2=3 에 근접하나, 방정식형태나 초기값에 따라 발산

② Newton-Raphson method

- $f(x) = 0$ 의 근은 first-order Taylor series 에 의해

$$\underbrace{f(x_{i+1})}_{0} = f(x_i) + (x_{i+1} - x_i) f'(x_i) \longrightarrow x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

- $f_1(x_1, x_2) = 0, \ f_2(x_1, x_2) = 0$ 의 근은 two variables 의 Taylor series 에 의해

$$\underbrace{f_{1,i+1}}_{0} = f_{1,i} + (x_{1,i+1} - x_{1,i}) \frac{\partial f_{1,i}}{\partial x_1} + (x_{2,i+1} - x_{2,i}) \frac{\partial f_{1,i}}{\partial x_2}$$

$$\underbrace{f_{2,i+1}}_{0} = f_{2,i} + (x_{1,i+1} - x_{1,i}) \frac{\partial f_{2,i}}{\partial x_1} + (x_{2,i+1} - x_{2,i}) \frac{\partial f_{2,i}}{\partial x_2}$$

$$\left[\begin{aligned} \frac{\partial f_{1,i}}{\partial x_1} \underline{x_{1,i+1}} + \frac{\partial f_{1,i}}{\partial x_2} \underline{x_{2,i+1}} &= -f_{1,i} + x_{1,i} \frac{\partial f_{1,i}}{\partial x_1} + x_{2,i} \frac{\partial f_{1,i}}{\partial x_2} \\ \frac{\partial f_{2,i}}{\partial x_1} \underline{x_{1,i+1}} + \frac{\partial f_{2,i}}{\partial x_2} \underline{x_{2,i+1}} &= -f_{2,i} + x_{1,i} \frac{\partial f_{2,i}}{\partial x_1} + x_{2,i} \frac{\partial f_{2,i}}{\partial x_2} \end{aligned}\right. \rightarrow \left[\begin{aligned} x_{1,i+1} &= x_{1,i} - \frac{f_{1,i} \frac{\partial f_{2,i}}{\partial x_2} - f_{2,i} \frac{\partial f_{1,i}}{\partial x_2}}{\frac{\partial f_{1,i}}{\partial x_1} \frac{\partial f_{2,i}}{\partial x_2} - \frac{\partial f_{1,i}}{\partial x_2} \frac{\partial f_{2,i}}{\partial x_1}} \\ x_{2,i+1} &= x_{2,i} - \frac{f_{2,i} \frac{\partial f_{1,i}}{\partial x_1} - f_{1,i} \frac{\partial f_{2,i}}{\partial x_1}}{\frac{\partial f_{1,i}}{\partial x_1} \frac{\partial f_{2,i}}{\partial x_2} - \frac{\partial f_{1,i}}{\partial x_2} \frac{\partial f_{2,i}}{\partial x_1}} \end{aligned}\right.$$

unknown

- $\begin{cases} f_1(x_1,x_2,\dots,x_n)=0 \\ f_2(x_1,x_2,\dots,x_n)=0 \\ \vdots \\ f_n(x_1,x_2,\dots,x_n)=0 \end{cases}$ 의 근은 multiple variables 의 Taylor series 에 의해

n 개의 equation 중에서 k 번째 equation

$$\frac{\partial f_{k,i}}{\partial x_1}x_{1,i+1}+\frac{\partial f_{k,i}}{\partial x_2}x_{2,i+1}+\dots+\frac{\partial f_{k,i}}{\partial x_n}x_{n,i+1}=-f_{k,i}+x_{1,i}\frac{\partial f_{k,i}}{\partial x_1}+x_{2,i}\frac{\partial f_{k,i}}{\partial x_2}+\dots+x_{n,i}\frac{\partial f_{k,i}}{\partial x_n}$$

$\rightarrow$ $[J]\{x_{i+1}\}=-\{f\}+[J]\{x_i\}$

$$\lfloor x_{i+1}\rfloor^T=\lfloor x_{1,i+1}\quad x_{2,i+1}\quad\cdots\quad x_{n,i+1}\rfloor$$

$$\lfloor f\rfloor^T=\lfloor f_{1,i}\quad f_{2,i}\quad\cdots\quad f_{n,i}\rfloor$$

$$\lfloor x_i\rfloor^T=\lfloor x_{1,i}\quad x_{2,i}\quad\cdots\quad x_{n,i}\rfloor$$

$$[J]=\begin{bmatrix}\frac{\partial f_{1,i}}{\partial x_1} & \frac{\partial f_{1,i}}{\partial x_2} & \cdots & \frac{\partial f_{1,i}}{\partial x_n}\\ \frac{\partial f_{2,i}}{\partial x_1} & \frac{\partial f_{2,i}}{\partial x_2} & \cdots & \frac{\partial f_{2,i}}{\partial x_n}\\ \vdots & \vdots & & \vdots\\ \frac{\partial f_{n,i}}{\partial x_1} & \frac{\partial f_{n,i}}{\partial x_2} & \cdots & \frac{\partial f_{n,i}}{\partial x_n}\end{bmatrix}$$

Jacobian matrix

$\Rightarrow$ $\boxed{\{x_{i+1}\}=\{x_i\}-[J]^{-1}\{f\}}$

Ex) Use the multiple-equation Newton-Raphson method to determine roots.

$$\begin{cases}x_1^2+x_1x_2=10\\ x_2+3x_1x_2^2=57\end{cases}\longrightarrow\begin{cases}x_1^2+x_1x_2-10=0 \longrightarrow f_1(x_1,x_2)=0\\ x_2+3x_1x_2^2-57=0 \longrightarrow f_2(x_1,x_2)=0\end{cases}$$

$$[J]=\begin{bmatrix}\frac{\partial f_{1,i}}{\partial x_1} & \frac{\partial f_{1,i}}{\partial x_2}\\ \frac{\partial f_{2,i}}{\partial x_1} & \frac{\partial f_{2,i}}{\partial x_2}\end{bmatrix}$$

$$\frac{\partial f_{1,i}}{\partial x_1}=2x_1+x_2,\quad \frac{\partial f_{1,i}}{\partial x_2}=x_1$$

$$\frac{\partial f_{2,i}}{\partial x_1}=3x_2^2\quad,\quad \frac{\partial f_{2,i}}{\partial x_2}=1+6x_1x_2$$

$$[J]^{-1}=\frac{1}{|J|}\begin{bmatrix}\frac{\partial f_{2,i}}{\partial x_2} & -\frac{\partial f_{1,i}}{\partial x_2}\\ -\frac{\partial f_{2,i}}{\partial x_1} & \frac{\partial f_{1,i}}{\partial x_1}\end{bmatrix}$$

- 초기 추정값 x_1=1.5, x_2=3.5 으로 설정

$\rightarrow$ $\frac{\partial f_{1,0}}{\partial x_1}=6.5,\quad \frac{\partial f_{1,0}}{\partial x_2}=1.5,\quad \frac{\partial f_{2,0}}{\partial x_1}=36.75,\quad \frac{\partial f_{2,0}}{\partial x_2}=32.5$ $\rightarrow$ $[J]^{-1}=\frac{1}{|J|}\begin{bmatrix}32.5 & -1.5\\ -36.75 & 6.5\end{bmatrix}$

$\rightarrow$ $|J|=6.5(32.5)-1.5(36.75)=156.125$

$\rightarrow$ $f_{1,0}=(1.5)^2+1.5(3.5)-10=-2.5,\qquad f_{2,0}=3.5+3(1.5)(3.5)^2-57=1.625$

$$x_1=1.5-\frac{-2.5(32.5)-1.625(1.5)}{156.125}=2.03603,\quad x_2=3.5-\frac{1.625(6.5)-(-2.5)(36.75)}{156.125}=2.84388$$

☞ 새로운 x_1 , x_2 로 계산 반복하면 참 값 x_1=2, x_2=3 에 근접

```
>> x = [1.5;3.5];
>> J = [ 2*x(1)+x(2)  x(1) ; 3*x(2)^2  1+6*x(1)*x(2)]
 J =
     6.5000    1.5000
    36.7500   32.5000
>> f = [ x(1)^2+x(1)*x(2)-10 ; x(2)+3*x(1)*x(2)^2-57]
 f =
    -2.5000
     1.6250
>> x = x-J\f
 x =
     2.0360
     2.8439
```



```
function f = fun(x)
  f=[ x(1)^2+x(1)*x(2)-10 ; x(2)+3*x(1)*x(2)^2-57 ];
end
```

fun.m

```
>> options = optimset('display','iter')
>> [x, fx]= fsolve( @fun, [1.5;3.5], options )
```

비선형 시스템의 해를 찾는 built-in 함수

Iteration	Func-count	f(x)	Norm of step	First-order optimality	Trust-region radius
0	3	8.89063		49.1	1
1	4	8.89063	0.847247	49.1	1
…					
6	19	2.01948e-28	4.34443e-08	5.26e-13	0.53

```
x = 2.0000
    3.0000
fx = 1.0e-13*
     0
     0.1421
```