Lee Jung Bok

전자계산기에 의한 다원연립 일차방정식의 해법에 관한 연구

A Study on the Simultaneous Linear Equations by Computer

이 정 복*

There are several methods which have been presented up to now in solving the simultaneous linear equations by computer.

They are Gaussian Elimination Method, Gauss-Jordan Method, Inverse matrix Method and Gauss-Seidel iterative Method.

This paper is not only discussed in their mechanisms compared with their algorithms, depicted flow charts, but also calculated the numbers of arithmetic operations and comparisons in order to criticize thier availability.

Inverse Matrix Method among them is founded out the smallest in the number of arithmetic operation, but is not the shortest operation time.

This paper also indicates the many problems in using these methods and propose the new method which is able to applicate to even small or middle size computers.

1. 서 론

과학기술분야 및 경영분야에서의 문제해결에 연립일차방정식을 풀어야 할 경우가 허다하게 발생된다.

일반적으로 전자계산기에 의한 연립일차방정식의 수치해법은 크게 두가지 형으로 직접법과 간접법으로 분류한다. 직접법은 이론적으로는 ROUND-OFF 또는 그 외의 오차가 없기 때문에 유한번의 기본적인 연산으로 정확한 해를 얻는다. 그러나 실제로는 유한인 언어의 길이로 작용하기 때문에 직접법으로 인한 정확한 해를 유도해 낼 수는 없다. 사실 오차는 ROUND-OFF, 불안정, 유한숫자의 상실 등으로 극히 빈약하거나 쓸모없는 결과가 나타난다. 직접법에 하나로 GAUSSIAN ELIMIMATED MEATHOD가 있다.

반복법은 초기 근사치로 시작하여 적절한 계산법을 써서 더 근사치를 얻는 방법이다. 반복법을 쓰면 설혹 반복수가 수렴할지라도 근사해밖에 구하지 못한다.

본 연구는 COMPUTOR로 흔히 사용하는 일반적인 4가지 방법인, 즉 GAUSSIAN ELIMINATION METHOD, GAUSS-JORDAN METHOD, INVERSE MATRIX METHOD, GAUSS-SEIDEL ITERATIVE METHOD에 대하여 검토하고 물리적 SYSTEM을 기술하여 각 방법들의 해를 구하는 기법과 FLOW-CHART를 제공하였고 연산속도를 비교검토하였다.

2. 본 론

2.1 CRAMER의 공식에 의한 해법

대수학에서 연립일차방정식의 해를 구하는 일반적 방법으로 두가지를 들 수 있다. 방정식의 조합으로부터 미지수를 소거하는 방법과 행렬식을 이용(CRAMER의 공식)해서 구하는 방법이 있다. 3원연립일차방정식을

* 명지실업전문대학 전자계산학과 조교수

구할 때는 CRAMER의 방법이 소거법보다 유리하지만 방정식에 수가 많은 경우에는 소행렬식으로 전개해야 하는 행렬식의 방법은 실제적으로는 사용 불가능하다. N원연립일차방정식을 CRAMER의 공식에 의하여 풀 때, 행렬식의 계산을 소행렬식으로 전개하면 (N−1) (N+1)! 회의 곱셈이 필요하다. 예를들면 10원연립일차방정식의 해를 구할 때는 359,251,200회의 곱셈이 필요하다. COMPUTER로 1초간에 2,600회의 곱셈을 행한다고 할 때 해를 얻으려면 약 38시간이 걸린다는 결론이 된다. N = 26일 때는 $10^{18)}$ 년이 걸린다. 일반적으로는 미지수가 100 내지 1,000개 되는 연립방정식일 경우가 많다. 그러므로 행렬식을 소행렬식으로 전개하여 계산하는 방법은 실지적으로는 사용하지 않는다.

N개의 미지수를 포함하는 N개의 방정식에 대한 일반형은 다음과 같다.

$$\left.\begin{array}{l} a_{11}x_1 + a_{12}x_2 + \cdots\cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots\cdots + a_{2n}x_n = b_2 \\ \vdots \qquad \vdots \qquad\qquad \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots\cdots + a_{nn}x_n = b_n \end{array}\right\} \cdots\cdots (1-1)$$

또 일반형 (1−1)식에서

$$A = \begin{bmatrix} a_{11}\, a_{12} \cdots\cdots a_{1n} \\ a_{21}\, a_{22} \cdots\cdots a_{2n} \\ \vdots \qquad\qquad \vdots \\ a_{n1} \cdots\cdots a_{nn} \end{bmatrix} \cdots\cdots (1-2)$$

은 계수행렬이라 하고 정방행렬 (N * N행렬)이며 역행렬을 갖는다. 또 계수행렬에 정수항을 부가하여

$$C = \begin{bmatrix} a_{11}\, a_{12} \cdots\cdots a_{1n}\, b_1 \\ a_{21} \quad \cdots\cdots a_{2n}\, b_2 \\ \vdots \qquad\qquad \vdots \\ a_{nn} \quad \cdots\cdots a_{nn}\, b_n \end{bmatrix} \cdots\cdots (1-3)$$

을 확대행렬(AUGMENTED MATRIX)라고 한다.

일반형 (1−1)식을 행렬로 표기하면,

$$AX = B \cdots\cdots (1-4)$$

여기서 행렬A가 역함수를 가지면

$$DET(A) = 0 \cdots\cdots (1-5)$$

이다. (1−5)식은 (1−4)식의 X가 해를 갖기 위한 필요충분조건이다. 즉

$$A = \begin{pmatrix} a_{11} \cdots\cdots a_{1n} \\ \vdots \qquad \vdots \\ a_{n1} \cdots\cdots a_{nn} \end{pmatrix}, \qquad X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \qquad B = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

일 때

$$AX = B$$

를 만족시키려면

$$J = 1, 2, \cdots\cdots, N$$

에 대하여

$$DET\ (A^j) = X_j\ DET\ (A)$$

$$X_j = DET\ (A^j) / DET\ (A) \cdots\cdots (1-6)$$

여기서 A^j 은 행렬 A의 J열을 B로 대치해서 얻은 행렬이다. 이 (1−6)식이 바로 연립 1 차방정식

AX = B의 해인 X의 원소를 구하는 CRAMER의 법칙이다.

2.2 GAUSSIAN ELIMINATION METHOD

대수학에서 연립 1차방정식의 해법으로 제일 많이 쓰이는 방법은 방정식의 조합에서 미지수를 소거하는 방법이다. 소거과정의 한 방법으로 GAUSS의 소거법이 있다.

GAUSS의 방법을 사용할 때 N개의 미지수를 포함하는 N개의 방정식을 이와 동치인 삼각형방정식으로 변환하여 이것을 차례로 설명하는 "후방대입"이라고 하는 간단한 수단으로 쉽게 해를 얻는다. 일반형 (1−1)식과 동치인 삼각형방정식은 다음과 같다.

$$\left.\begin{array}{l} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots\cdots + a_{1n}x_n = C_1 \\ \qquad a'_{22}x_2 + a'_{23}x_3 + \cdots\cdots + a'_{2n}x_n = C'_2 \\ \qquad\qquad a''_{23}x_3 + \cdots\cdots + a''_{3n}x_n = C''_3 \\ \qquad\qquad \cdots\cdots\cdots\cdots\cdots\cdots \\ \qquad\qquad a^{n-2}_{n-1\ n-1}x_{n-1} + a^{n-2}_{n-1\ n}x_n = C^{n-2}_{n-1} \\ \qquad\qquad\qquad a^{n-1}_{nn}x_n = C^{n-1}_n \end{array}\right\} \cdots\cdots (2-1)$$

로 변환된다. 이 변환과정은 다음과 같다.

1) 첫번째 단계

$a_{11} \neq 0$이라면

$$m_i = \frac{a_{i1}}{a_{11}}, \quad i = 1, 2, \cdots\cdots, n$$

먼저 m_i를 구하면 첫번째 방정식에서 m_i 배하여 I번째 방정식을 각각 빼주면 x_1을 둘째, 세째, …… N번째 방정식에서 소거할 수 있으며 a_{ij}와 b_i는 다음과 같이 변한다.

$a'_{ij} = a_{ij} - m_i a_{1j}$, $\quad i = 2, \cdots\cdots, n$

$b'_i = b_i - m_i b_1$, $\quad j = 1, \cdots\cdots, n$

$\therefore a_{i1} = 0$, $\quad i = 2, 3, \cdots\cdots, n$

이것은

$$\left.\begin{array}{l} a_{11}x_1 + a_{12}x_2 + \cdots\cdots + a_{1j}x_j + \cdots\cdots + a_{1m}x_m = b_1 \\ \qquad a'_{22}x_2 + \cdots\cdots + a'_{2j}x_j + \cdots\cdots + a'_{2n}x_n = b'_2 \\ \qquad \cdots\cdots\cdots\cdots\cdots\cdots \\ \qquad a'_{i2}x_2 + \cdots\cdots + a'_{ij}x_j + \cdots\cdots + a'_{in}x_n = b_i \\ \qquad \cdots\cdots\cdots\cdots\cdots\cdots \\ \qquad a'_{n2}x_2 + \cdots\cdots + a'_{nj}x_j + \cdots\cdots + a'_{nn}x_n = b_n \end{array}\right\} \cdots\cdots (2-2)$$

와 같은 형태의 새로운 방정식이 된다.

식 (2−2)의 각 방정식은 식 (2−1)의 두 방정식에서 나왔고 반대로 식 (2−2)에서 식 (2−1)이 변환될 수 있으므로 식 (2−1)과 식 (2−2)는 같은 해를 갖는다.

따라서 a'_{ij}, b'_i를 a_{ij}, b_i로 RENAME하면 (2−3)식을 얻는다.

$$\left.\begin{array}{l} a_{11}x_1 + a_{12}x_2 + \cdots\cdots + a_{1j}x_j + \cdots\cdots + a_{1n}x_n = b_1 \\ \qquad a_{22}x_2 + \cdots\cdots + a_{2j}x_j + \cdots\cdots + a_{2n}x_n = b_2 \\ \qquad \cdots\cdots\cdots\cdots\cdots\cdots \\ \qquad a_{i2}x_2 + \cdots\cdots + a_{ij}x_j + \cdots\cdots + a_{in}x_n = b_i \\ \qquad \cdots\cdots\cdots\cdots\cdots\cdots \\ \qquad a_{n2}x_2 + \cdots\cdots + a_{nj}x_j + \cdots\cdots + a_{nn}x_n = b_n \end{array}\right\} \cdots\cdots (2-3)$$

2) 두번째 단계

$a_{22} \neq 0$이라면

$m_2 = \frac{a_{i2}}{a_{22}}$, $i = 3, 4, \cdots\cdots, n$.

두번째 방정식에서 m_i 배하여 i 번째 방정식을 각각 빼준다면 x_2 를 세째, ……, N번째 방정식에서 소거할 수 있으며 다음과 같이 정의한다.

$a'_{ij} = a_i - m_i a_{2j}$, $i = 3, \cdots\cdots, n$

$b'_i = b_i - m_i b_2$, $j = 2, \cdots\cdots, n$

이므로 위방법과 같이 RENAME 하면 아래와 같은 모양의 방정식이 된다.

$$\left.\begin{array}{l} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots\cdots + a_{1j}x_j + a_1 x_n = b_1 \\ \quad a_{22}x_2 + a_{23}x_3 + \cdots\cdots + a_{2j}x_j + a_2 x_n = b_2 \\ \quad\quad a_{33}x_3 + \cdots\cdots + a_{3j}x_j + a_3 x_n = b_3 \\ \quad\quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ \quad\quad a_{n3}x_3 + \cdots\cdots + a_{nj}x_j + a_{nn}x_n = b_n \end{array}\right\} \cdots\cdots\cdots\cdots (2-4)$$

3) K번째 단계

$a_{kk} \neq 0$ 일 때

$m_i = \frac{a_{ik}}{a_{kk}}$, $i = k+1, \cdots\cdots, n$

그러면

$a'_{ij} = a_{ij} - m_i a_{kj}$, $i = k+1, \cdots\cdots, n$

$b'_i = b_i - m_i b_k$, $j = k, \cdots\cdots, n$

이렇게하여 N－1 단계에서 x_{n-1} 을 소거하면 식 (2－5)와 같은 연립방정식이 된다. 이것은 (2－1) 식과 동일한 것이다.

$$\left.\begin{array}{l} a_{11}x_1 + a_{12}x_2 + \cdots\cdots + a_{1j}x_j + \cdots\cdots + a_{1n}x_n = b_1 \\ \quad a_{22}x_2 + \cdots\cdots + a_{2j}x_j + \cdots\cdots + a_{2n}x_n = b_2 \\ \quad\quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ \quad\quad a_{jj}x_j + \cdots\cdots + a_{jn}x_n = b_j \\ \quad\quad\quad \cdots\cdots\cdots\cdots \\ \quad\quad\quad a_{nn}x_n = b_n \end{array}\right\} \cdots\cdots\cdots\cdots (2-5)$$

이렇게 삼각형방정식을 얻고나서는 (2－5)식의 최후의 방정식으로부터 후방대입법으로 x_n 을 구하고 $x_{n-1}, \cdots\cdots x_1$ 으로 차례로 방정식을 풀어 나가면 식 (2－1)의 해집합 $x_1, x_2, \cdots\cdots, x_n$ 을 구할 수 있다.

$$\left.\begin{array}{l} x_n = \dfrac{b_n}{a_{nn}} \\ \vdots \\ x_j = \dfrac{b_j - (a_{j,\,j+1}x_{j+1} + \cdots\cdots + a_{jn}x_n)}{a_{jj}} \\ \vdots \\ x_1 = \dfrac{b_1 - (a_{12}x_2 + \cdots\cdots + a_{1n}x_n)}{a_{11}} \end{array}\right\} \cdots\cdots\cdots\cdots (2-6)$$

이 소거과정에서 분수가 생기는데 계산기에서는 분수를 유한소수로 표시하게 되므로 해에는 오차가 생긴다. 이 오차는 ROUND OFF 오차라고 한다. 상당히 큰 방정식일 때 ROUND OFF ERROR의 누적효과는 해의 오차에 대하여 비교적 크므로 일반적으로 GAUSS의 소거법을 만족하는 해를 얻기 위한 연립방정식의 원소의 수는 15 내지 20개까지로 제안하고 있다. 이 오차를 될수록 작게하려고 축원소(PIVOT)로는 일반적으로 행교환을 해서 절대치가 큰 것을 택하기로 한다. 이와같은 방법을 피봇전략법(PIVOTING STRATEGY)이라고 말한다. 이 밖에도 여러가지의 나쁜조건(ILL-CONDITIONED)이 많아서 그 수정방법도 여러가지가 있다.

다음은 연립 1 차방정식을 구하는 일반적인 흐름도이다.

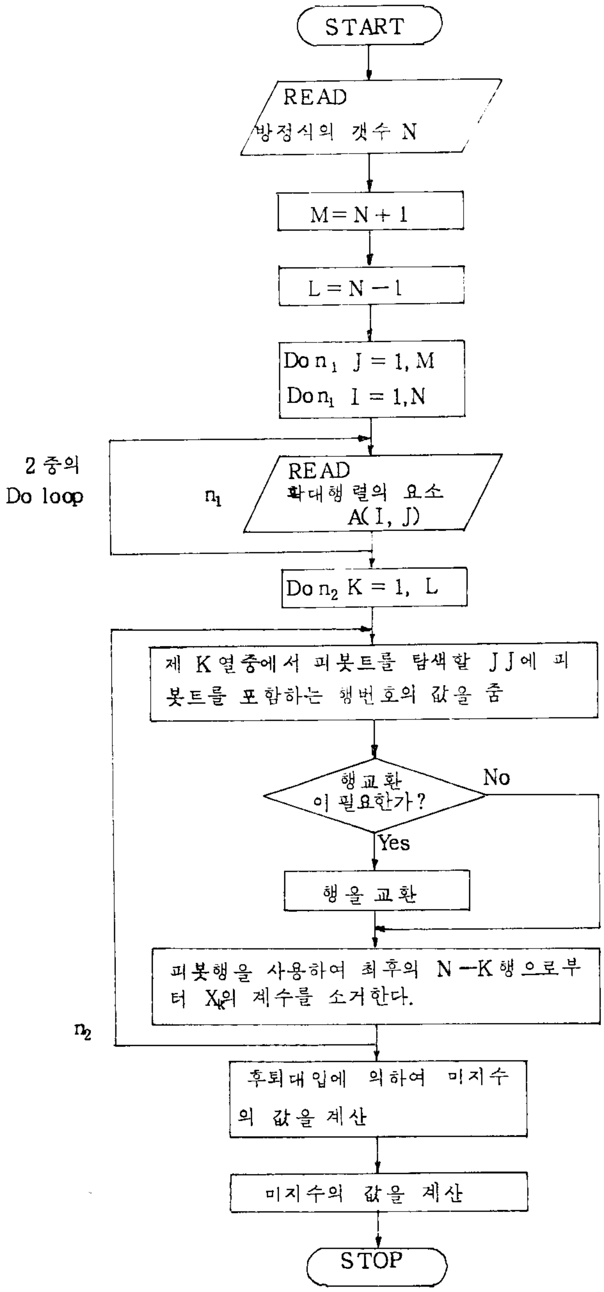


그림 2 − 1. **Gaussian Elimination Method의 Flow chart**

2.3 GAUSS JORDAN ELIMINTION METHOD

이 방법은 GAUSS 소거법의 변형으로 15 내지 20개의 미지수를 가진 연립 1 차방정식의 해를 구하는데 적당하다. 이 방법이 GAUSS 소거법과의 차이점은 미지수를 소거할 때에 PIVOT 방정식 뿐만 아니라 그 전의 방정식도 포함한 전부의 방정식으로부터 소거하는 점이다. 그러니까 확대행렬(AUGMENTED METRIX) (1 − 3)식에서 대각선 요소 이외에는 전부 0으로 소거한다.

GAUSSIAN ELIMINATION와 비슷한 소거 경로를 거쳐 식 (3 − 1)과 같이 변환된다.

$$\left.\begin{array}{llllll}
a_{11} & 0 \cdots\cdots 0 & \cdots\cdots 0 & b_i \\
0 & a_{22} \cdots & 0 \cdots\cdots 0 & b_2' \\
\cdots\cdots\cdots & \cdots\cdots\cdots & \cdots\cdots\cdots & \\
0 & 0 \cdots\cdots a_{ii} & \cdots 0 & b_i' \\
\cdots\cdots\cdots & \cdots\cdots\cdots & \cdots\cdots\cdots & \\
0 & 0 \cdots\cdots & 0 \cdots\cdots a_{nn} & b_n'
\end{array}\right\} \cdots\cdots\cdots\cdots \quad (3-1)$$

이때 SOLUTION VECTOR X는

$$\left.\begin{array}{l}
x_1 = b_1' / a_{11} \\
x_2 = b_2' / a_{22} \\
\vdots \\
x_n = b_n' / a_{nn}
\end{array}\right\} \cdots\cdots\cdots\cdots \quad (3-2)$$

이다.

FLOW-CHART는 다음과 같다.

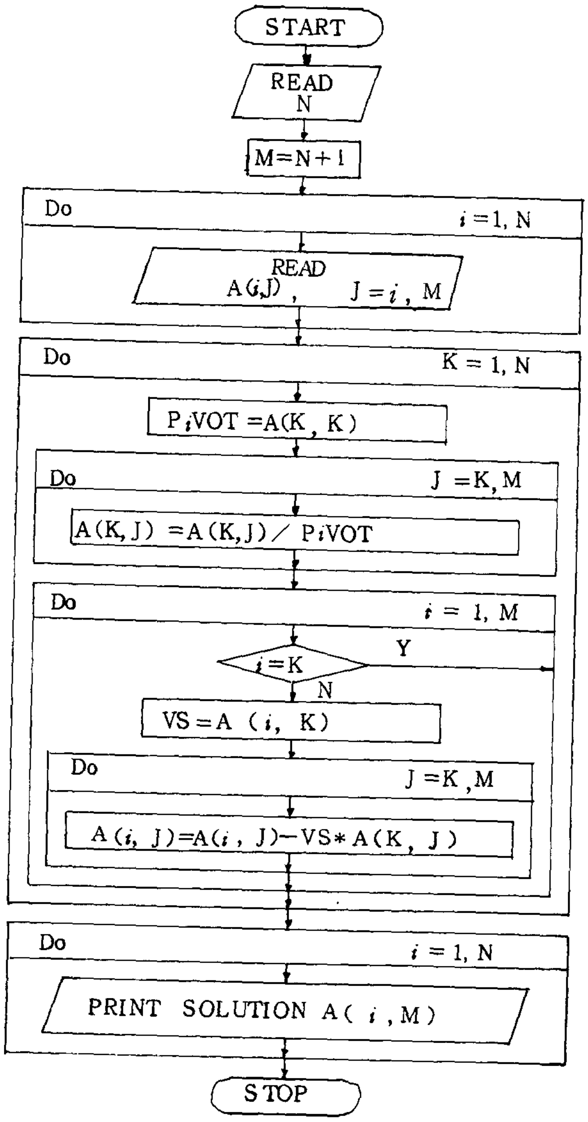


그림 3－1. **Gauss－Jordan Elimination Method 의 Flow chart**

2.4 INVERSE MATRIX에 의한 해법(INVERSE MATRIX METHOD)

식 (1-4)의 행렬방정식 $AX = B$는 행렬의 정의에 의해 다음과 같이 변환이 되어진다.

$$A^{-1}AX = A^{-1}B$$
$$IX = A^{-1}B$$
$$\therefore X = A^{-1}B$$

따라서 A^{-1}를 구할 수 있다면 SOLUTION VECTOR X도 구할 수 있다.

A의 INVERSE MATRIX는 다음의 예와 같은 수치적 방법으로 구하는 것이 전자계산기를 이용하는데 편리하다.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & 1 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 & 1 & 0 \\ a_{31} & a_{32} & a_{33} & 0 & 0 & 1 \end{bmatrix}$$

왼쪽의 A의 요소를 다음과 같이 소거할 수 있다.

$$\begin{bmatrix} 1, & a_{12}/a_{11}, & a_{13}/a_{11}, & 1/a_{11}, & 0, & 0 \\ 0, & a_{22}-a_{21}(a_{12}/a_{11}), & a_{23}-a_{21}(a_{13}/a_{11}), & -a_{21}/a_{11}, & 1, & 0 \\ 0, & a_{32}-a_{31}(a_{12}/a_{11}), & a_{33}-a_{31}(a_{13}/a_{11}), & -a_{31}/a_{11}, & 0, & 0 \end{bmatrix}$$

A의 요소가 I MATRIX가 될 때까지 소거를 하면

$$\begin{bmatrix} 1 & 0 & 0 & c_{11} & c_{12} & c_{13} \\ 0 & 1 & 0 & c_{21} & c_{22} & c_{23} \\ 0 & 0 & 1 & c_{31} & c_{32} & c_{33} \end{bmatrix}$$

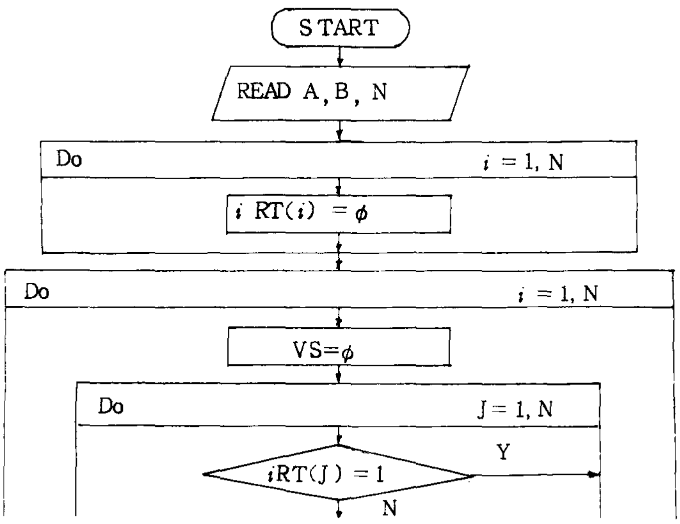
와 같은 형태로 변환된다.

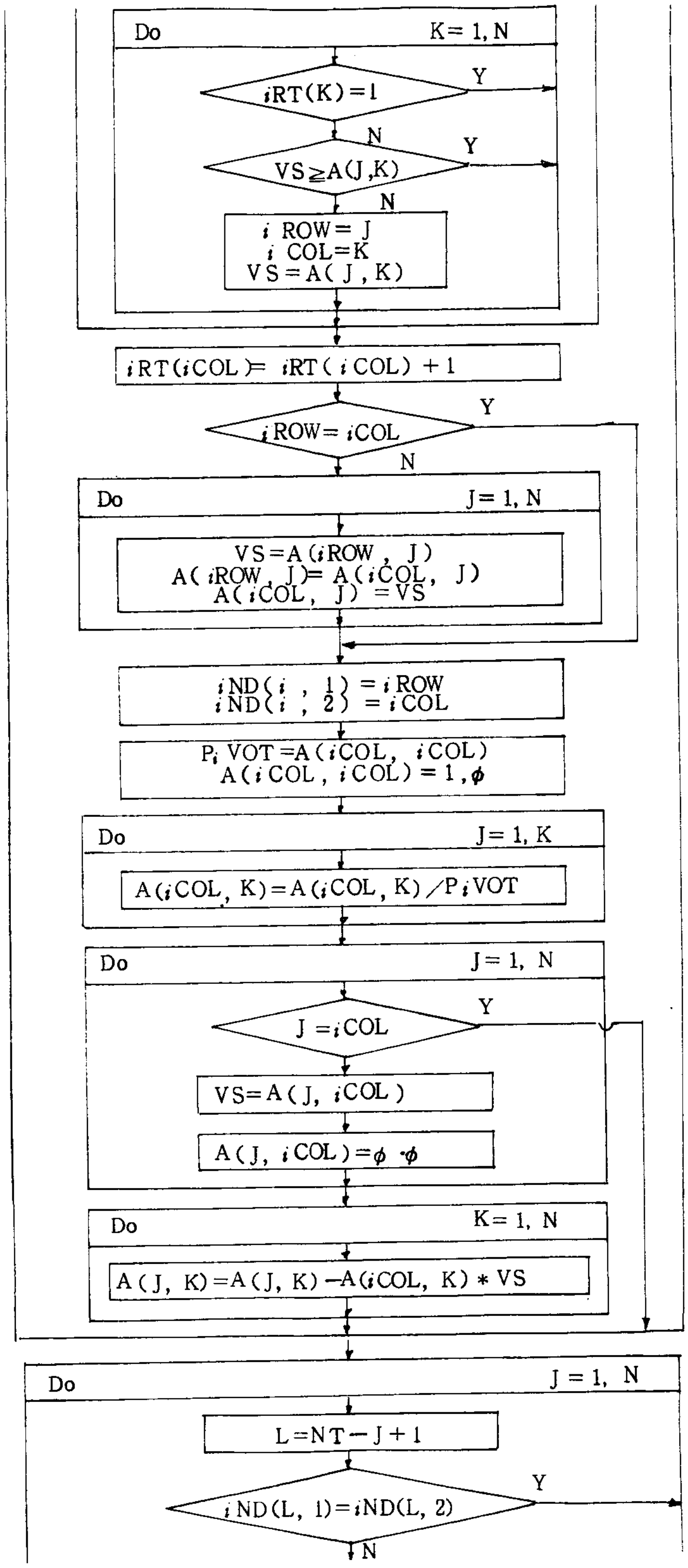
이때 c_{ij}는 A_{ij}의 INVERSE MATRIX가 된다.

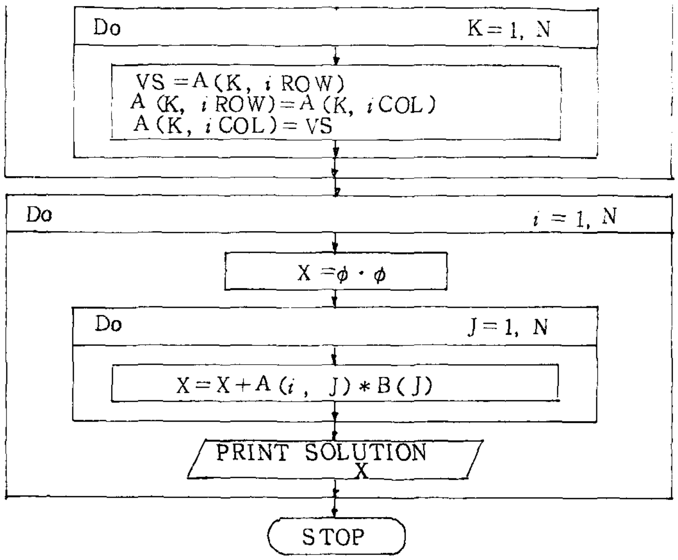
SOLUTION VECTOR X는

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

이고 FLOW-CHART는 다음공 같다.




Do K= 1, N
iRT(K)=1
Y
N
VS≧A(J,K)
Y
N
i ROW= J
i COL=K
VS=A(J , K)
iRT(iCOL)= iRT(i COL) + 1
iROW= iCOL
Y
N
Do J= 1, N
VS=A (iROW , J)
A(iROW, J)= A(iCOL, J)
A(iCOL, J) =VS
iND(i , 1) = i ROW
iND(i , 2) = i COL
Pi VOT =A (iCOL, iCOL)
A(i COL , i COL) = 1 ,ϕ
Do J= 1, K
A(iCOL, K)=A(iCOL, K) /PiVOT
Do J= 1, N
J =iCOL
Y
VS=A(J, iCOL)
A(J, iCOL)=ϕ ·ϕ
Do K= 1, N
A (J, K) =A (J, K) –A(iCOL, K) * VS
Do J = 1, N
L=NT–J+1
iND(L, 1)=iND(L, 2)
Y
N



2.5 GAUSS SEIDEL ITERATIVE METHOD

식 (2－1)의 연립 1차방정식에서 (K＝1, 2, ……, N)이면 X_k에 대하여 변환하여 풀면 식 (4－1)과 같의 된다.

$$\left.\begin{aligned} x_1 &= \frac{1}{a_{11}}(b_1 - a_{12}x_2 - \cdots\cdots - a_{1j}x_j - \cdots\cdots - a_{1n}x_n) \\ x_2 &= \frac{1}{a_{22}}(b_2 - a_{21}x_1 - \cdots\cdots - a_{2j}x_j - \cdots\cdots - a_{2n}x_n) \\ &\cdots\cdots\cdots\cdots\cdots\cdots \\ x_n &= \frac{1}{a_n}(b_n - a_{n1}x_1 - \cdots\cdots - a_{nj}x_j - \cdots\cdots - a_{n,n-1}x_{n-1}) \end{aligned}\right\} \cdots\cdots\cdots\cdots (4-1)$$

식 (4－1)의 초기가정치 $x_1^{(0)}$, $x_2^{(0)}$, ……, $x_n^{(0)}$를 대입하여 개량치 $x_1^{(1)}$, $x_2^{(1)}$를 식 (4－2)와 같이 얻을 수 있다.

$$\left.\begin{aligned} x_1^{(1)} &= \frac{1}{a_{11}}(b_1 - a_{12}x_2^{(0)} - a_{13}x_3^{(0)} - \cdots\cdots - a_{1j}x_j^{(0)} - \cdots\cdots - a_{1n}x_n^{(0)}) \\ x_2^{(1)} &= \frac{1}{a_{22}}(b_2 - a_{21}x_1^{(0)} - a_{23}x_3^{(0)} - \cdots\cdots - a_{2j}x_j^{(0)} - \cdots\cdots - a_{2n}x_n^{(0)}) \\ &\cdots\cdots\cdots\cdots\cdots\cdots \\ x_n^{(2)} &= \frac{1}{a_{nn}}(b_n - b_{n1}x_1^{(0)} - a_{n2}x_2^{(0)} - \cdots\cdots - a_{nj}x_j^{(0)} - \cdots\cdots - a_{n,n-1}x_{n-1}^{(0)}) \end{aligned}\right\} \cdots (4-2)$$

식 (4－2)에서 구한 x_i^1을 다시 식 (4－1)에 대입하여 x_i^2를 구한다. 이와같이 반복하면 K번째 근사해는

$$\left.\begin{aligned} x^{(k)} &= \frac{1}{a_{11}}(b_1 - a_{12}x_2^{(k-1)} - \cdots\cdots - a_{1j}x_j^{(k-1)} \cdots\cdots - a_{1n}x_n^{(k-1)}) \\ x_2^{(k)} &= \frac{1}{a_{22}}(b_2 - a_{21}x^{(k-1)} - \cdots\cdots - a_{2j}x_j^{(k-1)} \cdots\cdots - a_{2n}x_n^{(k-1)}) \\ &\cdots\cdots\cdots\cdots\cdots\cdots \\ x_n^{(k)} &= \frac{1}{a_{nn}}(b_n - a_{n1}x_1^{(k-1)} - \cdots\cdots - a_{nj}x_j^{(k-1)} \cdots\cdots - a_{n,n-1}x_{n-1}^{(k-1)}) \end{aligned}\right\} \cdots\cdots (4-3)$$

과 같이 계산되며 허용오차(TOLERANCE)가 $|x_i^{(k)} - x_i^{(k-1)}| < \varepsilon$ 일때 근사해 x_i를 찾을 수 있다.
이에 관한 FLOW-CHART는 다음과 같다.

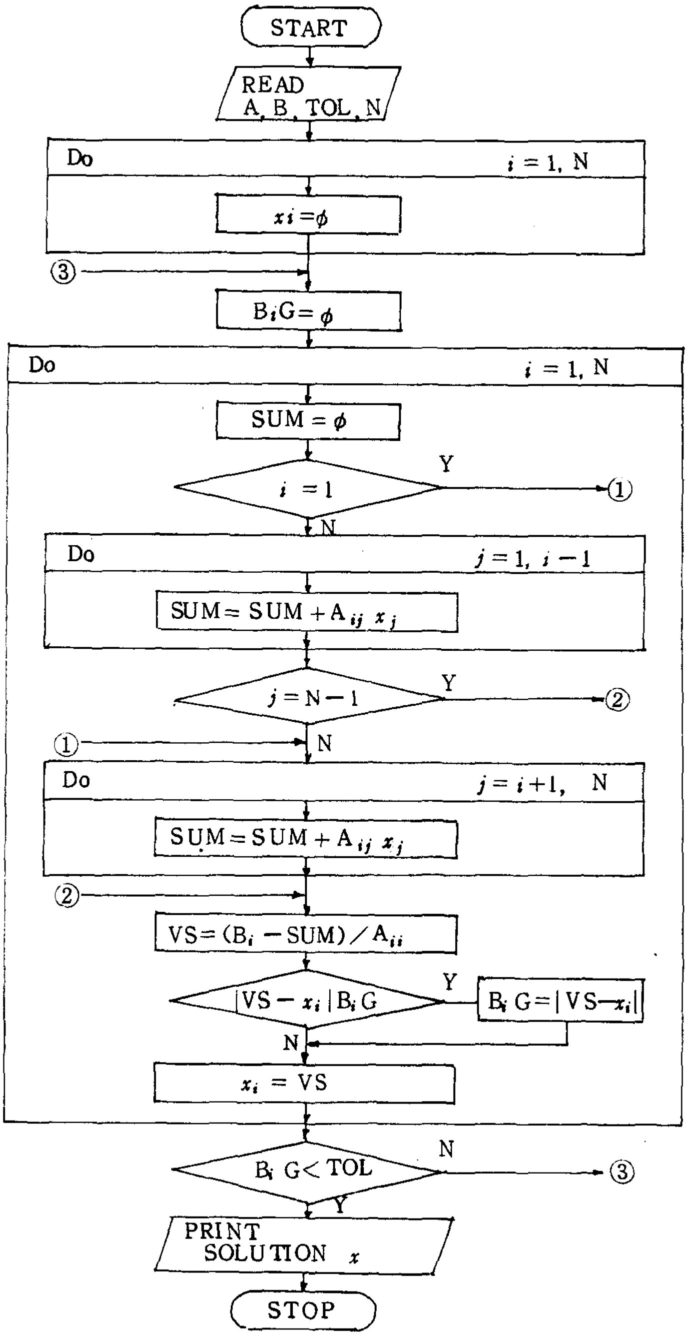


그림 3 − 1. **Gauss-Seidel Iterative Method 의 Flow chart**

2.6 연산작업 비교

N개의 미지수의 N원 연립 1 차방정식의 해를 구하는 각각의 METHOD들은 다음 표와 같은 ARITHMETIC OPERATION이 일어난다.

THE NUMBERS OF ARITHMETIC OPERATIONS (N * N MATRIX)

	*	÷	+	−
GAUSSIAN ELIMINATION	$\frac{1}{3}n^3 + n^2 + \frac{2}{3}n - 1$	$\frac{1}{2}n^2 + \frac{3}{2}n$	$\frac{1}{2}n^2 + \frac{3}{2}n$	$2n + 1$
GAUSS - JORDAN	$n^3 + n^2$	$n^2 + n$	1	$n^3 + n^2$
INVERSE MATRIX	$2n^2 - n$	n^2	$n^3 + n - 1$	$n^2 - n + 1$
GAUSS - SEIDEL	$n^2 - n$	n	$n^2 - n$	$n + \alpha$

* GAUSS - SEIDEL METHOD는 한번 반복당의 연산회수, $\alpha < N$

위 표에서 덧셈, 뺄셈, 나눗셈, 곱셈의 연산을 합하면 대략 다음과 같은 연산회수를 가진다.

1. GAUSSIAN ELIMINATION METHOD ······ $(2n^2 + 12n^2 + 34n)/6$번
2. GAUSS - JORDAN METHOD ······ $(2n^3 + 3n^2 + n)$번
3. INVERSE MATRIX METHOD ······ $(5n^2 - n)$번
4. GAUSS SEIDEL METHOD (PER ITERATION) ······ $(2n^2 + \alpha)$번

과 같은 ARITHMETIC OPERATION이 있으며, GAUSSIAN ELIMINATION에서는 약 $n(n-1)/2$번, INVERSE MATRIX에서는 약 n^3번의 비교 연산이 있다.

예를 들어 각 METHOD를 비교해 보자. 만약 미지수가 100개인 100원 연립 1 차방정식이라면

1. GAUSSIAN ELIMINATION ······ 약 354, 000번
2. GAUSS-JORDAN ······ 약 2, 030, 000번
3. INVERSE MATRIX ······ 약 50, 000번
4. GAUSS SEIDEL (PER ITERATION) ······ 약 20, 000번

와 같은 OPERATION이 일어난다.

따라서 GAUSS - JORDAN METHOD는 N이 크면 클수록 연산의 회수가 지수적으로 늘어나므로 OPERANTION시간도 지수적으로 늘어난다.

또 GAUSS-SEIDEL METHOD와 GAUSSIAN ELIMINATION을 비교해 보면 GAUSS - SEIDEL METHOD가 반복회수가 18번 이하일 경우에는 GAUSSIAN ELIMINATION METHOD보다 빠르게 해를 구할 수 있다.

그러나 GAUSS-SEIDEL METHOD는 연립 1 차방정식의 조건이 수렴할 수 있어야 하며 그렇지 못할 경우에는 무한히 연산을 하게 된다. 그러나 네 가지의 방법 가운데 ROUND OFF ERROR가 가장 적으며 해에 대한 근사값으로부터 시작하여 요구하는 정확한 값을 얻기 위하여 필요한 만큼 반복을 하므로 보다 정확한 값을 구할 수 있는 방법이다. 반복회수와 정확도는 주어진 TOLERANCE에 따라 좌우되며 계수행렬의 주 대각원소의 절대값이 적을 경우에는 수렴이 되지 않으므로 이 방법으로는 해를 구할 수 없다.

GAUSSIAN ELIMINATION METHOD와 GAUSS-JORDAN METHOD는 연산의 회수가 많으므로 GAUSS-SEIDEL METHOD보다 ROUND OFF ERROR가 많이 발생된다.

5. 결　론

COMPUTER에 의한 다원연립 일차방정식 해법으로, GAUSS ELIMINATION METHOD, GAUSS-JORDAN METHOD, INVERSE MATHOD, GAUSS-SEIDEL ITERATIVE METHOD를 비교하여 그 차이점과 연산 회수와 그로 인한 오차 등을 검토해 보았다. 그 중 INVERSE MATRIX에 연산 회수가 가장

적었으나 비교 연산이 N^3 (N은 미지수의 갯수)회나 있기 때문에 연산속도는 빠르지 못함도 알았다.

무엇보다도 이 네가지 방법이 거의 다 전자계산기의 기억 용량을 많이 차지해야 한다는 취약점을 가지고 있다. 보통의 기업체나 학교에서 쉽게 사용되는 소형 또는 중형 전자계산기에서는 미지수가 많은 다원연립 일차 방정식을 취급하기에는 조심스럽거나 취급할 수 없다. 그러므로 소형 전자계산기에서도 능히 구할 수 있는 새로운 해법의 개발이 필요하다고 본다.

INVERSE MATRIX METHOD에서 MATRIX의 분할 처리로서 해를 구하는 방법도 연구할 수 있을 것이다. 대수적으로 가능성이 입증되고 있지만 그 기술적인 뒷받침이 연구되어져야 할 것이다.

또 하나는 연산 횟수 및 비교 횟수를 최소화하는 문제가 연구되어져야 할 것이다. 연산 횟수에 최소화는 오차의 최소화와 연산 시간의 최소화를 의미하는 것이다.

다른 각도로서의 연구할 점은, 현재 사용하고 있는 해법에 있어서도 다원연립일차방정식에 포함된 미지수의 갯수에 따른 전자계산기의 사용 기억 용량을 계산하는 문제이다. 이러한 계산결과는 실제 업무 집행에 많은 도움을 줄 것이다.

참 고 문 헌

1. J.N.Franklin, *Matrix Theory*, Prentice-Hall, INC., 1968.
2. J.R.Westlake, *A Handbook of Numerical Matrix Inversion And Sdution of Linear Equations*, John Wiley & Sons, Inc., 1968.
3. D.D.Mccracken, W.S.Dorn, *Numerical Methods with Fortran IV Case Studies*, John Wiley & Sons, Inc., 1972.
4. S.S.Kuo, *Computer Applications of Numerical Methods*, Addison-Wesley Publishing Company, Inc., 1972.
5. I.B.M.System/360 Scientific Subroutine Package, 1970.
6. 金星一・李京煥, 數値解析, 螢雪出版社, 1975.
7. James, Smith & Wolford, 名取亮(譯), FORTRAN 1 =よる數値計算法の應用, 科學技術出版, 1978.