電子計算機에 依한 超大形 Matrix의 한 解法

Tri-diagonal Matrix Operation Scheme on a Computer

金 德 鉉

Kim, Duk Hyun, Ph.D.

서 론

고속도 전자계산기의 대두는, 종래에 응력해석이 거의 불가능하다고 생각되던 많은 문제들의 해결을 가능케 하였다. 그러나 많은 사람들이 착각하고 있드시 전자계산기가 모든 문제를 자동적으로 해결하는 것은 결코 아니다. 계산기의 성능이 발달할수록 더욱 더 어려운 문제의 해결이 가능해지는 것은 사실이나, 그렇게 되기 위해서는 사람의 머리에서 만들어 내야 하는 이론의 발달이 더욱 더 시급해지며 중요성을 띄게 되는 것이다. 즉 기계의 발달은 사람의 중요성을 더욱 절실하게 만드는 아이로니칼한 현상을 갖어 오게 된 것이다. 이러한 기계에 비한 사람의 중요성을 강조하기 위한 한 실예로서 본 논문에서는 방대한 크기를 가진 matrix equation 을 푸는 한 방법을 다루기로 했다.

일반적으로, 대부분의 응력해석을 할 때는 대단히 많은 미지수를 가진 연립방정식을 푸는 문제로 귀결되는 경우가 흔히 있다. 이럴 경우 소위 computer의 memory core의 제한은 이 matrix equation을 직접 다루는 것을 불가능하게 하는 고로 특별한 matrix operation을 해야만 하는데, 이 글에서는 필자가 과거에 유도하여 사용해온 이론 가운데 응력해석을 위해서는 가장 적절하다고 느꼈던 한 방법을 발표하여 장차 우리 나라에도 쓸만한 전자계산기가 도입될 경우 많은 기술자가 사용하는데 조금이나마 도움이 되게끔 하고자 한다.

이론의 전개

어느 곡면을 MI개의 점을 가진 MJ개의 선으로 나눈다면, 이 곡면의 영역에 대해서 다음과 같은 Matrix 방정식이 성립할 것이다.

$$
\begin{aligned}
&T_1X_1+S_{22}X_2+S_{23}X_3 &&=D_1\\
&S_1X_1+TX_2+S_2X_3 &&=D_2\\
&S_1X_2+TX_3+S_2X_4 &&=D_3\\
&\cdots\cdots\\
&\cdots\cdots\\
&S_1X_{MJ-3}+TX_{MJ-2}+S_2X_{MJ-1} &&=D_{MJ-2}\\
&S_1X_{MJ-2}+TX_{MJ-1}+S_2X_{MJ} &&=D_{MJ-1}\\
&S_{33}X_{MJ-2}+S_{32}X_{MJ-1}+T_2X_{MJ} &&=D_{MJ}
\end{aligned}
\quad\cdots\cdots(1)
$$

여기서 각 Matrix 항들은, shell 인 경우는 ($4MI\times4MI$), plate 인 경우는 ($2MI\times2MI$)의 size 를 갖고 있으며 T_1, S_{22}, S_{23}, T_2, S_{32}, S_{33} 등은 경계에서의 fictitious line 들의 처리에서 생기는 matrix 들이다[1][1]. (1)식은 다음과 같은 간단한 식으로 표현될 수 있다.

$$A\cdot X=D \quad\cdots\cdots(2)$$

여기서 A, X 및 D는 다음과 같이 표시되는 matrix 이다.

$$
A=\left\{\begin{matrix}
T_1 & S_{22} & S_{23} & & & & \\
S_1 & T & S_2 & & & & \\
 & S_1 & T & S_2 & & & \\
 & & \cdots\cdots & & & & \\
 & & \cdots\cdots & & & & \\
 & & & S_1 & T & S_2 & \\
 & & & & S_1 & T & S_2\\
 & & & & S_{33} & S_{32} & T_2
\end{matrix}\right\}\quad\cdots\cdots(3)
$$

$$
X=\left\{\begin{matrix} X_1\\ X_2\\ \vdots\\ X_{MJ-1}\\ X_{MJ}\end{matrix}\right\}\quad\cdots\cdots(4)
$$

$$
D=\left\{\begin{matrix} D_1\\ D_2\\ \vdots\\ D_{MJ-1}\\ D_{MJ}\end{matrix}\right\}\quad\cdots\cdots(5)
$$

A^{-1}를 A의 inverse 라 하면 X는

$$X=A^{-1}D \quad\cdots\cdots(6)$$

*正會員・陸軍士官學校 敎官・陸軍少領・工學博士

와 같이 구할 수 있고 stress tensor 는 X에 stiffness matrix 를 premultiply 함으로서 구할 수 있다.

그러나 (2)식의 각 matrix 는 복합 matrix 로서 그 size 가 너무 커, 경계조건에 따라 미지수는 수 백만개도 될 수 있는 만치 전자계산기가 고도로 발달되었다 하나 A^{-1}를 직접 구한다는 것은 거의 불가능하다.

따라서 computer 를 이용할 경우 사용해야만 할 sub program 으로서 다음과 같은 특별한 방법이 필요한 것이다.

(1)식의 첫번 방정식으로부터

$$X_1+T_1^{-1}S_{22}X_2+T_1^{-1}S_{23}X_3=T_1^{-1}D_1 \cdots\cdots\cdots\cdots (7)$$

양변의 각 항에 S_1 matrix 를 premultiply 하면 다음 식을 얻는다.

$$S_1X_1+S_1T_1^{-1}S_{22}X_2+S_1T_1^{-1}S_{23}X_3=S_1T_1^{-1}D_1 \cdots (8)$$

(1)의 둘째 식에서 (8)식을 빼면

$$(T-S_1T_1^{-1}S_{22})X_2+(S_2-S_1T_1^{-1}S_{23})X_3$$
$$=D_2-S_1T_1^{-1}D_1 \cdots\cdots\cdots\cdots (9)$$

새로운 matrix 들을

$$P_1=T-S_1T_1^{-1}S_{22}$$
$$Q_1=S_2-S_1T_1^{-1}S_{23}$$
$$C_1=D_2-S_1T_1^{-1}D_1 \cdots\cdots\cdots\cdots (10)$$

과 같이 정의하면 (9)식은 다음과 같이 된다:

$$P_1X_2+QX_3=C_1 \cdots\cdots\cdots\cdots (9a)$$

P_1^{-1}를 양변에 곱하면

$$X_2+P_1^{-1}Q_1X_3=P_1^{-1}C_1 \cdots\cdots\cdots\cdots (11)$$

S_1을 양변에 곱하면

$$S_1X_2+S_1P_1^{-1}Q_1X_3=S_1P_1^{-1}C_1 \cdots\cdots\cdots\cdots (12)$$

(1)의 셋째식에서 (12)식을 빼면

$$(T-S_1P^{-1}Q_1)X_3+S_2X_4=D_3-S_1P_1^{-1}C_1 \cdots\cdots (13)$$

새로운 matrix 들을

$$P_2=T-S_1P_1^{-1}Q_1$$
$$C_2=D_3-S_1P_1^{-1}C_1 \cdots\cdots\cdots\cdots (14)$$

라 정의하면 (13)식은

$$P_2X_3+S_2X_4=C_2 \cdots\cdots\cdots\cdots (15)$$

가 되고, 다시

$$X_3+P_2^{-1}S_2X_4=P_2^{-1}C_2 \cdots\cdots\cdots\cdots (16)$$

및

$$S_1X_3+S_1P_2^{-1}S_2X_4=S_1P_2^{-1}C_2 \cdots\cdots\cdots\cdots (17)$$

를 얻고 (17)을 (1)의 넷째식에서 빼면

$$(T-S_1P_2^{-1}S_2)X_4+S_2X_5=D_4-S_1P_2^{-1}C_2 \cdots\cdots (18)$$

이 얻어지고

$$P_3=T-S_1P_2^{-1}S_2$$
$$C_3=D_4-S_1P_2^{-1}C_2 \cdots\cdots\cdots\cdots (19)$$

라 하면 (18)은

$$P_3X_4+S_2X_5=C_3 \cdots\cdots\cdots\cdots (20)$$

와 같이 표시될 수 있고

$$X_4+P_3^{-1}S_2X_5=P_3^{-1}C_3 \cdots\cdots\cdots\cdots (21)$$

를 얻을 수 있다.

이와 같은 일을 계속해 나가면 $MJ-2$선에서는

$$X_{MJ-2}+P_{MJ-3}^{-1}S_2X_{MJ-1}=P_{MJ-3}^{-1}C_{MJ-3} \cdots\cdots (22)$$

를 얻을 수 있고 $MJ-1$에서는

$$X_{MJ-1}+P_{MJ-2}^{-1}S_2X_{MJ}=P_{MJ-2}^{-1}C_{MJ-2} \cdots\cdots (23)$$

를 얻을 수 있다.

(22)에 S_{33}를 곱하고 (1)의 마지막 식에서 빼면

$$(S_{32}-S_{33}P_{MJ-3}^{-1}S_2)X_{MJ-1}+T_2X_{MJ}$$
$$=D_{MJ}-S_{33}P_{MJ-3}^{-1}C_{MJ-3} \cdots\cdots\cdots\cdots (24)$$

또는

$$P_{MJ-1}=S_{32}+S_{33}P_{MJ-3}^{-1}S_2$$
$$C_{MJ-1}=D_{MJ}-S_{33}P_{MJ}^{-1}C_{MJ-3} \cdots\cdots\cdots\cdots (25)$$

라 할때

$$P_{MJ-1}X_{MJ-1}+T_2X_{MJ}=C_{MJ-1} \cdots\cdots\cdots\cdots (26)$$

를 얻게 되고 (23)식에 P_{MJ-1}를 곱하고 (26)식에서 빼면

$$(T_2-P_{MJ-2}^{-1}S_2)X_{MJ}$$
$$=C_{MJ-1}-P_{MJ-1}P_{MJ-2}^{-1}C_{MJ-2} \cdots\cdots\cdots\cdots (27)$$

또는

$$P_{MJ}=T_2-P_{MJ-1}P_{MJ-2}^{-1}S_2$$
$$C_{MJ}=C_{MJ-1}-P_{MJ-1}P_{MJ-2}^{-1}C_{MJ-2} \cdots\cdots\cdots\cdots (28)$$

라 할때

$$P_{MJ}X_{MJ}=C_{MJ} \cdots\cdots\cdots\cdots (29)$$

와 같은 간단한 식을 얻을 수 있다.

이 식으로부터 X_{MJ}를 구하면

$$X_{MJ}=P_{MJ}^{-1}C_{MJ} \cdots\cdots\cdots\cdots (30)$$

이것을 (23)에 대입하면 X_{MJ-1}을 얻고 X_{MJ-1}을 (22)에 대입하면 X_{MJ-2}를 얻을 수 있다. 이러한 과정을 계속하여 X_2까지 구한 다음 (7)식에 X_2 및 X_3를 대입하면 X_1을 구할 수가 있어 결국 solution vector X를 다 구한 셈이 된다. 여기서 solution vector X의 dimension 은 shell 인 경우에는 $4MI \times MJ$, plate 나 평면탄성체인 경우에는 $2MI \times MJ$가 될 것이다.

Program 의 Outline

Computer program 의 한 예로서, S_{23} 및 S_{33}가 Null matrix 인 경우의 개요를 소개하면 다음과 같다.

DIMENSION 및 *COMMON* Statemeat 가 끝나면 tape #1 을 rewind 하면서 이 Subroutine 이 시작된다.

```
REWIND 1
TT(I,J)=T(I,J)
SS(I,I)=S22(I,J)
XE(I,1)=D(I,1)
JJ=1
CALL MATINV(TT⁻¹를 구함)
```

```
5    CALL MTMU (TT^-1.SS)(2)
     CALL MTMU (TT^-1.XE)
     XXE (I,JJ)=XE (I,1)
     IF (JJ. GE. MJ) Gϕ Tϕ 200
     JJ=JJ+1
     IF (JJ. GE. MJ) Gϕ Tϕ
     CALL MTMU (S1·SS)
     CALL MTMU (S1·XE)
     Gϕ Tϕ 502
500  CALL MTMU (S32·SS)
     CALL MTMU (S32·XE)
502  TT(I,J)=TE(I,J)-SS(I,J)
     CALL MATINV (TT^-1)
     IF(JJ. GE. MJ) Gϕ Tϕ 150
     SS(I,J)=S2 (I,J)
     XE(J,1)=D (I,JJ)-XE (I,1)
     Gϕ Tϕ 5
150  XE(I,1)=D (I,JJ)-XE (I,1)
     CALL MTMU(TT, XE)
     XXE(I,JJ)=XE (I,1)
200  BACKSPACE 1
     JB=MJ-1
160  READ (1) SS
     XXE(I,JB)=XXE (I,JB)
       -SS (I,J) XXE (J,JB+1)
     IF(JB.LE.1) Gϕ Tϕ 250
     BACKSPACE 1
     BACKSPACE 1
     JB=JB-1
     Gϕ Tϕ 160
```

여기서는 program의 outline만 설명하기 위하여 구체적인 *Dϕ LϕϕP*나 check, write, routine. 등은 생략하였다.

결 근

서론에서 언급되었드시, 응력해석을 할때는 수천 수만, 때로는 수백만의 미지수를 가진 matrix 방정식을 풀어야 할 경우가 흔히 있다. computer가 대두되기 전에는 이런 문제를 다룬다는 것은 상상도 할 수 없는 일이었으나, 현재에 이르러서는「흔히 있을 수 있는 일」이 되어 버렸다. 그러나 전자계산기에도 한계가 있어, 한번의 operation으로 inversion을 구할 수 있는 matrix의 size는 극히 제한되어 있는 것이다. 본 논문에 실린 방법은 필자가 유도한 이론가운데 이런 경우에 사용될 수 있는 가장 효과적이고 정확한 방법중의 하나로서 비단 응력해석뿐만 아니라 많은 초대형 size의 matrix equation의 solution을 구할 때 쉽게 이용될 수 있는 이론일 것이다. 응력해석의 경우 공식 (2)의 A·matrix의 한 element는 (60×60)가 될 경우가 흔히 있는데, 40×40인 경우 desk calculator로서 64주 걸리는 것을 요지음 기계로 0.3초 만에 invert 할 수 있다는 것을 생각할 때, 대부분의 응력해석문제는 이 글의 이론을 쓰면 수십초만에 해결될 수 있다는 것을 용이하게 이해할 수 있을 것이다.

현재 이 matrix operation의 필요성이나 중요성을 이해 못하는 독자가 간혹 있을지 모르나, 가까운 장래에 우리 나라에 computer가 도입될 경우 이 이론을 자주 사용하게 될 것이라는 것을 확신하면서 이글을 글맺는다.

註

1. 괄호 안의 숫자는 참고 서적의 순서를 뜻한다.
2. MTMU의 결과는 뒤의 matrix에 store 된다. 즉 $SS=TT^{-1}\cdot SS$

참고서적

1. 김덕현, "Matrix에 의한 Multiple Shell의 한 해법", 대한토목학회지, 제13권 제4호, p. 9.