

용접 잔류 응력의 이해를 위한 Three Bar 모델 해설

Three Bar Model to Understand the Thermal Residual Stress in Welding



이 장 현*

*인하대학교 기계공학부 선박해양공학과 교수

1. 개 요

용접에 의한 구조물의 접합부는 필연적으로 잔류 응력이 가지고 있다. 잔류 응력은 피로 수명을 단축시키고 균열 진전 속도를 가속시키는 것으로 알려져 있다. 이러한 이유로

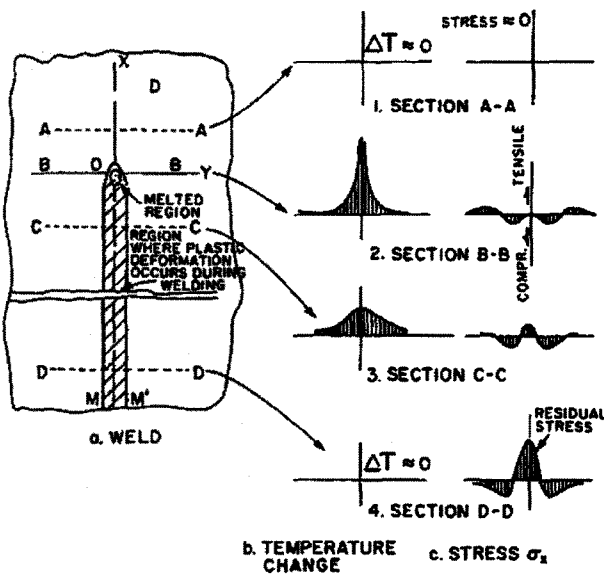


Figure 1 Schematic representation of changes in temperature and stresses during welding

용접 잔류 응력을 예측하기 위한 수치해석 기법과 실험 기법이 필요하다.

Figure 1은 용접 시 온도가 변화함에 따라 생성되는 응력 분포의 설명하는 대표적인 그림이다. 용접 잔류 응력의 발생과정을 3차원 형상을 이용하여 설명하기는 어렵기 때문에 이상화된 Three-bar(3봉) 모델을 많이 이용한다.

Figure 2은 전형적인 Three-bar 모델을 보인 것이며 잔류 응력의 정량적인 특징은 파악하기 어렵지만, 정성적인 특징은 설명할 수 있다. 그러나 Three-bar의 물리적 특징만을 이용하여 잔류 응력의 발생과정을 설명하기는 쉽지 않다.

본 기사는 Three-Bar를 이용하여 용접 잔류 응력 생성 과정을 정성적으로 설명하기 위한 계산 코드와 유한요소해석 코드사례를 소개하고자 한다. 계산 코드는 C++로 구성되었으며, 유한요소해석 코드는 ANSYS를 이용한 것이다.

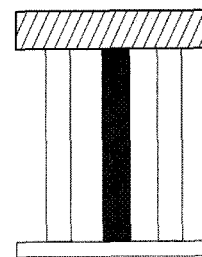


Figure 2 Three bar model for elasto-plastic process

2. 잔류 응력 발생과정

중앙의 봉은 입열에 의하여 가열과 냉각의 과정을 거치면 양쪽의 측면봉은 중앙봉의 변형을 지지하는 강성체의 역할을 한다. 중앙부재는 온도가 상승하면 측면부재의 영향으로 인하여 압축응력을 받으며 온도가 비교적 낮을 때는 탄성변형을 한다. 그러나 이 압축응력은 온도가 상승함에 따라 항복응력에 도달하게 된다. 이때 변형량은 적합조건을 만족하는 값으로부터 결정된다. 최고 온도에 도달한 후 냉각하는 과정에서 중앙부재는 수축을 하게 되지만 양측 부재의 강성으로 인하여 인장응력의 영향을 받게 된다. 온도가 낮아짐에 따라 중앙부재는 다시 인장에 의한 응력이 소성응력에 도달하게 되어 소성 상태에 접어들어 최종적으로 잔류응력이 발생한다. 온도에 따른 중앙봉의 응력은 Figure 3에 보인 것과 같이 가열과 냉각과정을 거친다.

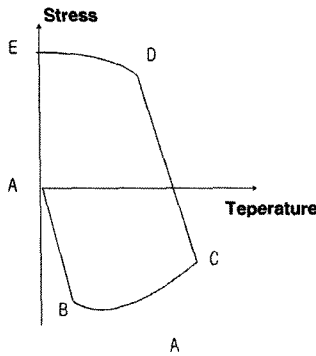


Figure 3 History of the stress in the center bar

2.1 잔류 응력 발생의 정식화

용접에 의한 응력 발생과정은 열탄소성 변형으로 파악할 수 있다. 3봉 구속 모델을 이용하여 열탄소성 현상의 역학적 특성을 설명하기 위하여 Figure 3의 응력변화를 각각 A, B, C, D, E점으로 분리하여 각 구간에서 응력 분포를 파악하면 잔류 응력의 발생과정을 설명할 수 있다. 연속체 역학에 바탕을 두고 각 구간에서 평형 방정식, 적합 방정식, 재료 구성 방정식을 만족하는 해를 구함으로써 잔류 응력 생성 과정을 설명하였다.

정식화를 위한 부호는 아래와 같이 정의한다.

• Nomenclature

T	Temperature [°C]
A _m	Area of the center bar
E _m (T)	Elastic coefficient of the center bar

σ _y (T)	Yield stress of the center bar
α	Thermal expansion coefficient
A _s	Area of the side bar
E _s	Elastic coefficient of the side bar
ε	Total strain
ε _m	Total strain of the center bar
ε _m ^e	Elastic strain of the center bar
ε _m ^p	Plastic strain of the center bar
ε _m th	Thermal strain of the center bar
A _ε	Strain at point A
ε _s	Total strain of the side bar
σ _m	Yield stress of the center bar
σ _s	Yield stress of the center bar

Assumption : The side bar is in the elastic state during all the process

2.1.1 Elastic state during heating (\overline{AB})

3개의 부재가 받는 힘은 평형을 이루므로 아래와 같은 평형 방정식을 만족해야 한다. 그리고 구성 방정식 및 변형의 적합조건을 정리하면 다음과 같다.

Equilibrium equation:

$$\sigma_m A_m + 2\sigma_s A_s = 0 \tag{1}$$

Constitutive equation

$$\sigma_s = E_s \epsilon_s \tag{2}$$

$$\begin{aligned} \sigma_m &= E_m (\epsilon_m - \epsilon^{th}) \\ &= E_m (\epsilon_m - \alpha T) \end{aligned} \tag{3}$$

Compatibility equation

$$\epsilon_m = \epsilon_s \tag{4}$$

식 (3)과 식 (4)를 식 (1)에 대입하면 다음과 같이 변형율을 구할 수 있다.

$$\begin{aligned} E_m (\epsilon - \alpha T) A_m + 2E_s \cdot \epsilon \cdot A_s &= 0 \\ \epsilon_m = \epsilon_s = \epsilon &= \frac{E_m(T) \cdot A_m \cdot \alpha(T)}{E_m(T) A_m + 2E_s A_s} \cdot T \end{aligned} \tag{5}$$

따라서 각 부재에 부과되는 응력은 다음과 같이 온도의 함수로 나타낼 수 있다.

$$\sigma_s = E_s \cdot \epsilon = \left[\frac{E_s \cdot E_m(T) A_m \alpha(T)}{E_m(T) A_m + 2E_s A_s} \right] \cdot T \tag{6}$$

$$\begin{aligned} \alpha_m &= E_m(T) \cdot \left[\frac{E_m(T)A_m\alpha(T)}{E_m(T)A_m + 2E_sA_s} - \alpha(T) \right] \cdot T \quad (7) \\ &= E_m(T) \cdot \left[\frac{-E_sA_s}{E_m(T)A_m + 2E_sA_s} \right] \alpha(T) \cdot T \end{aligned}$$

2.1.2 Plastic state during heating(\overline{BC})

온도가 계속 상승하면 어느 온도(B점)에서 중앙부재의 응력이 항복응력과 같아지므로 중앙부재의 응력은 $\sigma_Y(T)$ 로 일정하게 되며, 중앙부재의 탄성변형률 ϵ_m^e 은 변화가 없게 된다. 즉,

$$\sigma_m = \sigma_Y(T) \quad (8)$$

Equilibrium equation

$$\sigma_Y(T)A_m + 2\sigma_sA_s = 0 \quad (9)$$

$$\sigma_s = -\frac{\sigma_Y(T)A_m}{2A_s} \quad (10)$$

측면부재의 응력으로부터 측면 부재의 변형률을 구할 수 있으며 적합조건으로부터 중앙부재의 전체 변형률을 얻을 수 있다.

Compatibility condition

$$\epsilon_s = \epsilon_m = -\frac{\sigma_Y(T)A_m}{2E_sA_s} \quad (11)$$

따라서 각 부재에 걸리는 변형률은 다음과 같다.

Constitutive equation

$$\epsilon_m^{th} = \alpha(T) \cdot T \quad (12)$$

$$\begin{aligned} \epsilon_m &= B\epsilon_m^e + \epsilon_m^P + \epsilon_m^{th} \quad (13) \\ &= B\epsilon_m^e + \epsilon_m^P - \alpha \cdot T \end{aligned}$$

$$\epsilon_m^P = -\frac{\sigma_Y(T) \cdot A_m}{2E_s \cdot A_s} - B\epsilon_m^e - \alpha \cdot T \quad (14)$$

2.1.3 Elastic state during cooling(\overline{CD})

온도가 상승하다가 냉각되면 부재는 다시 탄성상태에 접어든다.

따라서 최고 온도인 지점에서 중앙부재의 변형률이 초기 변형률로써 존재하게 되며, 초기 응력을 가진 탄성상태가 된다.

평형방정식과 적합조건은 마찬가지로 만족해야 한다.

Equilibrium equation

$$\sigma_m A_m + 2\sigma_s A_s = 0 \quad (15)$$

Compatibility condition

$$\epsilon_m = \epsilon_s \quad (16)$$

열변형률의 성분은 다음과 같이 구할 수 있으며, 중앙부재의 응력은 점C에서 응력값을 초기 응력으로 갖는 값이 된다.

Constitutive equation

$$\begin{aligned} \epsilon_m^{th} &= \alpha(T) \cdot T \quad (17) \\ \sigma_m &= C \sigma_m - \frac{E_s E_m A_m}{E_m A_m + 2E_s} (\epsilon_m^{th} - C \epsilon_m^{th}) \end{aligned}$$

중앙부재의 응력이 구해지면 식 (17)의 응력값을 평형방정식에 대입하면, 측면부재의 응력을 구할 수 있다.

$$\sigma_s = \frac{\sigma_m A_m}{-2A_s} \quad (18)$$

식 (18)으로부터 각 부재의 전체 변형률을 다음과 같이 얻는다.

$$\epsilon_s = \frac{\sigma_s}{E_s} \quad (19)$$

$$\epsilon_m = \epsilon_s = \frac{\sigma_s}{E_s} \quad (20)$$

$$\epsilon_m^P = C \epsilon_m^P \quad (21)$$

$$\epsilon_m^e = \frac{\sigma_m}{E_m} \quad (22)$$

2.1.4 Plastic state during cooling(\overline{DE})

냉각이 계속 되면 중앙부재는 인장응력이 증가한다. 이 인장 응력은 점 D에서 소성응력에 도달하게 되어 소성 상태에 놓인다. 따라서 중앙부재의 응력은 온도에 의하여 결정되는 소성응력을 받게 된다. 즉,

$$\sigma_m = \sigma_Y(T) \quad (23)$$

평형 방정식으로부터 측면 부재의 응력을 구할 수 있다.

Equilibrium equation

$$\sigma_s = \frac{\sigma_m \cdot A_m}{-2A_s} \quad (24)$$

측면과 중앙 부재의 응력이 구해지므로 식 (24)에서 구한 응력으로부터 측면부재의 변형률을 구한다.

Constitutive equation

$$\epsilon_s = \frac{\sigma_s}{E_s} \quad (25)$$

적합조건으로부터 중앙부재의 전체 변형률은 측면부재의 변형률과 같고, 중앙부재의 탄성 변형률, 소성 변형률을 구할 수 있다.

Compatibility condition

$$\epsilon_m = \epsilon_s = \frac{\sigma_s}{E_s} \quad (26)$$

$$\epsilon_m^e = D \epsilon_m^e \quad (27)$$

$$\epsilon_m^p = \epsilon_m - \epsilon_m^e - (\epsilon_m^{th} - C \epsilon_m^{th}) \quad (28)$$

2.2 Material properties

온도의 변화에 따라 탄성계수, 소성응력, 열팽창률, 가공 경화지수가 변한다. 이때, 소성변형 특성은 완전 탄소성 (elastic perfectly plastic material)으로 가정하여 가공 경화의 영향은 고려하지 않았다. 온도변화에 따른 각 재료 특성 값을 다음의 Table 1에 나타내었다. 재료 특성 값은 기존 논문[1]의 것을 참고하였다.

Table 1 Material properties

온도[°C]	탄성계수	항복응력	열팽창 계수
20	206	247	1.160E-05
100	201	240	1.220E-05
200	198	232	1.270E-05
300	191	217	1.310E-05
400	181	201	1.350E-05
500	175	185	1.390E-05
600	163	145	1.440E-05
700	121	87.7	1.490E-05
800	80.3	44.5	1.260E-05
900	39.2	12.3	1.240E-05
1000	19.6	3.92	1.340E-05
1100	17	2.82	1.420E-05
1200	15	2.49	1.480E-05
1300	13.7	2.27	1.540E-05
1400	13.4	2.22	1.610E-05
1500	13.2	2.18	1.660E-05
1600	11.1	1.84	2.160E-05

2.3 계산 결과

중앙봉의 면적이 $A_m=0.01[m^2]$ 이고, 양쪽 측면 봉의 면적이 각각 $A_s=0.04[m^2]$ 이고, 측면부재는 온도의 영향을 받지 않는다고 가정하여 계산을 수행하였다.

해석적인 방법과 FEA(Finite Element Analysis)를 이용한 계산결과를 각각 비교하면 다음과 같다.

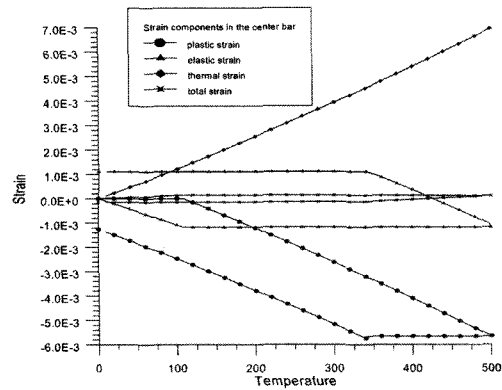


Figure 4 Strain components of the center bar by present work

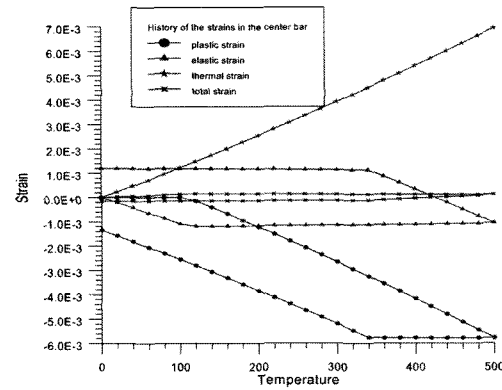


Figure 5 Strain components in the center bar by FEA

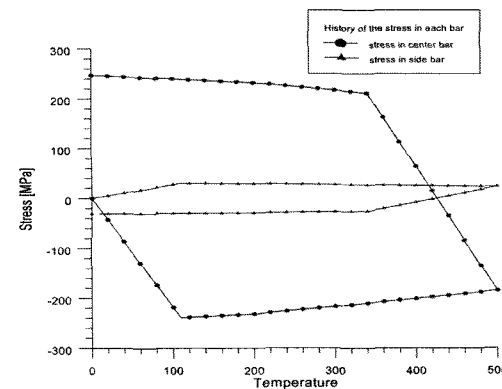


Figure 6 Stress in each bar by present work

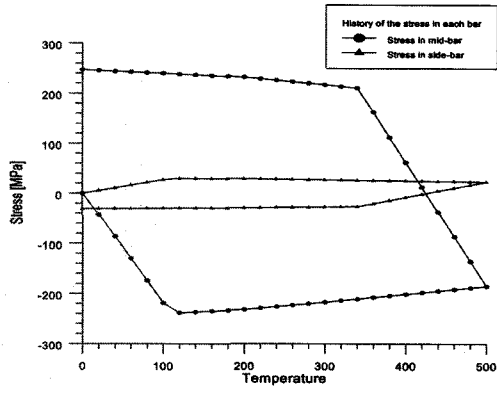


Figure 7 Stress in each bar by FEA

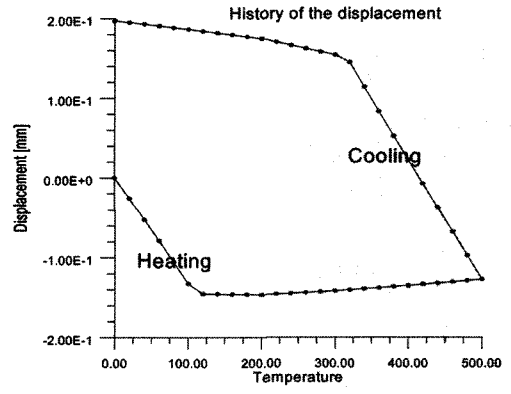


Figure 9 History of the displacement of the center bar

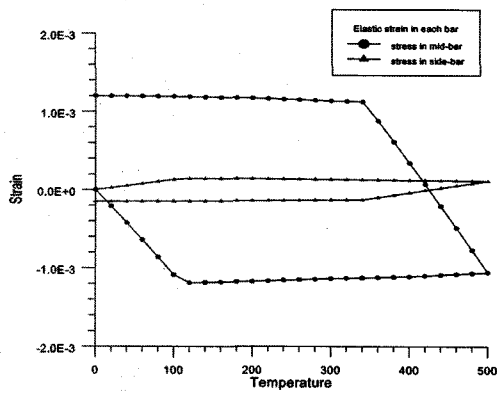



Figure 8 Elastic strain in each element

참고 문헌

1. 이동욱, 유한요소법을 이용한 GMAW의 열탄소성 모델링, 서울대학교 박사학위 논문, 1995 

[담당 : 이장현, 편집위원]

부록 자료1 : 해석 코드

아래는 계산을 위한 코드이며, C++를 이용한 것이다. 적절한 컴파일러를 선택하여 실행하면 결과를 얻을 수 있다.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <iostream.h>
#include <conio.h>
#include <math.h>
#include <iomanip.h>
#include "mp_ex.h"
#include "mp_alpx.h"
#include "mp_yield.h"

int main(int argc, char **argv)
{
    FILE *o1,*o2,*o3;
    o1=fopen("mp_ex.dat","wt");
    o2=fopen("mp_yield.dat","wt");
    o3=fopen("mp_alpx.dat","wt");

    FILE *mid,*side;
    mid=fopen("mid.dat","wt");
    side=fopen("side.dat","wt");

    FILE *eptoc,*stress;
    eptoc=fopen("eptoc.dat","wt");
    stress=fopen("stress.dat","wt");

    double Tp,E,Y,alpha;
    int i;

    int l_flag;          //1=heating , 0=cooling
    int y_flag; //1=yielding, 0=elastic
    double SY;          //yield stress at specified temp.

    double T[501];
    double T_max;

    double eptom[501]; //total strain in center
    double eplm[501]; //plastic strain in center
    double eplm_max=0.0;
    double epthm[501]; //thermal strain in center
    double epthm_max=0.0;
    double epelem[501]; //elastic strain in center
    double epelem_max=0.0;
    double sm[501];    //stress in center

    double eptos[501]; //total strain in side
    double ss[501];    //stress in side

    double Am;        //area in center
    double As;        //area in side
    double Em;        //elastic in center at specified temp.
    double Es;        //elastic in side
    double st;

    printf("\n- 3Bar simulation starts from here -\n");
    Es=mp_ex(0.0);
    printf("\n Area of the Side bar? : ");
    scanf("%lf",&As);
    printf("\t As = : %10.5f",As);
    printf("\n Area of the Center bar? : ");
    scanf("%lf",&Am);
    printf("\t Am = : %10.5f",Am);
    printf("\n Maximum Temperature? : ");
    scanf("%lf",&T_max);
    printf("\t T_max = : %10.5f",T_max);

    for(i=0;i<501;i++){
        double dT=T_max/250;
        if(i<251)
```

```

        T[i]=i*dT;
    else
        T[i]=1.0*dT*(i-250)+T_max;
    printf("\n T[%5d] = %10.5f",i,T[i]);
}

//Temperature assignment
for(i=0;i<501;i++){
    // Check the loading state
    if(T[i]>T[i-1])
        l_flag=1;
    else
        l_flag=0;

    Em=mp_ex(T[i]);
    alpha=mp_alpx(T[i]);
    SY=mp_yield(T[i]);

    // Check the Yield state
    if(l_flag){
        st=Em*(-2.0*Es*As)/(Em*Am+2.0*Es*As) *alpha*T[i] ;
    }

    if(l_flag && fabs(st)>=SY)
        y_flag=1;

    // loading and elastic state
    if(l_flag && !y_flag){

        sm[i]=Em*(-2.0*Es*As)/(Em*Am+ 2*Es*As) *alpha*T[i];
        ss[i]=sm[i]*Am/(-2.*As);

        eptos[i]=ss[i]/Es;
        eptom[i]=eptos[i];
        epthm[i]=alpha*T[i];
        epplm[i]=0.0;
        epelm[i]=eptos[i]-epthm[i]-epplm[i];

        if(fabs(epplm_max) > fabs(epplm[i])) epplm_max=epplm[i];
        if(fabs(epelm_max) > fabs(epelm[i])) epelm_max=epelm[i];
        if(fabs(epthm_max) > fabs(epthm[i])) epthm_max=epthm[i];
    }

    // Loading and plastic state
    else if(l_flag && y_flag){
        sm[i]= -1.0*SY;
        ss[i]= sm[i]*Am/(-2.*As);

        eptos[i] = ss[i]/Es;

        eptom[i]=eptos[i];
        epthm[i]=alpha*T[i];
        epelm[i]=epelm[i-1];
        epplm[i]=eptom[i]-epelm[i]-epthm[i];

        if(fabs(epplm_max) > fabs(epplm[i])) epplm_max=epplm[i];
        if(fabs(epelm_max) > fabs(epelm[i])) epelm_max=epelm[i];
        if(fabs(epthm_max) > fabs(epthm[i])) epthm_max=epthm[i];
    }

    // Unloading and elastic state
    else if(!l_flag){

        epthm[i]=alpha*T[i];

        st=sm[i-1]
            +Em*(2.*Es*As)/(Em*Am+ 2*Es*As)*fabs(epthm[i]-epthm[i-1]);

        if(fabs(st) >= SY)
            y_flag=1;
        else
            y_flag=0;
    }
}

```

```

        if(!y_flag){
            sm[i]=sm[i-1]
                +Em*(2.*Es*As)/(Em*Am+ 2*Es*As)*fabs(epthm[i]-epthm[i-1]);
            ss[i]=sm[i]*Am/(-2.*As);
            eptos[i]=ss[i]/Es;
            eptom[i]=eptos[i];
            eplm[i]=eplm[i-1];
            //epelm[i]=epelm[i-1]+eptom[i]-eplm[i]-epthm[i];
            epelm[i]=sm[i]/Em;
        }
        else{
            sm[i]=SY;
            ss[i]=sm[i]*Am/(-2.*As);
            eptos[i]=ss[i]/Es;
            eptom[i]=eptos[i];
            epelm[i]=epelm[i-1];
            eplm[i]=eptom[i]-epelm[i]-fabs(epthm[i]-epthm_max);
        }
    }
    else{
        printf("\n Error in the loading state! \n");
        getch();
    }
}

for(i=0;i<501;i++){
    fprintf(stress,"%10.5f %15.8e %15.8e \n",T[i],sm[i]/1.e6,ss[i]/1.e6);
    //printf("%10.5f %15.8e %15.8e \n",T[i],sm[i]/1.e6,ss[i]/1.e6);
    fprintf(eptoc,"%10.5f %15.8e %15.8e %15.8e %15.8e \n",
        T[i],eplm[i],epelm[i],epthm[i],eptom[i]);
    //printf("%10.5f %15.8e %15.8e \n",T[i],sm[i]/1.e6,ss[i]/1.e6);
}

/*
// Print out the material coefficients for certainty
for(i=0;i<33;i++){
    Tp = 0.5*i*100.0;
    E = mp_ex(Tp);
    //printf("\n temp = %10.5f \t Elasticity=%15.5f",Tp,E/1.e9);
    fprintf(o1,"%10.5e %15.5e\n",Tp,E/1.e9);
}

for(i=0;i<33;i++){
    Tp = 0.5*i*100.0;
    Y = mp_yield(Tp);
    //printf("\n temp = %10.5f \t Yield=%15.5e",Tp,Y/1.e6);
    fprintf(o2,"%10.5e %15.5e\n",Tp,Y/1.e6);
}

for(i=0;i<33;i++){
    Tp = 0.5*i*100.0;
    alpha = mp_alpx(Tp);
    //printf("\n temp = %10.5f \t thermal exp=%15.8e",Tp,alpha);
    fprintf(o3,"%10.5e %15.8e\n",Tp,alpha);
}
*/

printf("\n—— End of Line ——\n");
getch();
fclose(o1);
fclose(o2);
fclose(o3);
fclose(mid);
fclose(side);
fclose(stress);
fclose(eptoc);
return 0;
}
//End of Code

```


"mp_ex.h"

```
double mp_ex(double x)
{
    double E;
    int i,j;

    double temp[17];
    double e[17];

    //Temperature assignment
    for(i=0;i<17;i++){
        temp[i]=i*100.0;
    }

    // Elastic coefficient assignment
    e[0] =2.06e11;
    e[1] =2.01e11;
    e[2] =1.98e11;
    e[3] =1.91e11;
    e[4] =1.81e11;
    e[5] =1.75e11;
    e[6] =1.63e11;
    e[7] =1.21e11;
    e[8] =8.03e10;
    e[9] =3.92e10;
    e[10]=1.96e10;
    e[11]=1.70e10;
    e[12]=1.50e10;
    e[13]=1.37e10;
    e[14]=1.34e10;
    e[15]=1.32e10;
    e[16]=1.11e10;

    // Check the specified temperature
    j=-1;
    for(i=0;i<16;i++){
        if(x >= temp[i]){
            j=i;
        }
    }
    if(j == -1 ){
        printf("\n Warning : Temperature is out of range! \n");
        getch();
    }
    // Young's modulus calculation
    E = e[j]+(x-temp[j])*(e[j+1]-e[j])/(temp[j+1]-temp[j]);
    return(E);
}
```

"mp_alpx.h"

```

double mp_alpx(double x)
{
    double alpha;
    int i, j;

    double temp[17];
    double e[17];

    //Temperature assignment
    for(i=0;i<17;i++){
        temp[i]=i*100.0;
    }

    // Thermal expansion assignment
    e[0] =0.0000116;
    e[1] =0.0000122;
    e[2] =0.0000127 ;
    e[3] =0.0000131;
    e[4] =0.0000135;
    e[5] =0.0000139;
    e[6] =0.0000144;
    e[7] =0.0000149;
    e[8] =0.0000126;
    e[9] =0.0000124;
    e[10]=0.0000134;
    e[11]=0.0000142;
    e[12]=0.0000148;
    e[13]=0.0000154;
    e[14]=0.0000161;
    e[15]=0.0000166;
    e[16]=0.0000216;

    // Check the specified temperature
    j=-1;

    for(i=0;i<16;i++){
        if(x >= temp[i]){
            j=i;
        }
    }
    if(j == -1 ){
        printf("\n Warning : Temperature is out of range! \n");
        getch();
    }
    // Thermal expansion calculation
    alpha = e[j]+(x-temp[j])*(e[j+1]-e[j])/(temp[j+1]-temp[j]);
    return(alpha);
}

```

"mp_yield.h"

```
double mp_yield(double x)
{
    double Yield;
    int i,j;

    double temp[17];
    double e[17];

    //Temperature assignment
    for(i=0;i<17;i++){
        temp[i]=i*100.0;
    }

    // Yield stress assignment
    e[0] =2.47E+8;
    e[1] =2.40E+8;
    e[2] =2.32E+8;
    e[3] =2.17E+8;
    e[4] =2.01E+8;
    e[5] =1.85E+8;
    e[6] =1.45E+8;
    e[7] =8.77E+7;
    e[8] =4.45E+7;
    e[9] =1.23E+7;
    e[10]=3920000.0;
    e[11]=2820000.0;
    e[12]=2490000.0;
    e[13]=2270000.0;
    e[14]=2220000.0;
    e[15]=2180000.0;
    e[16]=1840000.0;

    // Check the specified temperature
    j=-1;
    for(i=0;i<16;i++){
        if(x >= temp[i]){
            j=i;
        }
    }

    if(j == -1 ){
        printf("\n Warning : Temperature is out of range! \n");
        getch();
    }

    // Yield stress calculation
    Yield = e[j]+(x-temp[j])*(e[j+1]-e[j])/(temp[j+1]-temp[j]);

    return(Yield);
}
```

부록 2 : 유한요소해석

아래는 유한요소해석 코드이며, ANSYS의 입력 파일이다.

```

!
! ***** [ 1 ] Preprocess *****
!
/title,Three Bar Model for welding analysis
/filename,threebar

/prep7
et,1,link1
mpTEMP, 1, 0, 100, 200, 300, 400, 500 ! actually from 0
mpTEMP, 7, 600, 700, 800, 900,1000,1100
mpTEMP,13,1200,1300,1400,1500,1600
!----- Young's Modulus -----
mpDATA,EX,1, 1,206e9,201e9,198e9,191e9,181e9,175e9
mpDATA,EX,1, 7,163e9,121e9,80.3e9,39.2e9,19.6e9,17.0e9
mpDATA,EX,1,13,15.0e9,13.7e9,13.4e9,13.2e9,11.1e9
!----- Thermal Expansion Coefficient -----
mpDATA,ALPX,1, 1,11.6e-6,12.2e-6,12.7e-6,13.1e-6,13.5e-6,13.9e-6
mpDATA,ALPX,1, 7,14.4e-6,14.9e-6,12.6e-6,12.4e-6,13.4e-6,14.2e-6
mpDATA,ALPX,1,13,14.8e-6,15.4e-6,16.1e-6,16.6e-6,21.6e-6
!----- Bilinear Isotropic Yield Stress -----
tb,bkin,1,6
tbTEMP, 0 $tbDATA,1,247.0e6,7.93e9 ! actually zero temp.
tbTEMP, 100 $tbDATA,1,240.0e6,7.79e9
tbTEMP, 200 $tbDATA,1,232.0e6,7.61e9
tbTEMP, 300 $tbDATA,1,217.0e6,6.20e9
tbTEMP, 400 $tbDATA,1,201.0e6,5.15e9
tbTEMP, 500 $tbDATA,1,185.0e6,4.23e9

!/view,1,0,0,0
tbplot,bkin,1

tref,0 ! Reference temperature
r,1,0.01
r,2,0.04
/pnum,node,0 $/pnum,elem,0
!
! Node & Element Generation
n,1,-1 $n,2,0 $n,3,1
ngen,2,3,all,,, -1

type,1 $real,1 $e,2,5
real,2 $e,1,4 $e,3,6
! B.C.
nSEL,S,LOC,Y,0
d,all,all,0
nall
! Coupling condition
cp,1,uy,5,4,6

/psbc,all,1

nplot
eplot

! Material properties
mplist,all,,,ex
mplist,all,,,alpx

```

```

tblast,bkin,1

! Material properties
/output,young,dat
mplist,all,,ex
/output,thermexp,dat
mplist,all,,alpx
/output,bkin,dat
tblast,bkin,1

finish
!
!
! ***** [ 2 ] S o l u t i o n *****
!
/SOLUTION
antype,static

/plot,all,1 $plot $/plot,all,0
!
! (2) Solve the Deformation
!
pstres,on      ! Prestress effect on
nlgeom,on      ! Large doformation effects
nropt,full,,   ! FULL mode is used if Nonlinearities are present
pred,on        ! Predictor on

t1=25
t2=50
Tmax=500
outres,all,all
outpr,all,all
*DO,J,0,t2
  lnsrch,on
  kbc,1        ! Steped load function
  time,j       ! Time at the end of the load step
  autots,on    ! Auto time stepping

      *if,j,le,t1,then
          tmpvar= j*Tmax/t1
          bfe,1,temp,,tmpvar
      *elseif,j,le,t2,then
          tmpvar= -1*Tmax*(j-t1)/(t2-t1) + Tmax
          bfe,1,temp,,tmpvar
      *endif
  lswrite
  solve
*ENDDO

finish

/post1
! reaction force
prrsol,f

! element solution with all data
/output,residual,dat
presol,elem
/output,reactf,dat
presol,force

```

```

finish

/post26
! Time history of axial stress
esol,2,1,,ls,1,sig
esol,3,2,,ls,1,sig
esol,4,3,,ls,1,sig
plvar,2,3,4
/output, stress, dat $prvar,2,3,4

! Time history of the plastic strain in the middle bar
esol,5,1,,leppl,1 $plvar,5
/output, epplc, dat $prvar,5

! Time history of the elastic strain in the middle bar
esol,6,1,,lepel,1 $plvar,6
/output, epelc, dat $prvar,6

! Time history of the elastic strain in the side bar
esol,7,2,,lepel,1 $plvar,7
/output, epels, dat $prvar,7

/output, epelcs, dat $plvar,6,7
! Time history of the thermal strain in the middle bar
esol,8,1,,lepth,1 $plvar,8
/output, epthc, dat $prvar,8

plvar,5,6,8

/output, eptoc, dat $prvar,5,6,8
nsol,9,5,u,y $plvar,9
/output, dispc, dat $prvar,9
/output,
Finish

```