รายการอ้างอิง

ภาษาไทย

ปราโมทย์ เดชะอำไพ. ระเบียบวิธีเชิงตัวเลขในงานวิศวกรรม. พิมพ์ครั้งที่ 1. กรุงเทพมหานคร : โรงพิมพ์จุฬาลงกรณ์มหาวิทยาลัย, 2538.

ทรงพันธ์ วรรณมาศ. เครื่องปั้นดินเผา. กรุงเทพมหานคร: หน่วยศึกษานิเทศน์ กรมการฝึกหัดครู, 2530.

ทวี พรหมพฤกษ์. เตาเผาและการเผา. เอกสารนิเทศน์การศึกษาฉบับที่ 245. กรุงเทพมหานคร : หน่วยศึกษานิเทศน์ กรมการฝึกหัดครู, 2525.

นพดล วิลิฐศรีศักดิ์. การวิเคราะห์การใช้พลังงานในเตาเผาเซรามิคประเภทเตาแก๊ส. โครงงานพิเศษ สาขาเคมีอุตสาหกรรม คณะวิทยาศาสตร์ มหาวิทยาลัยเชียงใหม่, 2535.

สมศรี จงรุ่งเรือง. ระเบียบวิธีวิเคราะห์การถ่ายเทความร้อน. พิมพ์ครั้งที่ 1.. กรุงเทพมหานคร : โรงพิมพ์จุฬาลงกรณ์มหาวิทยาลัย, 2542.

สุภาภรณ์ ขวัญยืน. การศึกษาหัวพ่นไฟและเทคนิคการเผาเตาไฟเบอร์. โครงงานพิเศษ สาขาเคมีอุตสาหกรรม คณะวิทยาศาสตร์ มหาวิทยาลัยเชียงใหม่, 2538.

อรนุช ฟองสมุทร. การวิเคราะห์การใช้พลังงานในเตาเผาเซรามิคประเภทเตาแก๊ส. โครงงานพิเศษ สาขาเคมีอุตสาหกรรม คณะวิทยาศาสตร์ มหาวิทยาลัยเชียงใหม่, 2534.

ภาษาอังกฤษ

Adams,J.A. and Rogers,D.F. Computer-Aided Heat Transfer Analysis. New York : McGraw-Hill, 1973.

De monte F. Transient heat conduction in one-dimensional composite slab. A 'natural' analytic approach. International Journal of Heat and Mass Transfer 2000 : 3607-3619.

Incropera,F. P. and Dewitt D. P.. Fundamentals of Heat and Mass Transfer.. 4th ed.. New York : Wiley, 1996.

Nyhoff,L. and Leestma,S. Fortran 90 for Engineers and Scientists. New Jersey : Prentice-Hall International, 1997.

Patankar,S.V.. Numerical transfer and fluid flow. Series in computational method in mechanics and thermal sciences. U.S.A : Hemishere Publishing, 1980.

Reed, J.R. North American combustion handbook.. 2nd ed.. Cleveland : North American Manufacturing Company,1978.

ภาคผนวก

ภาคผนวก ก
โปรแกรมคอมพิวเตอร์

```
PROGRAM THESIS04
USE PORTLIB
        REAL    TEMPT(621,841),TEMPT_OLD(621,841),TEMP_S01(5,400),TEMP_T01(5,400),ENERGY_TS(400),&
                ENERGY_CF(400),ENERGYOUT(400),TEMPB(231,841),TEMPB_OLD(231,841),TEMP_B01(5,400),
                TTB(231,400)
        INTEGER  TIME_IN01,TIME_OUT01,N_TEST
        CHARACTER*8     TIME_IN,TIME_OUT
        CHARACTER*20    SUB_FILE(400),TIME_FILE

        OPEN(UNIT=15,FILE='TEST BOUNDARY.TXT',STATUS='UNKNOWN')

        CALL TIME(TIME_IN)
        WRITE(*,*) TIME_IN

        I_STEP=5

        J_STEP=I_STEP

        WRITE(*,*) 'HOW MANY NUMBER OF TEST?'
        READ(*,*) N_TEST

        WRITE(*,*) 'HOW MANY TIME TO CALCULATE'
        READ(*,22) K_STEP

        IF(N_TEST==7) THEN
                K_BURN=197
        ELSE IF(N_TEST==8) THEN
                K_BURN=199
        ELSE IF(N_TEST==9) THEN
                K_BURN=229
        ELSE IF(N_TEST==11) THEN
                K_BURN=238
        ELSE IF(N_TEST==12) THEN
                K_BURN=217
        ELSE IF(N_TEST==13) THEN
                K_BURN=343
        END IF

        TIME_IN01=TIME()
        CALL ENERGY_INOUT(ENERGYOUT,N_TEST,K_BURN,K_STEP)
        CALL CALCULATE_ENERGYSTORE_CERAMIC_AND_FERNITURE(K_BURN,ENERGY_CF,N_TEST)
```

```
DO K=1,K_BURN,K_STEP
        IF(K==1) THEN
                CALL BOUNDARY_EDGE(TEMPT,I_STEP,J_STEP,1,N_TEST)
                CALL NAME_OF_FILE(SUB_FILE,1)
                        TIME_FILE=SUB_FILE(1)
                CALL WRITE_FILE(TEMPT,I_STEP,J_STEP,TIME_FILE)
        ELSE IF(K>=2.AND.K<=K_BURN) THEN
                CALL BOUNDARY_EDGE(TEMPT,I_STEP,J_STEP,K,N_TEST)
                CALL NAME_OF_FILE(SUB_FILE,K-K_STEP)
                TIME_FILE=SUB_FILE(K-K_STEP)
                CALL READ_FILE(TEMPT_OLD,I_STEP,J_STEP,TIME_FILE)
                CALL CALCULATE_TEMP(TEMPT,TEMPT_OLD,I_STEP,J_STEP,K_STEP)
                CALL NAME_OF_FILE(SUB_FILE,K)
                TIME_FILE=SUB_FILE(K)
                CALL WRITE_FILE(TEMPT,I_STEP,J_STEP,TIME_FILE)
        END IF
END DO
        CALL E_BACK_WALL
        CALL CALCULATE_ENERGY_STORE(I_STEP,J_STEP,K_STEP,K_BURN,ENERGY_TS)
        CALL READ_FILE_SIDE01(TEMP_S01,K_BURN,I_STEP,J_STEP,K_STEP)
        CALL READ_FILE_TOP01(TEMP_T01,K_BURN,I_STEP,J_STEP,K_STEP)
        CALL READ_FILE_BACK01(TEMP_B01,K_BURN,I_STEP,J_STEP,K_STEP)
        OPEN(UNIT=8,FILE='SIDE_POINT.TXT',STATUS='UNKNOWN')
        OPEN(UNIT=9,FILE='TOP_POINT.TXT',STATUS='UNKNOWN')
        OPEN(UNIT=12,FILE='BACK_POINT.TXT',STATUS='UNKNOWN')

DO K=1,K_BURN,6
        WRITE( 8,20)
        WRITE( 9,20)
        WRITE(12,20)
        DO I=1,5
                WRITE(8,21) TEMP_S01(I,K)
                WRITE(9,21) TEMP_T01(I,K)
                WRITE(12,21) TEMP_B01(I,K)
        END DO
END DO
        CLOSE(8)
        CLOSE(9)
        CLOSE(12)

TIME_OUT01=TIME()
```

```
        WRITE(*,23) TIME_OUT01-TIME_IN01
        CALL TIME(TIME_OUT)
        WRITE(*,*) TIME_OUT
20      FORMAT(/)
21      FORMAT(2X,F8.4,\)
22      FORMAT(I5)
23      FORMAT(2X,'TOTAL TIME',I4,3X,'S')
24      FORMAT(2X,F12.4)
26      FORMAT(4(2X,F12.8))
END PROGRAM THESIS04


SUBROUTINE BOUNDARY_EDGE(TEMPB,I_STEP,J_STEP,K_BOUNDARY,N_TEST)
        REAL     TEMPB(621,841),T_KILN,T_TOP,DUMMY01
        INTEGER,INTENT(IN)::        I_STEP,J_STEP,K_BOUNDARY,N_TEST
        IF(K_BOUNDARY==1) THEN
                CALL TEMP_IN_KILN(K_BOUNDARY,T_KILN,T_TOP,DUMMY01,N_TEST)
                WRITE(*,*) 'HOW MANY BOUNDARY TEMPERATURE?'
                READ (*,*) TEMP_BOUNDARY

                DO I=1,621,I_STEP
                        TEMPB(I,  1)=34 !T_SURFACE(K_BOUNDARY)
                        IF(I>1 .AND. I<231) THEN
                                TEMPB(I,JEND(I))=  TEMP_BOUNDARY !T_INITIAL
                        ELSE IF(I>=231 .AND. I<=621) THEN
                                TEMPB(I,JEND(I))=T_KILN
                        END IF
                END DO

                DO J=1,841,J_STEP
                        TEMPB(  1,J)=34 !T_SURFACE(K_BOUNDARY)

                        IF(J>1 .AND. J<231) THEN
                                TEMPB(IEND(J),J)= TEMP_BOUNDARY !T_INITIAL
                        ELSE IF(J>231 .AND. J<=841) THEN
                                TEMPB(IEND(J),J)=T_KILN !T_KILN
                        END IF
                END DO

                DO I=1+I_STEP,621-I_STEP,I_STEP
                        DO J=1+J_STEP,JEND(I)-J_STEP,J_STEP
                                TEMPB(I,J)=TEMP_BOUNDARY
```

```
                END DO
            END DO
        ELSE IF(K_BOUNDARY>=2) THEN
            CALL TEMP_IN_KILN(K_BOUNDARY,T_KILN,T_TOP,DUMMY01,N_TEST)
            !T_KILN=TEMP_OF_KILN(K_BOUNDARY)
            !T_TOP=TEMP_OF_TOP(T_KILN)
            !WRITE(*,*) T_KILN,T_TOP
            DO I=1,621,I_STEP
                TEMPB(I,  1)=34 !T_SURFACE(K_BOUNDARY)
                IF(I>=231 .AND. I<=621) THEN
                    TEMPB(I,JEND(I))=T_TOP
                END IF
            END DO

            DO J=1,841,J_STEP

                TEMPB(  1,J)=34 !T_SURFACE(K_BOUNDARY)
                IF(J>231 .AND. J<=841) THEN
                    TEMPB(IEND(J),J)=T_TOP
                END IF
            END DO
        END IF

END SUBROUTINE BOUNDARY_EDGE



SUBROUTINE WRITE_FILE(TEMPW,I_STEP,J_STEP,TIME_FILE)
        CHARACTER*20 TIME_FILE
        REAL TEMPW(621,841)
        INTEGER I_STEP,J_STEP
        !CALL NAME_OF_FILE(SUB_FILE,1)

        !TIME_FILE=SUB_FILE(1)

        OPEN(UNIT=7,FILE=TIME_FILE,STATUS='UNKNOWN')
        DO J=1,841,J_STEP
            WRITE(7,20)
            DO I=1,621,I_STEP
                WRITE(7,21) TEMPW(I,J)
            END DO
        END DO
```

```
        CLOSE(7)
20      FORMAT(/)
21      FORMAT(2X,F8.4,\)
END SUBROUTINE WRITE_FILE



SUBROUTINE WRITE_FILE_BACK(TEMPW,I_STEP,J_STEP,TIME_FILE)
        CHARACTER*20 TIME_FILE
        REAL TEMPW(231,841)
        INTEGER I_STEP,J_STEP
        OPEN(UNIT=7,FILE=TIME_FILE,STATUS='UNKNOWN')
        DO J=1,841,J_STEP
                WRITE(7,20)
                DO I=1,231,I_STEP
                        WRITE(7,21) TEMPW(I,J)
                END DO
        END DO
        CLOSE(7)
20      FORMAT(/)
21      FORMAT(2X,F8.4,\)
END SUBROUTINE WRITE_FILE_BACK



SUBROUTINE READ_FILE(TEMPR,I_STEP,J_STEP,TIME_FILE)
        CHARACTER*20 TIME_FILE
        REAL TEMPR(621,841)
        INTEGER I_STEP,J_STEP
        !CALL NAME_OF_FILE(SUB_FILE,1)
        !TIME_FILE=SUB_FILE(1)
        OPEN(UNIT=7,FILE=TIME_FILE,STATUS='UNKNOWN')
        DO J=1,841,J_STEP
                READ (7,20)
                DO I=1,621,I_STEP
                        READ(7,21) TEMPR(I,J)
                END DO
        END DO
        CLOSE(7)
20      FORMAT(/)
21      FORMAT(2X,F8.4,\)
END SUBROUTINE READ_FILE
```

```
SUBROUTINE READ_FILE_BACK(TEMPR,I_STEP,J_STEP,TIME_FILE)
        CHARACTER*20 TIME_FILE
        REAL TEMPR(231,841)
        INTEGER I_STEP,J_STEP
                !TIME_FILE=SUB_FILE(1)
                OPEN(UNIT=7,FILE=TIME_FILE,STATUS='UNKNOWN')
        DO J=1,841,J_STEP
                READ (7,20)
                DO I=1,231,I_STEP
                        READ(7,21) TEMPR(I,J)
                END DO
        END DO
        CLOSE(7)
20      FORMAT(/)
21      FORMAT(2X,F8.4,\)
END SUBROUTINE READ_FILE_BACK



SUBROUTINE READ_FILE_SIDE01(TEMP_S01,K_S01,I_STEP,J_STEP,K_STEP)
        REAL TEMP_S01(5,400),DUMMY_TEMP01
        INTEGER I_STEP,J_STEP,K_STEP
        CHARACTER*20 TIME_FILE,SUB_FILE(400)
        DO K=1,K_S01,K_STEP
                CALL NAME_OF_FILE(SUB_FILE,K)
                TIME_FILE=SUB_FILE(K)
                OPEN(UNIT=8,FILE=TIME_FILE,STATUS='UNKNOWN')
                DO J=1,(551-1)/J_STEP
                        READ(8,20)
                END DO
                DO J=1,(76-1)/I_STEP
                        READ(8,21) DUMMY_TEMP01
                END DO
                        READ(8,21) TEMP_S01(1,K)
                        !READ(8,20)
                DO J=1,(36-1)/I_STEP
                        READ(8,21) DUMMY_TEMP01
                END DO
                        READ(8,21) TEMP_S01(2,K)
                        !READ(8,20)
                DO J=1,(11-1)/I_STEP
                        READ(8,21) DUMMY_TEMP01
```

```
                END DO
                        READ(8,21) TEMP_S01(3,K)
                        !READ(8,20)
                DO J=1,(46-1)/I_STEP
                        READ(8,21) DUMMY_TEMP01
                END DO
                        READ(8,21) TEMP_S01(4,K)
                        !READ(8,20)
                DO J=1,(16-1)/I_STEP
                        READ(8,21) DUMMY_TEMP01
                END DO
                        READ(8,21) TEMP_S01(5,K)
                CLOSE(8)
        END DO
20 FORMAT(/)
21 FORMAT(2X,F8.4,\)
END SUBROUTINE READ_FILE_SIDE01



SUBROUTINE READ_FILE_BACK01(TEMP_B01,K_B01,I_STEP,J_STEP,K_STEP)
        REAL TEMP_B01(5,400),DUMMY_TEMP01
        INTEGER I_STEP,J_STEP,K_STEP
        CHARACTER*20 TIME_FILE,SUB_FILE(400)
        DO K=1,K_B01,K_STEP

                CALL NAME_OF_FILE_BACK(SUB_FILE,K)
                TIME_FILE=SUB_FILE(K)
                OPEN(UNIT=8,FILE=TIME_FILE,STATUS='UNKNOWN')
                DO J=1,(551-1)/J_STEP
                        READ(8,20)
                END DO
                DO J=1,(81-1)/I_STEP
                        READ(8,21) DUMMY_TEMP01
                END DO
                        READ(8,21) TEMP_B01(1,K)
                        !READ(8,20)
                DO J=1,(36-1)/I_STEP
                        READ(8,21) DUMMY_TEMP01
                END DO
                        READ(8,21) TEMP_B01(2,K)
                        !READ(8,20)
```

```
                DO J=1,(26-1)/I_STEP
                        READ(8,21) DUMMY_TEMP01
                END DO
                        READ(8,21) TEMP_B01(3,K)
                        !READ(8,20)
                DO J=1,(26-1)/I_STEP                                    !21
                        READ(8,21) DUMMY_TEMP01
                END DO
                        READ(8,21) TEMP_B01(4,K)
                        !READ(8,20)
                DO J=1,(26-1)/I_STEP
                        READ(8,21) DUMMY_TEMP01
                END DO
                        READ(8,21) TEMP_B01(5,K)
                        !READ(8,20)
                CLOSE(8)
        END DO
20 FORMAT(/)
21 FORMAT(2X,F8.4,\)
END SUBROUTINE READ_FILE_BACK01



SUBROUTINE READ_FILE_TOP01(TEMP_T01,K_T01,I_STEP,J_STEP,K_STEP)
        REAL TEMP_T01(5,400),DUMMY_TEMP01
        CHARACTER*20 TIME_FILE,SUB_FILE(400)
        INTEGER I_STEP,J_STEP,K_STEP
        DO K=1,K_T01,K_STEP

                CALL NAME_OF_FILE(SUB_FILE,K)
                TIME_FILE=SUB_FILE(K)
                OPEN(UNIT=8,FILE=TIME_FILE,STATUS='UNKNOWN')
                DO J=1,(41-1)/J_STEP
                        READ(8,20)
                END DO
                DO J=1,(451-1)/I_STEP
                        READ(8,21) DUMMY_TEMP01
                END DO
                        READ(8,21) TEMP_T01(1,K)
                DO J=1,(41-1)/J_STEP
                        READ(8,20)
```

```
                END DO
                DO J=1,(451-1)/I_STEP
                        READ(8,21) DUMMY_TEMP01
                END DO
                        READ(8,21) TEMP_T01(2,K)
                DO J=1,(41-1)/J_STEP
                        READ(8,20)
                END DO
                DO J=1,(451-1)/I_STEP
                        READ(8,21) DUMMY_TEMP01
                END DO
                  READ(8,21) TEMP_T01(3,K)
                DO J=1,(41-1)/J_STEP
                        READ(8,20)
                END DO
                DO J=1,(451-1)/I_STEP
                        READ(8,21) DUMMY_TEMP01
                END DO
                        READ(8,21) TEMP_T01(4,K)
                DO J=1,(41-1)/J_STEP
                        READ(8,20)
                END DO
                DO J=1,(451-1)/I_STEP
                        READ(8,21) DUMMY_TEMP01
                END DO
                        READ(8,21) TEMP_T01(5,K)
        END DO
20 FORMAT(/)
21 FORMAT(2X,F8.4,\)
END  SUBROUTINE



SUBROUTINE READFILE_TOP_FOR_BACK(TTB,K_BURN,I_STEP,J_STEP,K_STEP)
        REAL TTB(231,400),DUMMY_TEMP01
        CHARACTER*20 TIME_FILE,SUB_FILE(400)
        INTEGER I_STEP,J_STEP,K_STEP
        DO K=1,K_BURN,K_STEP
        CALL NAME_OF_FILE(SUB_FILE,K)
        TIME_FILE=SUB_FILE(K)
        OPEN(UNIT=8,FILE=TIME_FILE,STATUS='UNKNOWN')
        READ(8,20)
```

```
        DO NI=1,231,J_STEP
                DO J=1,(6-1)/J_STEP
                        READ(8,20)
                END DO
                DO I=1,(551-1)/I_STEP
                        READ(8,21) DUMMY_TEMP01
                END DO
                        READ(8,21) TTB(NI,K)
         END DO
        END DO
20 FORMAT(/)
21 FORMAT(2X,F8.4,\)
END SUBROUTINE READFILE_TOP_FOR_BACK



SUBROUTINE CALCULATE_TEMP(TEMPC,TEMPC_OLD,I_STEP,J_STEP,K_STEP)
        REAL TEMPC(621,841),TEMPC_OLD(621,841),TE,TW,TN,TS,AE,AW,AN,AS
        !CHARACTER*20 OLD_TIME_FILE,SUB_FILE_C(400)
        INTEGER,INTENT(IN)::I_STEP,J_STEP,K_STEP
        AP0= (210)*(1E-3)*(1E-3)*(1070)*(I_STEP)*(J_STEP)/(60*K_STEP)
100     SUM_OF_ERROR=0.00
        DO I=1+I_STEP,621-I_STEP,I_STEP
                ERROR_J=0.00
                DO J=1+J_STEP,JEND(I)-J_STEP,J_STEP
                        TEMPPRV=TEMPC(I,J)
                        TW=TEMPC(I-I_STEP,J )
                        TE=TEMPC(I+I_STEP,J )
                        TN=TEMPC(I  ,J-J_STEP)
                        TS=TEMPC(I ,J+J_STEP)
                        TP=TEMPC(I ,J )
                        CALL A01(AE,AW,AN,AS,TE,TW,TN,TS,TP,I,J)
                        TEMPC(I,J)=(AW*TEMPC(I-I_STEP,J )+AE*TEMPC(I+I_STEP,J)+&
                                AN*TEMPC(I ,J-J_STEP)+AS*TEMPC(I,J+J_STEP)+&
                                AP0*TEMPC_OLD(I,J))/(AE+AW+AN+AS+AP0)
                        ERROR_J=ERROR_J+ABS(TEMPPRV-TEMPC(I,J))
                END DO
                SUM_OF_ERROR=SUM_OF_ERROR+ERROR_J
        END DO
        DO I=621,621
                ERROR_J01=0.00
                DO J=1+J_STEP,JEND(I)-J_STEP,J_STEP
```

```
                    TEMPPRV=TEMPC(I,J)
                    TW=TEMPC(I-I_STEP,J )
                    !TWE=TEMPC(I+I_STEP,J   )
                    TN=TEMPC(I  ,J-J_STEP)
                    TS=TEMPC(I ,J+J_STEP)
                    TP=TEMPC(I ,J )
                    CALL A01(AE,AW,AN,AS,TE,TW,TN,TS,TP,I,J)
                    TEMPC(I,J)=(2*AW*TEMPC(I-I_STEP,J )+&
                              AN*TEMPC(I  ,J-J_STEP)+AS*TEMPC(I ,J+J_STEP)+&
                              AP0*TEMPC_OLD(I,J))/(2*AW+AN+AS+AP0)
                    ERROR_J01=ERROR_J01+ABS(TEMPPRV-TEMPC(I,J))
                END DO
                SUM_OF_ERROR=SUM_OF_ERROR+ERROR_J01
            END DO
            DO J=841,841
                ERROR_J02=0.00
                DO I=1+I_STEP,IEND(J)-I_STEP,I_STEP
                    TEMPPRV=TEMPC(I,J)
                    TW=TEMPC(I-I_STEP,J )
                    TE=TEMPC(I+I_STEP,J   )
                    TN=TEMPC(I  ,J-J_STEP)
                    !TS=TEMPC(I ,J+J_STEP)
                    TP=TEMPC(I ,J )
                    CALL A01(AE,AW,AN,AS,TE,TW,TN,TS,TP,I,J)
                    TEMPC(I,J)=(AW*TEMPC(I-I_STEP,J )+AE*TEMPC(I+I_STEP,J )+&
                              2*AN*TEMPC(I ,J-J_STEP)+&
                              AP0*TEMPC_OLD(I,J))/(AE+AW+2*AN+AP0)
                    ERROR_J02=ERROR_J02+ABS(TEMPPRV-TEMPC(I,J))
                END DO
                SUM_OF_ERROR=SUM_OF_ERROR+ERROR_J02
            END DO
      IF(SUM_OF_ERROR>=0.1) GOTO 100
END SUBROUTINE CALCULATE_TEMP


SUBROUTINE CALCULATE_TEMP_BACK(TEMPCB,TEMPCB_OLD,I_STEP,J_STEP,K_STEP)
            REAL TEMPCB(231,841),TEMPCB_OLD(231,841),TE,TW,TN,TS,AE,AW,AN,AS
            !CHARACTER*20 OLD_TIME_FILE,SUB_FILE_C(400)
            INTEGER,INTENT(IN)::I_STEP,J_STEP,K_STEP
            AP0= (210)*(1E-3)*(1E-3)*(1070)*(I_STEP)*(J_STEP)/(60*K_STEP)
100         SUM_OF_ERROR=0.00
```

```
DO I=1+I_STEP,231-I_STEP,I_STEP
        ERROR_J=0.00
        DO J=1+J_STEP,841-J_STEP,J_STEP
                TEMPPRV=TEMPCB(I,J)
                TW=TEMPCB(I-I_STEP,J )
                TE=TEMPCB(I+I_STEP,J )
                TN=TEMPCB(I ,J-J_STEP)
                TS=TEMPCB(I ,J+J_STEP)
                TP=TEMPCB(I,J )
                CALL A02(AE,AW,AN,AS,TE,TW,TN,TS,TP,I,J)
                TEMPCB(I,J)=(AW*TEMPCB(I-I_STEP,J)+AE*TEMPCB(I+I_STEP,J  )+&
                        AN*TEMPCB(I ,J-J_STEP)+AS*TEMPCB(I ,J+J_STEP)+&
                        AP0*TEMPCB_OLD(I,J))/(AE+AW+AN+AS+AP0)
                ERROR_J=ERROR_J+ABS(TEMPPRV-TEMPCB(I,J))
        END DO
        SUM_OF_ERROR=SUM_OF_ERROR+ERROR_J
END DO
DO J=841,841
        ERROR_J02=0.00
        DO I=1+I_STEP,231-I_STEP,I_STEP
                TEMPPRV=TEMPCB(I,J)
                TW=TEMPCB(I-I_STEP,J )
                TE=TEMPCB(I+I_STEP,J )
                TN=TEMPCB(I ,J-J_STEP)
                TP=TEMPCB(I,J )
                CALL A02(AE,AW,AN,AS,TE,TW,TN,TS,TP,I,J)
                TEMPCB(I,J)=(AW*TEMPCB(I-I_STEP,J)+AE*TEMPCB(I+I_STEP,J )+&
                        2*AN*TEMPCB(I,J-J_STEP)+&
                        AP0*TEMPCB_OLD(I,J))/(AE+AW+2*AN+AP0)
                ERROR_J02=ERROR_J02+ABS(TEMPPRV-TEMPCB(I,J))
        END DO
        SUM_OF_ERROR=SUM_OF_ERROR+ERROR_J02
END DO
DO I=231,231
        ERROR_I01=0.00
        DO J=1+J_STEP,231-J_STEP,J_STEP
        TEMPPRV=TEMPCB(I,J)
                TW=TEMPCB(I-I_STEP,J  )
                !TE=TEMPCB(I+I_STEP,J )
                TN=TEMPCB(I ,J-J_STEP)
```

```
                    TS=TEMPCB(I,J+J_STEP)
                    TP=TEMPCB(I ,J)
                    CALL A02(AE,AW,AN,AS,TE,TW,TN,TS,TP,I,J)
                    TEMPCB(I,J)=(2*AW*TEMPCB(I-I_STEP,J)+AS*TS+&
                                AN*TEMPCB(I ,J-J_STEP)+&
                                AP0*TEMPCB_OLD(I,J))/(AS+2*AW+AN+AP0)
                    ERROR_I01=ERROR_I01+ABS(TEMPPRV-TEMPCB(I,J))
                END DO
                SUM_OF_ERROR=SUM_OF_ERROR+ERROR_I01
            END DO
        IF(SUM_OF_ERROR>=0.1) GOTO 100
    END SUBROUTINE CALCULATE_TEMP_BACK


    SUBROUTINE E_BACK_WALL
            REAL E_BW(231),TEMP(231),E_ST_BACK
            DO I=2,231
                    TEMP(I)=38.4279141*EXP(0.131028501*REAL(I-1)/10)
                    E_BW(I)=1.07*210*(TEMP(I)-TEMP(1))*1E-3*0.61*0.74
                    E_ST_BACK=E_ST_BACK+E_BW(I)
            END DO
                    OPEN(UNIT=7,FILE='E_BACK.TXT',STATUS='UNKNOWN')
            WRITE(7,*) E_ST_BACK
    END SUBROUTINE


    FUNCTION IEND(J)
            INTEGER,INTENT(IN)::J
            IF(J>=1 .AND. J<231) THEN
                    IEND=621
            ELSE IF(J>231 .AND. J<=841) THEN
                    IEND=231
            END IF
    END FUNCTION


    FUNCTION JEND(I)
            INTEGER,INTENT(IN)::I
            IF(I>=1 .AND. I<231) THEN
                    JEND=841
```

```
        ELSE IF(I>=231 .AND. I<=621) THEN
                JEND=231
        END IF
END FUNCTION



SUBROUTINE
TEMP_IN_KILN(K_KILN,TEMP_OF_KILN,TEMP_OF_TOP,TEMP_OF_FLUEGAS,NUMBER_OF_TEST)

        INTEGER K_KILN,NUMBER_OF_TEST
        REAL TIME_KILN,TEMP_OF_KILN,TEMP_OF_TOP,TEMP_OF_FLUEGAS
        TIME_KILN=K_KILN-1
        IF(NUMBER_OF_TEST==7)THEN
                IF(TIME_KILN>=0 .AND. TIME_KILN<=60) THEN
                        TEMP_OF_KILN=0.0027*TIME_KILN**3-0.3425*TIME_KILN**2+14.851*TIME_KILN+38
                        TEMP_OF_TOP=3E-5*TEMP_OF_KILN**3-0.0098*TEMP_OF_KILN**2
                                +1.6907*TEMP_OF_KILN+27
                        TEMP_OF_FLUEGAS=3E-16*TIME_KILN**3-(3E-15) *TIME_KILN**2
                                +0.0052*TIME_KILN+30.896
                ELSE IF(TIME_KILN>=61 .AND. TIME_KILN<=108) THEN
                        TEMP_OF_KILN=-3*10**(-6)*TIME_KILN**3-0.0257*TIME_KILN**2+6.2418*TIME_KILN
                        TEMP_OF_TOP=254.99-0.37312*TEMP_OF_KILN+0.0026409*TEMP_OF_KILN**2
                                -0.0000016893*TEMP_OF_KILN**3
                        TEMP_OF_FLUEGAS=- 0.000674569772*TIME_KILN**3+ 0.188573232*TIME_KILN**2
                                -15.773064*TIME_KILN+445.20303
                ELSE IF(TIME_KILN>=109 .AND.TIME_KILN<=196) THEN
                        TEMP_OF_KILN=-2752.5+58.196*TIME_KILN-.34936*TIME_KILN**2
                                +0.00073728*TIME_KILN**3
                        TEMP_OF_TOP=2453.1-10.706*TEMP_OF_KILN+0.018683*TEMP_OF_KILN**2
                                -0.0000097785*TEMP_OF_KILN**3
                        TEMP_OF_FLUEGAS=0.000627849358*TIME_KILN**3- 0.314083639*TIME_KILN**2
                                + 54.8852837*TIME_KILN-2962.09928
                END IF
        ELSE IF(NUMBER_OF_TEST==8)THEN
                IF(TIME_KILN>=0 .AND. TIME_KILN<=30) THEN
                        TEMP_OF_KILN=- 0.00104166667*TIME_KILN**3- 0.174107143*TIME_KILN**2
                                + 11.6309524*TIME_KILN+38.1285714
                        TEMP_OF_TOP=-9.2891829 + 2.48418261*TEMP_OF_KILN-0.0153783059
                                * TEMP_OF_KILN**2+ 0.0000408811604*TEMP_OF_KILN**3
                        TEMP_OF_FLUEGAS=30.9003663 + 0.0653294237*TIME_KILN+0.0000158791825
                                *TIME_KILN**2 - 0.0000052759312*TIME_KILN**3
```

```
        ELSE IF(TIME_KILN>=31 .AND. TIME_KILN<=66) THEN
                TEMP_OF_KILN=- 0.000347222222*TIME_KILN**3+ 0.0333002646*TIME_KILN**2
                        +0.0198412698*TIME_KILN+182.085714
                TEMP_OF_TOP=-7068.1212 + 100.856585*TEMP_OF_KILN - 0.469043117
                        * TEMP_OF_KILN**2   + 0.000732504881*TEMP_OF_KILN**3
                TEMP_OF_FLUEGAS=30.9003663 + 0.0653294237*TIME_KILN + 0.0000158791825
                        *TIME_KILN**2 - 0.0000052759312*TIME_KILN**3
        ELSE IF(TIME_KILN>=67 .AND.TIME_KILN<=132) THEN
                TEMP_OF_KILN=-2923.44266 + 86.5193862*TIME_KILN
                        +0.737794613*TIME_KILN**2+ 0.00215725632*TIME_KILN**3
                TEMP_OF_TOP=197.39373 + 0.264005544*TEMP_OF_KILN +
                        0.000964540112*TEMP_OF_KILN**2
                        - 2.64676852E-07*TEMP_OF_KILN**3
                TEMP_OF_FLUEGAS=-793.864802 + 15.2164206*TIME_KILN –
                                0.0497248122*TIME_KILN**2
                                - 0.0000163673775*TIME_KILN**3
        ELSE IF(TIME_KILN>=133 .AND.TIME_KILN<=198) THEN
                TEMP_OF_KILN=20.2999992 + 6.12482519*TIME_KILN - 0.013927739*TIME_KILN**2
                        + 0.00000647500666*TIME_KILN**3
                TEMP_OF_TOP=4428.24365 - 18.7401684*TEMP_OF_KILN + 0.0295937795
                        *TEMP_OF_KILN**2 - 0.0000147899273*TEMP_OF_KILN**3
                TEMP_OF_FLUEGAS=-198.130769 + 5.48939394*TIME_KILN - 0.0136460761
                        *TIME_KILN**2 + 0.0000113312613*TIME_KILN**3
        END IF
ELSE IF(NUMBER_OF_TEST==9)THEN
        IF(TIME_KILN>=0 .AND. TIME_KILN<=54) THEN
                TEMP_OF_KILN=189.106154 + 3.47222546*TIME_KILN - 0.0441935379*TIME_KILN**2
                        + 0.000178242539*TIME_KILN**3
                TEMP_OF_TOP=-9510.2923 + 118.562953*TEMP_OF_KILN - 0.481156352
                        *TEMP_OF_KILN**2 + 0.000653097951*TEMP_OF_KILN**3
                TEMP_OF_FLUEGAS=66.798042 + 3.72703639*TIME_KILN - 0.0813843564
                        *TIME_KILN**2 + 0.000614136378*TIME_KILN**3
        ELSE IF(TIME_KILN>=55 .AND. TIME_KILN<=114) THEN
                TEMP_OF_KILN=182.268182 + 1.85275512*TIME_KILN - 0.00397889148
                        *TIME_KILN**2 + 0.0000346232985*TIME_KILN**3
                TEMP_OF_TOP=-4525.1769 + 43.9038419*TEMP_OF_KILN - 0.134196931
                        *TEMP_OF_KILN**2 + 0.00013881541*TEMP_OF_KILN**3
                TEMP_OF_FLUEGAS=-760.241608 + 29.1938067*TIME_KILN - 0.291967754
                        *TIME_KILN**2 + 0.00100731316*TIME_KILN**3
```

```
        ELSE IF(TIME_KILN>=115 .AND.TIME_KILN<=192) THEN
                TEMP_OF_KILN=-3070.74788 + 62.7311997*TIME_KILN - 0.365970488*TIME_KILN**2
                        + 0.000728216047*TIME_KILN**3
                TEMP_OF_TOP=652.84071 - 2.67429475*TEMP_OF_KILN + 0.00723087632
                        *TEMP_OF_KILN**2 - 0.00000467961243*TEMP_OF_KILN**3
                TEMP_OF_FLUEGAS=-1397.18999 + 27.7662101*TIME_KILN - 0.145997752
                        *TIME_KILN**2 + 0.00026821286*TIME_KILN**3
        ELSE IF(TIME_KILN>=193 .AND.TIME_KILN<=228) THEN
                TEMP_OF_KILN=-5133.11429 + 67.9609127*TIME_KILN - 0.264219577*TIME_KILN**2
                        + 0.000347222222*TIME_KILN**3
                TEMP_OF_TOP=-13138.746 + 57.5308323*TEMP_OF_KILN - 0.0806848906
                        *TEMP_OF_KILN**2 + 0.000038345034*TEMP_OF_KILN**3
                TEMP_OF_FLUEGAS=6286.2619 - 90.7960979*TIME_KILN + 0.460251323
                        *TIME_KILN**2 - 0.000758744856*TIME_KILN**3
        END IF
ELSE IF(NUMBER_OF_TEST==11) THEN
        IF(TIME_KILN>=0 .AND. TIME_KILN<=84) THEN
                TEMP_OF_KILN=28.8561146 + 15.8961689*TIME_KILN - 0.719563906*TIME_KILN**2
                        + 0.0164151239*TIME_KILN**3&- 0.000177319428 *TIME_KILN**4
                        + 7.22952134E-07*TIME_KILN**5
                TEMP_OF_TOP=1.3273593 + 2.03152259*TEMP_OF_KILN - 0.0112080396
                        *TEMP_OF_KILN**2 + 0.0000295628677*TEMP_OF_KILN**3
                TEMP_OF_FLUEGAS=21.5444118 + 0.0229939668*TIME_KILN + 0.00000404007757
                        *TIME_KILN**2
        ELSE IF(TIME_KILN>=85 .AND. TIME_KILN<=210) THEN
                TEMP_OF_KILN=-4437.37056 + 120.322708*TIME_KILN - 1.10765706*TIME_KILN**2
                        + 0.00456580624*TIME_KILN**3& - 0.00000698708639
                        *TIME_KILN**4
                TEMP_OF_TOP=-173.984171 + 2.36518684*TEMP_OF_KILN - 0.00276078974
                        *TEMP_OF_KILN**2 + 0.00000180055566*TEMP_OF_KILN**3
                TEMP_OF_FLUEGAS=4177.4082 - 161.792789*TIME_KILN + 2.3413235
                        *TIME_KILN**2 - 0.0158314134*TIME_KILN**3 &
                        + 0.00005161098*TIME_KILN**4
                        - 6.56243804E-08*TIME_KILN**5
        ELSE IF(TIME_KILN>=211 .AND.TIME_KILN<=237) THEN
                TEMP_OF_KILN=-1795.24263 + 19.6192089*TIME_KILN - 0.0371739754
                        *TIME_KILN**2
                TEMP_OF_TOP=58051.0178 - 240.619581*TEMP_OF_KILN + 0.335119534
                        *TEMP_OF_KILN**2 - 0.000154751432*TEMP_OF_KILN**3
```

```
                TEMP_OF_FLUEGAS=-846.559329 + 9.3354304*TIME_KILN - 0.0159055444
                                *TIME_KILN**2

            END IF
        ELSE IF(NUMBER_OF_TEST==12) THEN
            IF(TIME_KILN>=0 .AND. TIME_KILN<=72) THEN
                TEMP_OF_KILN=32.9656755 + 16.3058057*TIME_KILN - 0.550891885*TIME_KILN**2
                                + 0.00849420868*TIME_KILN**3& - 0.0000466726825*TIME_KILN**4
                TEMP_OF_TOP=3.03700964 + 1.77917122*TEMP_OF_KILN - 0.00671808602
                                *TEMP_OF_KILN**2 + 0.0000139628223*TEMP_OF_KILN**3
                TEMP_OF_FLUEGAS=25.7769231 + 0.00641025641*TIME_KILN
            ELSE IF(TIME_KILN>=73 .AND. TIME_KILN<=144) THEN
                TEMP_OF_KILN=-1396.59263 + 40.0885302*TIME_KILN - 0.289477883*TIME_KILN**2
                                + 0.000733699171*TIME_KILN**3
                TEMP_OF_TOP=-916.876802 + 7.87199421*TEMP_OF_KILN - 0.0162189663
                                *TEMP_OF_KILN**2 + 0.0000125932592*TEMP_OF_KILN**3
                TEMP_OF_FLUEGAS=17672.2841 - 841.024087*TIME_KILN + 15.59849
                                *TIME_KILN**2 - 0.141369786*TIME_KILN**3&
                                + 0.000631284036*TIME_KILN**4 - 0.00000111446773
                                *TIME_KILN**5
            ELSE IF(TIME_KILN>=145 .AND.TIME_KILN<=216) THEN
                TEMP_OF_KILN=-1316.58574 + 25.0156878*TIME_KILN - 0.106674575*TIME_KILN**2
                                + 0.000165247561*TIME_KILN**3
                TEMP_OF_TOP=-643.243445 + 4.31847421*TEMP_OF_KILN - 0.00512895119
                                *TEMP_OF_KILN**2 + 0.00000256895866*TEMP_OF_KILN**3
                TEMP_OF_FLUEGAS=-722.657018 + 13.3697351*TIME_KILN - 0.0549401987
                                *TIME_KILN**2 + 0.0000872776914*TIME_KILN**3
            END IF
        ELSE IF(NUMBER_OF_TEST==13) THEN
            IF(TIME_KILN>=0 .AND. TIME_KILN<=108) THEN
                TEMP_OF_KILN=25.1422197 + 5.39199479*TIME_KILN - 0.174542875*TIME_KILN**2
                                + 0.00281300883*TIME_KILN**3 &- 0.0000213481139*TIME_KILN**4
                                + 6.18459055E-08*TIME_KILN**5
                TEMP_OF_TOP=-11.8637433 + 2.83453726*TEMP_OF_KILN - 0.0347760428
                                *TEMP_OF_KILN**2 + 0.000168148491*TEMP_OF_KILN**3
                TEMP_OF_FLUEGAS=22.9473684 + 0.0289473684*TIME_KILN
            ELSE IF(TIME_KILN>=109 .AND. TIME_KILN<=210) THEN
                TEMP_OF_KILN=-8400.40017 + 202.246532*TIME_KILN - 1.77363535*TIME_KILN**2
                                + 0.00695211929*TIME_KILN**3 & - 0.0000102013336*TIME_KILN**4
                TEMP_OF_TOP=-118.715524 + 2.68255937*TEMP_OF_KILN - 0.00676822779
                                *TEMP_OF_KILN**2 + 0.0000084793657*TEMP_OF_KILN**3
```

```
                    TEMP_OF_FLUEGAS=497.311156 - 14.046369*TIME_KILN + 0.155775315
                                    *TIME_KILN**2 - 0.000763265008*TIME_KILN**3&
                                    + 0.00000140119881*TIME_KILN**4
            ELSE IF(TIME_KILN>=211 .AND.TIME_KILN<=342) THEN
                    TEMP_OF_KILN=-6613.51277 + 73.9794065*TIME_KILN - 0.25711962*TIME_KILN**2
                                    + 0.000304456681*TIME_KILN**3
                    TEMP_OF_TOP=-581.674567 + 4.23563623*TEMP_OF_KILN - 0.00561935475
                                    *TEMP_OF_KILN**2 + 0.000003232139*TEMP_OF_KILN**3
                    TEMP_OF_FLUEGAS=-29287.6031 + 390.202303*TIME_KILN - 1.92864876
                                    *TIME_KILN**2 + 0.00423229328*TIME_KILN**3 &
                                    - 0.0000034628752*TIME_KILN**4
            END IF
        END IF


END SUBROUTINE


SUBROUTINE CALCULATE_ENERGY_STORE(I_STEP,J_STEP,K_CALE,K_BURN,ENERGY_TS)


        REAL TEMPE(621,841),TEMPE_OLD(621,841),DELTA_TEMP,LENGTH,ENERGY_TS(400),V1,V2,V3,&
                M1,M2,M3
        INTEGER I_STEP,J_STEP,K_CALE,K_BURN
        CHARACTER*20 SUB_FILE_OLD(400),TIME_FILE_OLD,SUB_FILE(400),TIME_FILE
        LENGTH=0.72
        DENSITY=210
        CP=1.070
        V1=I_STEP*J_STEP*(1E-3)*(1E-3)*LENGTH
        V2=I_STEP*J_STEP*(1E-3)*(1E-3)*LENGTH/2
        V3=I_STEP*J_STEP*(1E-3)*(1E-3)*LENGTH/4
        M1=V1*DENSITY
        M2=V2*DENSITY
        M3=V3*DENSITY
        DO K=1+K_CALE,K_BURN,K_CALE
        CALL NAME_OF_FILE(SUB_FILE_OLD,K-K_CALE)
        TIME_FILE_OLD=SUB_FILE_OLD(K-K_CALE)
        CALL READ_FILE(TEMPE_OLD,I_STEP,J_STEP,TIME_FILE_OLD)
        CALL NAME_OF_FILE(SUB_FILE,K)
```

```
TIME_FILE=SUB_FILE(K)
CALL READ_FILE(TEMPE,I_STEP,J_STEP,TIME_FILE)
!INTERIOR NODE
        DELTA_ENERGY=0.00
        DO I=1+I_STEP,621-I_STEP,I_STEP
                DO J=1+J_STEP,JEND(I)-J_STEP,J_STEP
                        DELTA_TEMP=(TEMPE(I,J)-TEMPE_OLD(I,J))
                        DELTA_ENERGY=DELTA_ENERGY+DELTA_TEMP*M1*CP
                END DO
        END DO
        ENERGY_STORE_INTERIOR_NODE=DELTA_ENERGY
!TOP SURFACE
        DELTA_ENERGY=0.00
                DO I=1+I_STEP,IEND(1)-I_STEP,I_STEP
                        DELTA_TEMP=(TEMPE(I,1)-TEMPE_OLD(I,1))
                        DELTA_ENERGY=DELTA_ENERGY+DELTA_TEMP*M2*CP

                END DO
        ENERGY_STORE_TOP_SURFACE=DELTA_ENERGY
!LEFT SURFACE
         DELTA_ENERGY=0.00
                DO J=1+J_STEP,JEND(1)-J_STEP,J_STEP
                        DELTA_TEMP=(TEMPE(1,J)-TEMPE_OLD(1,J))
                        DELTA_ENERGY=DELTA_ENERGY+DELTA_TEMP*M2*CP
                END DO
        ENERGY_STORE_LEFT_SURFACE=DELTA_ENERGY
!TOP SURFACE IN KILN
        DELTA_ENERGY=0.00
                DO I=231+I_STEP,621-I_STEP,I_STEP
                        DELTA_TEMP=(TEMPE(I,231)-TEMPE_OLD(I,231))
                        DELTA_ENERGY=DELTA_ENERGY+DELTA_TEMP*M2*CP
                END DO
        ENERGY_STORE_TOP_KILN=DELTA_ENERGY
!LEFT SURFACE IN KILN
        DELTA_ENERGY=0.0
                DO J=231+J_STEP,841-J_STEP,J_STEP
                        DELTA_TEMP=(TEMPE(231,J)-TEMPE_OLD(231,J))
                        DELTA_ENERGY=DELTA_ENERGY+DELTA_TEMP*M2*CP
                END DO
        ENERGY_STORE_LEFT_KILN=DELTA_ENERGY
!INSULATE RIGHT SURFACE
```

```
                DELTA_ENERGY=0.00


                        DO J=1+J_STEP,231-J_STEP,J_STEP
                                DELTA_TEMP=(TEMPE(621,J)-TEMPE_OLD(621,J))
                                DELTA_ENERGY=DELTA_ENERGY+DELTA_TEMP*M2*CP
                        END DO
                ENERGY_STORE_INSULATE_RIGHT=DELTA_ENERGY
        !INSULATE BOTTOM SURFACE
                DELTA_ENERGY=0.00
                        DO I=1+I_STEP,231-I_STEP,I_STEP
                                DELTA_TEMP=(TEMPE(I,841)-TEMPE_OLD(I,841))
                                DELTA_ENERGY=DELTA_ENERGY+DELTA_TEMP*M2*CP
                        END DO
                ENERGY_STORE_INSULATE_BOTTOM=DELTA_ENERGY
        ! CONNER OF KILN
                DELTA_ENERGY=0.00
                DELTA_ENERGY=DELTA_ENERGY+(TEMPE( 1, 1)-TEMPE_OLD( 1, 1))*M3*CP
                DELTA_ENERGY=DELTA_ENERGY+(TEMPE( 1,841)-TEMPE_OLD( 1, 1))*M3*CP
                DELTA_ENERGY=DELTA_ENERGY+(TEMPE(231,231)-TEMPE_OLD(231,231))*M3*CP
                DELTA_ENERGY=DELTA_ENERGY+(TEMPE(231,841)-TEMPE_OLD(231,841))*M3*CP
                DELTA_ENERGY=DELTA_ENERGY+(TEMPE(621, 1)-TEMPE_OLD(621, 1))*M3*CP
                DELTA_ENERGY=DELTA_ENERGY+(TEMPE(621,231)-TEMPE_OLD(621,231))*M3*CP
                ENERGY_STORE_CONNER=DELTA_ENERGY
                ENERGY_STORE_TOTAL=
                ENERGY_STORE_INTERIOR_NODE+ENERGY_STORE_TOP_SURFACE+ &
                ENERGY_STORE_LEFT_SURFACE+ENERGY_STORE_TOP_KILN+ &
                ENERGY_STORE_LEFT_KILN+ENERGY_STORE_INSULATE_RIGHT+  &
                ENERGY_STORE_INSULATE_BOTTOM+ENERGY_STORE_CONNER
                ENERGY_TS(K)=ENERGY_STORE_TOTAL
                ENERGY_STT=ENERGY_STT+ENERGY_STORE_TOTAL
        END DO
        OPEN(UNIT=8,FILE='E_ST_TS.TXT',STATUS='UNKNOWN')
        WRITE(8,30) ENERGY_STT
        CLOSE(8)
        OPEN(UNIT=9,FILE='E_ST_TS_TIME.TXT',STATUS='UNKNOWN')
        DO I=1+K_CALE,K_BURN,K_CALE
                WRITE(9,25) ENERGY_TS(I)
        END DO
        CLOSE(9)
25  FORMAT(2X,F8.4)
30      FORMAT(2X,'ENERGY STORE IN WALL',F12.4,'KJ')
```

```
END SUBROUTINE CALCULATE_ENERGY_STORE


SUBROUTINE CALCULATE_ENERGYSTORE_CERAMIC_AND_FERNITURE(K_END,E_ST_CAF,N_TEST)

REAL E_ST_CAF(400),MASS_FER,MASS_CE,CP_FER,CP_CE,TEMP_KILN,TEMP_KILN_OLD,DUMMY_TEMP,&
                DUMMY01,E_CF_INTV(400)
        INTEGER K_END,KM1,N_TEST
        WRITE(*,*) 'HOW MANY MASS OF CERAMIC'
        READ (*,*) MASS_CE
        WRITE(*,*) 'HOW MANY MASS OF FERNITURE'
        READ (*,*) MASS_FER
        CP_CE=0.88
        CP_FER= 0.74 !0.65
        K_STEP=6
        DO K=2,K_END,1
                KM1=K-1
                CALL TEMP_IN_KILN(KM1,TEMP_KILN_OLD,DUMMY_TEMP,DUMMY01,N_TEST)
                CALL TEMP_IN_KILN(K,TEMP_KILN,DUMMY_TEMP,DUMMY01,N_TEST)
                DELTA_TEMP=TEMP_KILN-TEMP_KILN_OLD
                E_ST_CAF(K)=(MASS_FER*CP_FER+MASS_CE*CP_CE)*DELTA_TEMP
                E_ST_CF=E_ST_CF+(MASS_FER*CP_FER+MASS_CE*CP_CE)*DELTA_TEMP
        END DO
        OPEN(UNIT=8,FILE='ENERGY STORE IN CERAMIC AND FERNITURE.TXT',STATUS='UNKNOWN')
        WRITE(8,23) E_ST_CF
        OPEN(UNIT=9,FILE='ENERGY SORE IN CARAMIC AND
FERNITURE_EACH_TIME.TXT',STATUS='UNKNOWN')
        DO K=1,K_END,1
                WRITE(9,25) E_ST_CAF(K)
        END DO
        DO K=1+K_STEP,K_END,K_STEP
                DO I=1,K_STEP,1
                        E_CF_INTV(K)= E_CF_INTV(K)+E_ST_CAF(K-I)
                END DO
        END DO
        OPEN(UNIT=10,FILE='ENERGY SORE IN CARAMIC AND
FERNITURE_INTIVAL_TIME.TXT',STATUS='UNKNOWN')
        DO I=1+K_STEP,K_END,K_STEP
                WRITE(10,25) E_CF_INTV(I)
        END DO
```

```
            CLOSE(8)
23          FORMAT(2X,'ENERGY STORE IN CERAMIC AND FERNITURE IS',F12.4,'KJ')
25  FORMAT(2X,F12.4)
END SUBROUTINE CALCULATE_ENERGYSTORE_CERAMIC_AND_FERNITURE

FUNCTION P_GAS(PG)
            INTEGER PG
            IF (PG==1) THEN
                        P_GAS=0.011333
            ELSE IF (PG==2) THEN
                        P_GAS=   0.02266 !0.025131
            ELSE IF (PG==3) THEN
                        P_GAS=  0.026322 !0.029857
            ELSE IF (PG==4) THEN
                        P_GAS=   0.030564  !0.03622
            END IF
END FUNCTION

SUBROUTINE ANALYSIS_FLUEGAS(MASS_N2,MASS_H2O,MASS_O2,MASS_CO2)
            REAL,INTENT(INOUT):: MASS_N2(4),MASS_H2O(4),MASS_O2(4),MASS_CO2(4)
            REAL MOLE_AIR_TH,MW_N2,MW_O2,MW_CO2,MW_AIR,MW_LPG,COP(4)
            MOLE_AIR_TH=5.6
            MW_N2=28
            MW_O2=32
            MW_CO2=28
            MW_AIR=137.28
            MW_LPG=49.6
            MW_H2O=18
            COP(1)=1.068
            COP(2)=1.068
            COP(3)=1.13
            COP(4)=1.20
            DO K=1,4
                        MASS_N2(K)=(3.76*MOLE_AIR_TH*MW_N2*EXCESS_AIR(K))/MW_LPG*P_GAS(K)*COP(K)
!KG/MIN
                        MASS_O2(K)=(EXCESS_AIR(K)-1)*5.6*MW_O2/MW_LPG*P_GAS(K)*COP(K)
                        MASS_CO2(K)=3.4*MW_CO2/MW_LPG*P_GAS(K)
                        MASS_H2O(K)=4.4*MW_H2O/MW_LPG*P_GAS(K)+MOLE_AIR_TH*EXCESS_AIR(K)*MW_AIR*0.0
                                    24*COP(K)*P_GAS(K)/MW_LPG
```

```
        END DO


END SUBROUTINE ANALYSIS_FLUEGAS
FUNCTION EXCESS_AIR(EA)
        INTEGER EA
        IF(EA==1) THEN
                PER_O2=18
        ELSE IF(EA==2) THEN
                PER_O2=16.5
        ELSE IF (EA==3) THEN
                PER_O2=9
        ELSE IF (EA==4) THEN
                PER_O2=5
        END IF
        EXCESS_AIR=(21/(21-PER_O2))
END FUNCTION

SUBROUTINE ENERGY_INOUT(E_OUT,N_TEST,K_BURN,K_STEP)
        REAL E_OUT(400),FG_TEMP,DUMMY01,DUMMY02,M_N2(4),M_H2O(4),M_O2(4),M_CO2(4),&
         FG_TEMP_OLD,TIMES(4),MGAS(4),MASS_GAS,E_IN(400),M_IN(400),OPEL_OUT(400),B_TEMP,&
                E_OUT_INTV(400),OPEL_INTV(400)
        INTEGER  K_BURN,K_STEP
        !WRITE(*,*)'WHAT ARE STARTING HALF(H) OR FULL(F) BURNNER'
        !READ (*,*)

        IF(N_TEST==13) THEN
                TIMES(1)=107
                TIMES(2)=101
                TIMES(3)=131
        ELSE IF(N_TEST==11) THEN
                TIMES(1)=0
                TIMES(2)=87
                TIMES(3)=120
                TIMES(4)=27
        ELSE
        WRITE(*,*) 'HOW MANY TIME BURNING AT PRESSURE 0.2 kPa/sq.cm.(HALF BURNER)'
        READ(*,*) TIMES(1)
        WRITE(*,*) 'HOW MANY TIME BURNING AT PRESSURE 0.2 kPa/sq.cm.(FULL BURNER)'
        READ(*,*) TIMES(2)
        WRITE(*,*) 'HOW MANY TIME BURNING AT PRESSURE 0.25 kPa/sq.cm.(FULL BURNER)'
```

```
READ(*,*) TIMES(3)
WRITE(*,*) 'HOW MANY TIME BURNING AT PRESSURE 0.3 kPa/sq.cm.(FULL BURNER)'
READ(*,*) TIMES(4)
END IF
MASS_GAS=0.00
DO K=1,4
        MGAS(K)=P_GAS(K)*TIMES(K)
        MASS_GAS=MASS_GAS+MGAS(K)
END DO
OPEN(UNIT=8,FILE='MASS GASS USED.TXT',STATUS='UNKNOWN')
WRITE (8,23) MASS_GAS
DO I=1,K_BURN,1
        IF(I>=1 .AND. I<=TIMES(1)+1) THEN
        M_IN(I)= P_GAS(1)
        ELSE IF(I>TIMES(1)+1 .AND. I<=TIMES(1)+TIMES(2)+1) THEN
         M_IN(I)= P_GAS(2)
        ELSE IF(I>TIMES(1)+TIMES(2)+1 .AND. I<=TIMES(1)+TIMES(2)+TIMES(3)+1) THEN
        M_IN(I)= P_GAS(3)
        ELSE IF(I>TIMES(1)+TIMES(2)+TIMES(3)+1 .AND.
I<=TIMES(1)+TIMES(2)+TIMES(3)+TIMES(4)+1) THEN
        M_IN(I)= P_GAS(4)
        END IF
END DO



DO K=1+K_STEP,K_BURN,K_STEP
        E_IN(K)=0.00
        DO I=1,K_STEP
                E_IN(K)=E_IN(K)+M_IN(K-I)*49E3
        END DO
END DO
OPEN(UNIT=9,FILE='E_IN_TIME.TXT',STATUS='UNKNOWN')
DO I=1+K_STEP,K_BURN,K_STEP
        WRITE(9,27) E_IN(I)
END DO
CLOSE(9)
CALL ANALYSIS_FLUEGAS(M_N2,M_H2O,M_O2,M_CO2)
B_TEMP=23.00
ENERGYOUT_P1=0.00
ENERGYOUT_P2=0.00
```

```
      ENERGYOUT_P3=0.00
      ENERGYOUT_P4=0.00
      ENERGY_OUT_TOTAL=0.00
!     DO I=1+K_STEP,K_BURN,K_STEP
!           IM1=I-K_STEP
!           CALL TEMP_IN_KILN(I  ,DUMMY01,DUMMY02,FG_TEMP,N_TEST)
!           CALL TEMP_IN_KILN(IM1,DUMMY01,DUMMY02,FG_TEMP_OLD,N_TEST)
!           DELTA_H_N2= (H_N2(FG_TEMP)-H_N2(B_TEMP))
!           DELTA_H_O2= (H_O2(FG_TEMP)-H_O2(B_TEMP))
!           DELTA_H_CO2=(H_CO2(FG_TEMP)-H_CO2(B_TEMP))
!           DELTA_H_H2O=(H_H2O(FG_TEMP)-H_H2O(B_TEMP))
!           IF(I>=1 .AND. I<=TIMES(1)+1) THEN!
!           E_OUT(I)= M_N2 (1)*DELTA_H_N2*K_STEP+M_O2 (1)*DELTA_H_O2*K_STEP+&
!                 M_CO2(1)*DELTA_H_CO2*K_STEP+M_H2O(1)*DELTA_H_H2O*K_STEP
!           ENERGYOUT_P1=ENERGYOUT_P1+E_OUT(I)*0.01
!           ELSE IF(I>TIMES(1)+1 .AND. I<=TIMES(1)+TIMES(2)+1) THEN
!           E_OUT(I)= M_N2 (2)*DELTA_H_N2*K_STEP+M_O2 (2)*DELTA_H_O2*K_STEP+&
!                 M_CO2(2)*DELTA_H_CO2*K_STEP+M_H2O(2)*DELTA_H_H2O*K_STEP
!           ENERGYOUT_P2=ENERGYOUT_P2+E_OUT(I)*0.3
!           ELSE IF(I>TIMES(1)+TIMES(2)+1 .AND. I<=TIMES(1)+TIMES(2)+TIMES(3)+1) THEN
!           E_OUT(I)=M_N2 (3)*DELTA_H_N2*K_STEP+M_O2 (3)*DELTA_H_O2*K_STEP+&
!                 M_CO2(3)*DELTA_H_CO2*K_STEP+M_H2O(3)*DELTA_H_H2O*K_STEP
!           ENERGYOUT_P3=ENERGYOUT_P3+E_OUT(I)*0.9
!           ELSE IF(I>TIMES(1)+TIMES(2)+TIMES(3)+1 .AND.
            I<=TIMES(1)+TIMES(2)+TIMES(3)+TIMES(4)+1) THEN

!           E_OUT(I)= M_N2 (4)*DELTA_H_N2*K_STEP+M_O2 (4)*DELTA_H_O2*K_STEP+&
!                 M_CO2(4)*DELTA_H_CO2*K_STEP+M_H2O(4)*DELTA_H_H2O*K_STEP
!           ENERGYOUT_P4=ENERGYOUT_P4+E_OUT(I)*0.9
!           END IF
!     END DO
!     ENERGY_OUT_TOTAL=ENERGYOUT_P1+ENERGYOUT_P2+ENERGYOUT_P3+ENERGYOUT_P4
      F_1=.01
      F_2=0.2
      F_3=0.8
      F_4=0.8
      DO I=2,K_BURN,1
            IM1=I-1
            CALL TEMP_IN_KILN(I  ,DUMMY01,DUMMY02,FG_TEMP,N_TEST)
            CALL TEMP_IN_KILN(IM1,DUMMY01,DUMMY02,FG_TEMP_OLD,N_TEST)
            DELTA_H_N2= (H_N2(FG_TEMP)-H_N2(B_TEMP))
```

```
            DELTA_H_O2= (H_O2(FG_TEMP)-H_O2(B_TEMP))
            DELTA_H_CO2=(H_CO2(FG_TEMP)-H_CO2(B_TEMP))
            DELTA_H_H2O=(H_H2O(FG_TEMP)-H_H2O(B_TEMP))
            IF(I>=1 .AND. I<=TIMES(1)+1) THEN
            E_OUT(I)= (M_N2 (1)*DELTA_H_N2+M_O2 (1)*DELTA_H_O2+&
                        M_CO2(1)*DELTA_H_CO2+M_H2O(1)*DELTA_H_H2O)*F_1
            ENERGYOUT_P1=ENERGYOUT_P1+E_OUT(I)
            ELSE IF(I>TIMES(1)+1 .AND. I<=TIMES(1)+TIMES(2)+1) THEN
            E_OUT(I)= (M_N2 (2)*DELTA_H_N2+M_O2 (2)*DELTA_H_O2+&
                        M_CO2(2)*DELTA_H_CO2+M_H2O(2)*DELTA_H_H2O)*F_2
            ENERGYOUT_P2=ENERGYOUT_P2+E_OUT(I)
            ELSE IF(I>TIMES(1)+TIMES(2)+1 .AND. I<=TIMES(1)+TIMES(2)+TIMES(3)+1) THEN
            E_OUT(I)=(M_N2 (3)*DELTA_H_N2+M_O2 (3)*DELTA_H_O2+&
                        M_CO2(3)*DELTA_H_CO2+M_H2O(3)*DELTA_H_H2O)*F_3
            ENERGYOUT_P3=ENERGYOUT_P3+E_OUT(I)
            ELSE IF(I>TIMES(1)+TIMES(2)+TIMES(3)+1 .AND.
            I<=TIMES(1)+TIMES(2)+TIMES(3)+TIMES(4)+1) THEN
            E_OUT(I)= (M_N2 (4)*DELTA_H_N2+M_O2 (4)*DELTA_H_O2+&
                        M_CO2(4)*DELTA_H_CO2+M_H2O(4)*DELTA_H_H2O)*F_4
            ENERGYOUT_P4=ENERGYOUT_P4+E_OUT(I)
            END IF
END DO
DO K=1+K_STEP,K_BURN,K_STEP
            E_OUT_INTV(K)=0.00
            DO I=1,K_STEP
                        E_OUT_INTV(K)=E_OUT_INTV(K)+E_OUT(K-I)
            END DO
END DO
ENERGY_OUT_TOTAL=ENERGYOUT_P1+ENERGYOUT_P2+ENERGYOUT_P3+ENERGYOUT_P4
OPELOUT_P1=0.00
OPELOUT_P2=0.00
OPELOUT_P3=0.00
OPELOUT_P4=0.00
OPEL_OUT_TOTAL=0.00
            DO I=2,K_BURN,1
            IM1=I-1
            CALL TEMP_IN_KILN(I  ,FG_TEMP,DUMMY02,DUMMY01,N_TEST)
            CALL TEMP_IN_KILN(IM1,FG_TEMP_OLD,DUMMY02,DUMMY01,N_TEST)
            DELTA_H_N2= (H_N2(FG_TEMP)-H_N2(B_TEMP))
            DELTA_H_O2= (H_O2(FG_TEMP)-H_O2(B_TEMP))
            DELTA_H_CO2=(H_CO2(FG_TEMP)-H_CO2(B_TEMP))
```

```
            DELTA_H_H2O=(H_H2O(FG_TEMP)-H_H2O(B_TEMP))
            IF(I>=1 .AND. I<=TIMES(1)+1) THEN
            OPEL_OUT(I)=(M_N2 (1)*DELTA_H_N2+M_O2 (1)*DELTA_H_O2+&
                        M_CO2(1)*DELTA_H_CO2+M_H2O(1)*DELTA_H_H2O)*(1-F_1)
            OPELOUT_P1=OPELOUT_P1+OPEL_OUT(I)
            ELSE IF(I>TIMES(1)+1 .AND. I<=TIMES(1)+TIMES(2)+1) THEN
            OPEL_OUT(I)= (M_N2 (2)*DELTA_H_N2+M_O2 (2)*DELTA_H_O2+&
                         M_CO2(2)*DELTA_H_CO2+M_H2O(2)*DELTA_H_H2O)*(1-F_2)
            OPELOUT_P2=OPELOUT_P2+OPEL_OUT(I)
            ELSE IF(I>TIMES(1)+TIMES(2)+1 .AND. I<=TIMES(1)+TIMES(2)+TIMES(3)+1) THEN
            OPEL_OUT(I)= (M_N2 (3)*DELTA_H_N2+M_O2 (3)*DELTA_H_O2+&
                         M_CO2(3)*DELTA_H_CO2+M_H2O(3)*DELTA_H_H2O)*(1-F_3)
            OPELOUT_P3=OPELOUT_P3+OPEL_OUT(I)
            ELSE IF(I>TIMES(1)+TIMES(2)+TIMES(3)+1 .AND.
            I<=TIMES(1)+TIMES(2)+TIMES(3)+TIMES(4)+1) THEN
            OPEL_OUT(I)= (M_N2 (4)*DELTA_H_N2+M_O2 (4)*DELTA_H_O2+&
                         M_CO2(4)*DELTA_H_CO2+M_H2O(4)*DELTA_H_H2O)*(1-F_4)
            OPELOUT_P4=OPELOUT_P4+OPEL_OUT(I)
            END IF
END DO
OPEL_OUT_TOTAL=OPELOUT_P1+OPELOUT_P2+OPELOUT_P3+OPELOUT_P4
DO K=1+K_STEP,K_BURN,K_STEP
            OPEL_INTV(K)=0.00
            DO I=1,K_STEP
                        OPEL_INTV(K)=OPEL_INTV(K)+OPEL_OUT(K-I)
            END DO
END DO
OPEN(UNIT=7,FILE='FLUEGAS_OUT.TXT',STATUS='UNKNOWN')
WRITE(7,20)          ENERGY_OUT_TOTAL
OPEN(UNIT=8,FILE='FLUEGAS_OUT_EACH_TIME.TXT',STATUS='UNKNOWN')
DO I=1,K_BURN,1
            WRITE(8,25) E_OUT(I)
END DO
OPEN(UNIT=9,FILE='FLUEGAS_OUT_INV_TIME.TXT',STATUS='UNKNOWN')
DO I=1+K_STEP,K_BURN,K_STEP
            WRITE(9,25) E_OUT_INTV(I)
END DO
CLOSE(7)
CLOSE(8)
CLOSE(9)
OPEN(UNIT=7,FILE='OPEN_LOSS_OUT.TXT',STATUS='UNKNOWN')
```

```
        WRITE(7,20)         OPEL_OUT_TOTAL
        OPEN(UNIT=8,FILE='OPEN_LOSS_OUT_EACH_TIME.TXT',STATUS='UNKNOWN')
        DO I=1,K_BURN,1
                WRITE(8,25) OPEL_OUT(I)
        END DO
        OPEN(UNIT=9,FILE='OPEN_LOSS_OUT_INVT_TIME.TXT',STATUS='UNKNOWN')
        DO I=1+K_STEP,K_BURN,K_STEP
                WRITE(9,25) OPEL_INTV(I)
        END DO
        CLOSE(7)
        CLOSE(8)
        CLOSE(9)
20      FORMAT(2X,'ENERGY OUT FROM FLUE GAS IS',2X,F12.4,'KJ')
23      FORMAT(2X,'MASS USED IN PROCESS IS',F6.3,'kg')
25  FORMAT(2X,F12.4)
27  FORMAT(2X,F9.4)
END SUBROUTINE ENERGY_INOUT
FUNCTION H_CO2(FG_TEMP)
        REAL FG_TEMP
        H_CO2 = -8963.8+0.84283*(FG_TEMP)+0.00041569*(FG_TEMP)**2-1.3513E-07*(FG_TEMP)**3
END FUNCTION H_CO2
!-------------------------------------------------------------------------------------------------------
!                                       use for find entahlpy of Nitrogen each times
!-------------------------------------------------------------------------------------------------------


FUNCTION H_N2(FG_TEMP)
        REAL FG_TEMP
        H_N2=-24.172+1.0095*(FG_TEMP)+0.00010909*(FG_TEMP)**2
END FUNCTION
!-------------------------------------------------------------------------------------------------------
!                                       use for find entahlpy of Oxegen each times
!-------------------------------------------------------------------------------------------------------
FUNCTION H_O2(FG_TEMP)
        REAL FG_TEMP
        H_O2=-22.225+0.8952*(FG_TEMP)+0.00019472*(FG_TEMP)**2-5.6961E-08*(FG_TEMP)**3
END FUNCTION H_O2
!-------------------------------------------------------------------------------------------------------
!                                       use for find entahlpy of Vapor each times
!-------------------------------------------------------------------------------------------------------
FUNCTION H_H2O(FG_TEMP)
```

```
        REAL FG_TEMP
        H_H2O=-13467+1.8177*(FG_TEMP)+0.00029307*(FG_TEMP)**2+3.1642E-08*(FG_TEMP)**3
END FUNCTION H_H2O


SUBROUTINE A01(AE,AW,AN,AS,TE,TW,TN,TS,TP,I,J,I_STEP,J_STEP)
        REAL AW,AE,AN,AS,TW,TE,TN,TS,TP
        INTEGER I,J
        DX=1E-3*I_STEP
        DY=1E-3*J_STEP

        !COF01=1.5
        IM1=I-I_STEP
        IP1=I+I_STEP
        JM1=J-J_STEP
        JP1=J+J_STEP
        CE=CFIBER(TE)*COF(IP1,J  )*DY/DX
        CW=CFIBER(TW)*COF(IM1,J  )*DY/DX
        CN=CFIBER(TN)*COF(I  ,JM1)*DY/DX
        CS=CFIBER(TS)*COF(I  ,JP1)*DY/DX
        CP=CFIBER(TP)*COF(I  ,J  )*DY/DX
        CE01=CE
        CW01=CW
        CN01=CN
        CS01=CS

        IF (I==1) THEN
                IF (J==1) THEN
                        AE=CE01
                        AW=0
                        AN=0
                        AS=CS01
                ELSE IF(J>1 .AND. J<841) THEN
                        AN=CN01/2
                        AS=CS01/2
                        AE=CE01
                        AW=0
                ELSE IF (J==841) THEN

                        AE=CE01
                        AW=0
```

```
                AN=CN01
                AS=0
            END IF
        ELSE IF (I>1 .AND. I<231) THEN
            IF(J==1) THEN
                AE=CE01/2
                AW=CW01/2
                AN=0
                AS=CS01
            ELSE IF(J>1 .AND. J<841) THEN
                AE=CE01
                AW=CW01
                AN=CN01
                AS=CS01
            ELSE IF(J==841) THEN

                AE=CE01/2
                AW=CW01/2
                AN=CN01
                AS=0
            END IF
        ELSE IF(I==231)     THEN
            IF(J==1) THEN
                AE=CE01/2
                AW=CW01/2
                AN=0
                AS=CS01
            ELSE IF(J>1 .AND. J<231) THEN
                AE=CE01
                AW=CW01
                AN=CN01
                AS=CS01
            ELSE IF(J==231) THEN
                AE=CE01
                AW=CW01
                AN=CN01
                AS=CS01
            ELSE IF(J>231 .AND. J<841) THEN
                AE=0
                AW=CW01
                AN=CN01/2
```

```
                AS=CS01/2
            ELSE IF (J==841) THEN
                AE=0
                AW=CW01
                AN=CN01
                AS=0

            END IF
        ELSE IF(I>231 .AND. I<621) THEN
            IF(J==1) THEN
                AE=CE01/2
                AW=CW01/2
                AN=0
                AS=CS01
            ELSE IF (J>1 .AND. J<231) THEN
                AE=CE01
                AW=CW01
                AN=CN01
                AS=CS01
            ELSE IF(J==231) THEN
                AE=CE01/2
                AW=CW01/2
                AN=CN01
                AS=0
            END IF

        ELSE IF(I==621) THEN
            AE=0
            IF (J==1) THEN
                AW=CW01
                AN=0
                AS=CS01
            ELSE IF(J>1 .AND. J<231) THEN
                AW=CW01
                AN=CN01/2
                AS=CS01/2
            ELSE IF (J==231) THEN
                AW=CW01
                AN=CN01
                AS=0
            END IF
```

```
        END IF

END SUBROUTINE A01

SUBROUTINE A02(AE,AW,AN,AS,TE,TW,TN,TS,T,I,J)
        REAL AW,AE,AN,AS,TW,TE,TN,TS
        INTEGER I,J
        DX=1E-3
        DY=1E-3
        IM1=I-1
        IP1=I+1
        JM1=J-1
        JP1=J+1
        CE=0.0616938432*EXP(0.00148978098*TE)*COFB(IP1,J  )*DY/DX
        CW=0.0616938432*EXP(0.00148978098*TW)*COFB(IM1,J  )*DY/DX
        CN=0.0616938432*EXP(0.00148978098*TN)*COFB(I  ,JM1)*DY/DX
        CS=0.0616938432*EXP(0.00148978098*TS)*COFB(I  ,JP1)*DY/DX
        CP=0.0616938432*EXP(0.00148978098*T )*COFB(I  ,J  )*DY/DX
        CE01=2*CE*CP/(CE+CP)
        CW01=2*CW*CP/(CW+CP)
        CN01=2*CN*CP/(CN+CP)
        CS01=2*CS*CP/(CS+CP)
        IF (I==1) THEN
                IF (J==1) THEN
                        AE=CE01
                        AW=0
                        AN=0
                        AS=CS01
                ELSE IF(J>1 .AND. J<841) THEN
                        AN=CN01/2
                        AS=CS01/2
                        AE=CE01
                        AW=0
                ELSE IF (J==841) THEN
                        AE=CE01
                        AW=0
                        AN=CN01
                        AS=0
                END IF
        ELSE IF (I>1 .AND. I<231) THEN
                IF(J==1) THEN
```

```
                    AE=CE01/2
                    AW=CW01/2
                    AN=0
                    AS=CS01
                ELSE IF(J>1 .AND. J<841) THEN
                    AE=CE01
                    AW=CW01
                    AN=CN01
                    AS=CS01
                ELSE IF(J==841) THEN
                    AE=CE01/2
                    AW=CW01/2
                    AN=CN01
                    AS=0
                END IF
            ELSE IF(I==231)    THEN
                IF(J==1) THEN
                    AE=0
                    AW=CW01
                    AN=0
                    AS=CS01

                ELSE IF(J>1 .AND. J<841) THEN
                    AE=0
                    AW=CW01
                    AN=CN01/2
                    AS=CS01/2
                ELSE IF (J==841) THEN
                    AE=0
                    AW=CW01
                    AN=CN01
                    AS=0
                END IF
            END IF
END SUBROUTINE A02
FUNCTION AE(IAE,TE)
            INTEGER,INTENT(IN):: IAE
            REAL,INTENT(IN)::TE
            REAL COF
            !COF=2
            IF (IAE>=1 .AND. IAE<=116) THEN
```

```
                COF=1.5
        ELSE
                COF=2
        END IF
                CONDUCT=(0.0522*EXP(0.0017*TE))*COF
        DX=1E-3
        DY=1E-3
        IF(IAE==1) THEN
                AE=0.0
        ELSE IF(IAE>=2 .AND. IAE<=621) THEN
                AE=CONDUCT*DX/DY
        END IF
END FUNCTION AE
SUBROUTINE  NAME_OF_FILE(SUB_FILE,I)
        INTEGER I
        CHARACTER*20 SUB_FILE(400)
        IF (I==1) THEN
                SUB_FILE(I)='T0001.TXT'
        ELSE IF (I==2) THEN
                SUB_FILE(I)='T0002.TXT'
        ELSE IF (I==3) THEN
                SUB_FILE(I)='T0003.TXT'
        ELSE IF (I==397) THEN
                SUB_FILE(I)='T0397.TXT'
        ELSE IF (I==398) THEN
                SUB_FILE(I)='T0398.TXT'
        ELSE IF (I==399) THEN
                SUB_FILE(I)='T0399.TXT'
        ELSE IF (I==400) THEN
                SUB_FILE(I)='T0400.TXT'
        END IF
END SUBROUTINE NAME_OF_FILE
SUBROUTINE  NAME_OF_FILE_BACK(SUB_FILE,I)
        INTEGER I
        CHARACTER*20 SUB_FILE(400)
        IF (I==1) THEN
                SUB_FILE(I)='TB0001.TXT'

        ELSE IF (I==2) THEN
                SUB_FILE(I)='TB0002.TXT'
        ELSE IF (I==3) THEN
```

```
                SUB_FILE(I)='TB0003.TXT'
        ELSE IF (I==397) THEN
                SUB_FILE(I)='TB0397.TXT'
        ELSE IF (I==398) THEN
                SUB_FILE(I)='TB0398.TXT'
        ELSE IF (I==399) THEN
                SUB_FILE(I)='TB0399.TXT'
        ELSE IF (I==400) THEN
                SUB_FILE(I)='TB0400.TXT'
        END IF
END SUBROUTINE NAME_OF_FILE_BACK
```

ภาคผนวก ข
ตัวอย่างการคำนวณมวลและพลังงาน

<u>วิธีคำนวณหาอัตราการใช้เชื้อเพลิงและพลังงาน</u>

นำอัตราการไหลก๊าซหุงต้ม.ใช้ในการเผาแต่ละรอบมาคำนวณการใช้ก๊าซหุงต้มต่อนาทีในแต่ละช่วงของความดัน โดยแบ่งระดับการใช้ก๊าซหุงต้ม. เป็น 4 ระดับคือ

- ปริมาณการใช้ก๊าซหุงต้ม. ต่อนาที ที่ความดันก๊าซหุงต้ม. 0.2 $^{kg_f}/_{cm^2}$ สำหรับการจุดหัวพ่นไฟครึ่งหนึ่งของจำนวนหัวพ่นไฟทั้งหมดในเตาเซรามิค (P_1)
- ปริมาณการใช้ก๊าซหุงต้ม. ต่อนาที ที่ความดันก๊าซหุงต้ม. 0.2 $^{kg_f}/_{cm^2}$ สำหรับการจุดหัวพ่นไฟทั้งหมดภายในเตาเผาเซรามิค (P_2)
- ปริมาณการใช้ก๊าซหุงต้ม. ต่อนาที ที่ความดันก๊าซหุงต้ม. 0.25 $^{kg_f}/_{cm^2}$ สำหรับการจุดหัวพ่นไฟทั้งหมดภายในเตาเผาเซรามิค (P_2)
- ปริมาณการใช้ก๊าซหุงต้ม. ต่อนาที ที่ความดันก๊าซหุงต้ม. 0.3 $^{kg_f}/_{cm^2}$ สำหรับการจุดหัวพ่นไฟทั้งหมดภายในเตาเผาเซรามิค (P_3)

จากการวิเคราะห์ได้ค่าดังต่อไปนี้

P_1 = 0.01133 $^{kg}/_{min}$

P_2 = 0.02266 $^{kg}/_{min}$

P_3 = 0.02322 $^{kg}/_{min}$

P_4 = 0.030564 $^{kg}/_{min}$

นำไปคูณกับค่าเอนทาลปีจะได้อัตราการถ่ายเทพลังงานที่มวลก๊าซหุงต้มพาเข้าสู่ปริมาตรควบคุม

<u>การคำนวณหาอัตราการไหลอากาศที่ใช้ในการเผาไหม้และอัตราการไหลของก๊าซเสียชนิดต่างๆ</u>

จากสมการที่ (3-9) นำค่าเปอร์เซ็นก๊าซออกซิเจนในก๊าซเสียที่วัดได้มาคำนวณหาค่าอัตราส่วนอากาศที่ใช้จริงต่ออัตราส่วนอากาศตามทฤษฎี จากนั้นนำไปคูณเพื่อหาอัตราการไหลของอากาศที่ใช้ ได้เป็น $^{kg}/_{min}$

ตัวอย่างการคำนวณ

สมมุติให้เปอร์เซ็นก๊าซออกซิเจนในก๊าซเสีย = 16.38

$$\therefore \text{อัตราส่วนอากาศที่ใช้จริงต่ออัตราส่วนอากาศตามทฤษฎี} = \frac{\left(5.6 - 2.2 \times \frac{\%O_2}{100}\right)}{\left(5.6 - 26.656 \times \frac{\%O_2}{100}\right)}$$

$$= 4.2$$

นำไปใส่ในสมการ (3-8) จะได้

$$0.6C_3H_8 + 0.4C_4H_{10} + 23.52(O_2+3.76N_2) \rightarrow 3.4CO_2 + 4.4\ H_2O + 88.43N_2 + 17.92O_2$$

จากนั้นนำปริมาณก๊าซหุงต้ม. ที่ได้ไปคำนวณหาปริมาณอากาศและปริมาณก๊าซเสียแต่ละชนิด สมมุติให้ปริมาณเชื้อเพลิงในช่วงจุด 4 หัว ความดัน 0.2 $^{kg_f}/_{cm^2}$ (P_2) ∴ ปริมาณการใช้แก๊สเชื้อเพลิงคือ 0.02266 $^{kg}/_{min}$

ก๊าซหุงต้ม 1 โมล ใช้อากาศ 23.52 โมล

ก๊าซหุงต้ม. 1 โมลจะได้องค์ประกอบก๊าซเสียดังต่อไปนี้

ก๊าซคาร์บอนไดออกไซด์(CO_2)	3.4	โมล
ไอน้ำแห้ง(H_2O)	4.4	โมล
ก๊าซไนโตรเจน(N_2)	88.43	โมล
ก๊าซออกซิเจน(O_2)	17.92	โมล

ก๊าซหุงต้ม	1 โมล	มี	49.6	กรัม
อากาศ	1 โมล	มี	137.28	กรัม
ก๊าซคาร์บอนไดออกไซด์	1 โมล	มี	44	กรัม
ก๊าซออกซิเจน	1 โมล	มี	32	กรัม
ก๊าซไนโตรเจน	1 โมล	มี	28	กรัม
ไอน้ำแห้ง	1 โมล	มี	18	กรัม

ใช้ก๊าซหุงต้ม. 0.02266 $^{kg}/_{min}$ ∴ จะใช้อากาศ $= \frac{0.02266 \times 137.28}{49.6} \times 23.52$

$= 1.4751 \ ^{kg}/_{min}$

ใช้ก๊าซหุงต้ม. 0.02266 $^{kg}/_{min}$ ∴ จะได้ก๊าซคาร์บอนไดออกไซด์ $= \frac{0.02266 \times 44}{4.6} \times 3.4$

$= 0.068345 \ ^{kg}/_{min}$

ใช้ก๊าซหุงต้ม 0.02266 $^{kg}/_{min}$ ∴ จะได้ไอน้ำแห้ง $= \frac{0.02266 \times 18}{49.6} \times 4.4$

$= 0.036183 \ ^{kg}/_{min}$

ใช้ก๊าซหุงต้ม 0.02266 $^{kg}/_{min}$ ∴ จะได้ก๊าซไนโตรเจน $= \frac{0.02266 \times 28}{4.6} \times 88.43$

$= 1.13119 \ ^{kg}/_{min}$

ใช้ก๊าซหุงต้ม 0.02266 $^{kg}/_{min}$ ∴ จะได้ก๊าซออกซิเจน $= \frac{0.02266 \times 32}{49.6} \times 17.92$

$= 0.261979 \ ^{kg}/_{min}$

จากนั้นนำอัตราการไหล อากาศ ก๊าซคาร์บอนไดออกไซด์ ไอน้ำแห้ง ก๊าซไนโตรเจน และก๊าซออกซิเจน ไปคูณกับระยะเวลาที่ใช้ในช่วงเวลาแต่ที่ความดันก๊าซหุงต้มต่างๆกัน ก็จะได้ปริมาณอากาศที่ใช้และปริมาณก๊าซเสียชนิดต่างๆออกมา

การหาพลังงานที่เข้าสู่เตาเผาเซรามิค

นำอัตราการไหลของก๊าซหุงต้มและอากาศมาคำนวณหาพลังงานที่เข้าสู่ปริมาตรควบคุมโดยการนำอัตราการไหลมาคูณกับเอนทาลปี เช่น

ที่ความดันก๊าซหุงต้ม 0.2 $^{kg_f}/_{cm^2}$ จะได้อัตราไหลของก๊าซหุงต้มคือ 0.02266 $^{kg}/_{\min}$ ได้อัตราการไหลของอากาศที่ 1.4751 $^{kg}/_{\min}$ นำมาคำนวณอัตราพลังงานที่เข้าสู่ปริมาตรควบคุมได้เป็น

ที่อุณหภูมิ 25 $^{\circ}C$ เอนทาลปีของอากาศมีค่า 298.6 $^{kJ}/_{kg}$

เอนทาลปีของก๊าซหุงต้มมีค่า -2288 $^{kJ}/_{kg}$

$\therefore$ มีอัตราพลังงานเข้าสู่ปริมาตร 388.619 $^{kJ}/_{\min}$

การหาพลังงานที่ออกจากเตาเผาเซรามิค

นำอัตราการไหลของก๊าซเสียชนิดต่างๆมาคูณกับค่าเอนทาลปีที่อุณหภูมิของก๊าซเสียก็ได้อัตราการถ่ายเทพลังงานสูญเสียไปกับก๊าซเสีย

ตัวอย่าง สมมติให้อุณหภูมิก๊าซเสีย 105 $^{\circ}C$ และความดันก๊าซหุงต้ม. 0.20 $^{kg_f}/_{cm^2}$ จุดหัวพ่นไฟทั้งหมด จะได้

ค่าเอนทาลปีของก๊าซคาร์บอนไดออกไซด์ที่อุณหภูมิ 105 $^{\circ}C$ คือ	-8871	$^{kJ}/_{kg}$
ค่าเอนทาลปีของก๊าซออกซิเจนที่อุณหภูมิ 105 $^{\circ}C$ คือ	73.9	$^{kJ}/_{kg}$
ค่าเอนทาลปีของไอน้ำแห้งที่อุณหภูมิ 105 $^{\circ}C$ คือ	-13272	$^{kJ}/_{kg}$
ค่าเอนทาลปีของก๊าซไนโตรเจนที่อุณหภูมิ 105 $^{\circ}C$ คือ	83.34	$^{kJ}/_{kg}$
ค่าเอนทาลปีของก๊าซคาร์บอนไดออกไซด์ที่อุณหภูมิ 25 $^{\circ}C$ คือ	-8940	$^{kJ}/_{kg}$
ค่าเอนทาลปีของก๊าซออกซิเจนที่อุณหภูมิ 25 $^{\circ}C$ คือ	0	$^{kJ}/_{kg}$
ค่าเอนทาลปีของไอน้ำแห้งที่อุณหภูมิ 25 $^{\circ}C$ คือ	-13423	$^{kJ}/_{kg}$
ค่าเอนทาลปีของก๊าซไนโตรเจนที่อุณหภูมิ 25 $^{\circ}C$ คือ	0	$^{kJ}/_{kg}$
อัตราการไหลของก๊าซคาร์บอนไดออกไซด์คือ	0.068345	$^{kg}/_{\min}$
อัตราการไหลของก๊าซออกซิเจนคือ	0.261979	$^{kg}/_{\min}$

อัตราการไหลของไอน้ำแห้งคือ 0.036183 kg/min

อัตราการไหลของก๊าซไนโตรเจนคือ 1.13119 kg/min

จากสมการในบทที่ 3 จะได้

อัตราถ่ายเทพลังงานที่สูญเสียผ่านก๊าซคาร์บอนไดออกไซด์ 4.7158 kJ/min

อัตราถ่ายเทพลังงานที่สูญเสียผ่านก๊าซออกซิเจนคือ 19.3602 kJ/min

อัตราถ่ายเทพลังงานที่สูญเสียผ่านไอน้ำแห้งคือ 5.4636 kJ/min

อัตราถ่ายเทพลังงานที่สูญเสียผ่านก๊าซไนโตรเจนคือ 94.2734 kJ/min

ประวัติผู้เขียนวิทยานิพนธ์

นาย ธนวรรธน์ รัตนปริคณน์ .เกิดวันที่ 14 มิถุนายน พ.ศ. 2519 ที่โรงพยาบาลมหาราชนครศรีธรรมราช นครศรีธรรมราช สำเร็จการศึกษาปริญญาตรีวิศวกรรมศาสตร์บัณฑิต สาขาวิศวกรรมเครื่องกล ภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ ในปีการศึกษา 2540 ระหว่างการศึกษาได้รับการสนับสนุนการวิจัยจากบัณฑิตวิทยาลัย และกองทุนอนุรักษ์พลังงาน