

```
SUBROUTINE RESPON_FEA : CALCULATE TOATAL MASS;
: CALCULATE MAXIMUM PRINCIPAL STRESS;
: CALCULATE CORRESPONDING PRINCIPAL STRAIN;
: CALL CMM SUBROUTINE TO OBTAIN FATIGUE CRACK-INITIATION LIFE;
: FOR CURRENT GAUSS POINT/DESIGN POINT.
=====
Copyright (c) 2010-2014 [Xuchun Ren], All Rights Reserved.
SUBROUTINE RESPON_FEA(X0,Y,IFLAG_Y)
USE DEFINT
use cmmLib
use IFPORT
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
IMPLICIT NONE

REAL*8 X(30),Z(30),XMU(30),SIG(30), NMU(30),NSIG(30),DX(30),
1 X0(30), XL(30), XU(30), Y2(10)
REAL*8 Y, R_NF, R_NC
INTEGER NQ, N_ITER, IITER1, IITER_TOTAL, I_SQP
COMMON /SR02/ XMU,SIG, NMU, NSIG, NQ, N
COMMON /XR01/ N_F, N_GF, N_GC, N_EF,N_EC
INTEGER N_PFEA, N_FEA, NO_GAUSS
COMMON /XR05/ IITER1, IITER1, N_PFEA, N_FEA, IITER_TOTAL, I_SQP
COMMON /XR06/ NO_GAUSS
COMMON /XR09/ I_EVAL

INTEGER(2) n1, n2, n3, n4, n5, n6, n7, n8,n9, status,
1 MESH_ALGORITHM,M, I_FLAG_SAME_GAUSS
CHARACTER(320) buf, FND,FNP,FN_INP, FN_HM, FN_MASS, FN_TCL,
1 FN_BAT, FN_BAT0, FN_INP0, FN_HM0, FN_MASS0, FN_PY0, FN_PY,
2 FN_ODB, FN_ODB0, FN_MISES, FN_MISES0, JOBNAME, FUNC_NAME,
3 FN_PRT, FN_EAF, FN_ENF, FN_PAR, FN_GEO, FN_CL, FN_HD, FN_CAE,
4 FN_D5_D6
CHARACTER(320) MSG,MSG1,IDR,GDR, ELSETNAME, CTMP,CTMP1,CTMP2,
1 IDRO
CHARACTER(320) PATH, NAME, EXE_HM, EXE_ABAQUS
CHARACTER(320) WORKFOLDER
CHARACTER*320 CMD_TCL, CMD_PY, CMD_DOS
LOGICAL ex
LOGICAL(4) l_result
REAL*8 TOTAL_MASS, MAX_MISES, MAX_PRIN, E(3)
REAL*8 A(30), RTEMP, RTEMP2
INTEGER N_RENUM, N_INP_OK, N_ODB_OK, N_INP_BEST
COMMON /XR07/ PATH, WORKFOLDER
COMMON /XR10/ I_FLAG_SAME_GAUSS
MSG = 'PROGRAM NBAS(<10) NGAUSS(<10) NVAR_PDD(1,2,3) FN
1 INQ(1,2,...)
c MSG1 = 'IID = 0(INDEPENDENT), 1(IID), 2(DEPENDENT) '
R_NC = 6.36*10.0**9.0
R_NC = 1.0*10.0**6.0

N_INP_OK = 0
N_ODB_OK = 0
N_INP_BEST = 0
PSTRESS = 800.0
PSTRESS = 2200.0
MSG = 'PROGRAM PATH NAME'
FN_TCL = '1r-16.tcl
FN_BAT0 = 'RUN_TCL.BAT'
EXE_HM = 'C:\Altairwin64\hw9.1\hm\bin\win64\hmbatch.exe '
EXE_ABAQUS = 'abaqus cae noGUI='
FN_INP0 = 'LeverArm.INP'
FN_HM0 = 'LeverArm.hm'
FN_MASS0 = 'LA.MASS.DAT'
FN_PY = 'submit-04.py'
FN_PY0 = 'submit-04.py'
ELSETNAME = '1r'
FN_ODB = '3H BRACKET.ODB'
c$$$ FN_ODB0 = '3H BRACKET.ODB'
c$$$ FN_MISES0 = 'maxPrin.DAT'
FN_MISES = 'maxPrin.DAT'
FN_PRT = '1r-model-09-intialdem-04-hole2.prt.1'
FN_EAF = '1r-model-09-intialdem-04-hole2.eaf'
FN_ENF = '1r-model-09-intialdem-04-hole2.enf.abq'
FN_PAR = '1r-model-09-intialdem-04-hole2.par.abq'
FN_GEO = 'aba-geo-06.py'
FN_CL = 'aba-cl-03.py'
FN_CAE = '3cylinders-41r-05.cae'
FN_D5_D6 = 'D5_D6.DAT'

PI = 3.1415926
Z = 0.0
DO I = 1,N
Z(I) = NMU(I) + NSIG(I)*X0(I)
ENDDO
C ----- CALCULATE TRUNCATED DISTANCE FROM NMU AND NSIG -----
A = 0.0D0
A(1:2) = (/2.0, 2.0, 5.0, 2.0, 10.0, 10.0, 5.0, 5.0,
1 5.0, 5.0, 5.0, 5.0, 5.0, 2.0, 5.0, 2.0,
2 5.0, 5.0, 5.0, 5.0, 5.0, 5.0/)

RTEMP = (NMU(5)-2.0*NMU(6) - 60.0)/2.0
A(5) = MIN(RTEMP,A(5))
A(6) = MIN(RTEMP/2.0, A(6))

RTEMP = (625.0-1136.0)/(236.0-694.0)
RTEMP2 = ((NMU(7)-694.0)*RTEMP - (NMU(8)-1136.0))/2.0
A(7) = MIN(A(7), RTEMP2/RTEMP)
A(8) = MIN(A(8), RTEMP2)

RTEMP2 = ((NMU(13)-236.0)*RTEMP - (NMU(14)-635.0))/2.0
A(13) = MIN(A(13), RTEMP2/RTEMP)
A(14) = MIN(A(14), RTEMP2)

RTEMP = (814.0-500.0)/(760.0-252.0)
RTEMP2 = ((NMU(15)-252.0)*RTEMP - (NMU(16)-500.0))/2.0
A(15) = MIN(A(15), RTEMP2/RTEMP)
A(16) = MIN(A(16), RTEMP2)

RTEMP2 = ((NMU(23)-760.0)*RTEMP - (NMU(24)-814.0))/2.0
A(23) = MIN(A(23), RTEMP2/RTEMP)
A(24) = MIN(A(24), RTEMP2)

DO I = 1,N
IF (A(I) .LE. 0.4) THEN
PRINT*, 'A(I) < 0.4, I = ', I
PRINT*, 'A(I) = ', A(I)
PRINT*, 'NMU(I) = ', NMU(I)
A(I) = 2.5
ENDIF
ENDDO

CALL ToTruncatedG(Z, A, X)

IF (IFLAG_Y .EQ. 1) THEN
IF (I_FLAG_SAME_GAUSS .EQ. 0) THEN
FUNC_NAME = 'OBJ'
ELSE IF (I_FLAG_SAME_GAUSS .EQ. 1) THEN
FUNC_NAME = 'OBJ-CONST'
ELSE
PRINT*, 'WRONG I_FLAG_SAME_GAUSS!'
STOP
ENDIF
ELSEIF (IFLAG_Y .EQ. 2) THEN
IF (I_FLAG_SAME_GAUSS .EQ. 0) THEN
FUNC_NAME = 'CONST'
ELSE IF (I_FLAG_SAME_GAUSS .EQ. 1) THEN
FUNC_NAME = 'OBJ-CONST'
ELSE
PRINT*, 'WRONG I_FLAG_SAME_GAUSS!'
STOP
ENDIF
ELSE
PRINT*, 'WRONG IFLAG_Y!'
STOP
ENDIF

WRITE(IDRO, '(T1, I10)') I_SQP
WRITE(IDR, '(T1, I10)') IITER
IDRO = ADJUSTL(IDRO)
IDR = ADJUSTL(IDR)
IDR = TRIM(WORKFOLDER)//'\'/TRIM(IDR)//'\'/TRIM(IDRO)
INQUIRE (DIRECTORY = TRIM(IDR), EXIST=ex)
IF (.NOT. ex) THEN
l_result = MAKEDIRQ (TRIM(IDR))
IF (.NOT. l_result) THEN
PRINT*, 'FAIL TO CREATE FOLDER :', TRIM(IDR)
STOP
ENDIF
ENDIF

WRITE(GDR,*) I_EVAL
WRITE(CTMP,*) NO_GAUSS
GDR = ADJUSTL(GDR)
GDR = TRIM(IDR)//'\'/TRIM(GDR)//'\'/TRIM(FUNC_NAME)

INQUIRE (DIRECTORY = TRIM(GDR), EXIST=ex)
IF (.NOT. ex) THEN
l_result = MAKEDIRQ (TRIM(GDR))
IF (.NOT. l_result) THEN
PRINT*, 'FAIL TO CREATE FOLDER :', TRIM(GDR)
STOP
ENDIF
ENDIF

GDR = TRIM(GDR)//'\'/TRIM(ADJUSTL(CTMP))
INQUIRE (DIRECTORY = TRIM(GDR), EXIST=ex)
IF (.NOT. ex) THEN
l_result = MAKEDIRQ (TRIM(GDR))
IF (.NOT. l_result) THEN
PRINT*, 'FAIL TO CREATE FOLDER :', TRIM(GDR)
STOP
ENDIF
ENDIF

IF (IFLAG_Y .EQ. 2 .AND. I_FLAG_SAME_GAUSS .EQ. 1) GOTO 888
C----- COPY TCL, proE, PYTHON FILE -----
CMD_TCL = "copy '//TRIM(PATH)//'\'/TRIM(FN_TCL)/* "
1 //TRIM(GDR)
l_result = systemq (TRIM(CMD_TCL))
IF (.NOT. l_result) THEN
PRINT*, 'FAIL TO RUN COMMAND: ', TRIM(CMD_TCL)
STOP
ENDIF
CMD_TCL = "copy '//TRIM(PATH)//'\'/TRIM(FN_PRT)/* "
1 //TRIM(GDR)
l_result = systemq (TRIM(CMD_TCL))
IF (.NOT. l_result) THEN
PRINT*, 'FAIL TO RUN COMMAND: ', TRIM(CMD_TCL)
STOP
ENDIF
CMD_TCL = "copy '//TRIM(PATH)//'\'/TRIM(FN_EAF)/* "
1 //TRIM(GDR)
l_result = systemq (TRIM(CMD_TCL))
IF (.NOT. l_result) THEN
PRINT*, 'FAIL TO RUN COMMAND: ', TRIM(CMD_TCL)
STOP
ENDIF
CMD_TCL = "copy '//TRIM(PATH)//'\'/TRIM(FN_ENF)/* "
1 //TRIM(GDR)
l_result = systemq (TRIM(CMD_TCL))
IF (.NOT. l_result) THEN
PRINT*, 'FAIL TO RUN COMMAND: ', TRIM(CMD_TCL)
STOP
ENDIF
CMD_TCL = "copy '//TRIM(PATH)//'\'/TRIM(FN_PAR)/* "
1 //TRIM(GDR)
l_result = systemq (TRIM(CMD_TCL))
IF (.NOT. l_result) THEN
PRINT*, 'FAIL TO RUN COMMAND: ', TRIM(CMD_TCL)
STOP
ENDIF
CMD_TCL = "copy '//TRIM(PATH)//'\'/TRIM(FN_GEO)/* "
1 //TRIM(GDR)
l_result = systemq (TRIM(CMD_TCL))
IF (.NOT. l_result) THEN
PRINT*, 'FAIL TO RUN COMMAND: ', TRIM(CMD_TCL)
STOP
ENDIF
CMD_TCL = "copy '//TRIM(PATH)//'\'/TRIM(FN_CL)/* "
1 //TRIM(GDR)
l_result = systemq (TRIM(CMD_TCL))
IF (.NOT. l_result) THEN
PRINT*, 'FAIL TO RUN COMMAND: ', TRIM(CMD_TCL)
STOP
ENDIF
CMD_TCL = "copy '//TRIM(PATH)//'\'/TRIM(FN_CAE)/* "
1 //TRIM(GDR)
l_result = systemq (TRIM(CMD_TCL))
IF (.NOT. l_result) THEN
PRINT*, 'FAIL TO RUN COMMAND: ', TRIM(CMD_TCL)
STOP
ENDIF
C----- CREATE PATH_G.DAT, D5_D6.DAT 4HM.DAT FILE -----
M = 0
FND = TRIM(GDR)//'\'/PATH_G.DAT'
IF (M .GE. 1) THEN
WRITE(CTMP,*) M-1
CTMP = TRIM(GDR)//'\'/PATH_G//TRIM(ADJUSTL(CTMP))//'.DAT'
CMD_DOS = 'move '//TRIM(FND)//'\'/TRIM(CTMP)
l_result = systemq (TRIM(CMD_DOS))
IF (.NOT. l_result) THEN
PRINT*, 'FAIL TO RUN COMMAND: ', TRIM(CMD_DOS)
STOP
ENDIF
ENDIF
OPEN(201,FILE=trim(FND),STATUS='NEW')
DO J = 1,N
WRITE(CTMP,*) X(J)
WRITE(201, '(A)') TRIM(ADJUSTL(CTMP))
ENDDO
M = M+1
WRITE(201, '(A)') TRIM(ADJUSTL(GDR))//'\
CLOSE(201)
FN_HD = TRIM(GDR)//'\'/4HM.DAT'
FN_D5_D6 = TRIM(GDR)//'\'/D5_D6.DAT'
OPEN(202,FILE=trim(FN_HD),STATUS='NEW')
OPEN(301,FILE=trim(FN_D5_D6),STATUS='NEW')
WRITE(CTMP,*) X(5)
WRITE(202, '(A)') TRIM(ADJUSTL(CTMP))
WRITE(301, '(A)') TRIM(ADJUSTL(CTMP))
WRITE(CTMP,*) X(6)
WRITE(202, '(A)') TRIM(ADJUSTL(CTMP))
WRITE(301, '(A)') TRIM(ADJUSTL(CTMP))
FN_INP = TRIM(GDR)//'\'/TRIM(FN_INP0)
FN_HM = TRIM(GDR)//'\'/TRIM(FN_HM0)
FN_MASS = TRIM(GDR)//'\'/TRIM(FN_MASS0)
WRITE(202, '(A)') TRIM(ADJUSTL(FN_INP))
WRITE(202, '(A)') TRIM(ADJUSTL(FN_HM))
WRITE(202, '(A)') TRIM(ADJUSTL(FN_MASS))
WRITE(301, '(A)') TRIM(ADJUSTL(GDR))//'\
CLOSE(202)

C----- RUN PYTHON FILE -----
FN_BAT = TRIM(GDR)//'\'/TRIM(FN_BAT0)
OPEN(202,FILE=trim(FN_BAT),STATUS='NEW')
WRITE(202,*) , 'cd '//TRIM(GDR)
1 TRIM(FN_TCL)
CLOSE(202)
CMD_TCL = ADJUSTL(GDR(1:2))//'' & '//cd '//TRIM(GDR)//'' & '//
1 'TRIM(EXE_ABAQUS)//TRIM(FN_CL)
l_result = systemq (TRIM(CMD_TCL))
IF (.NOT. l_result) THEN
PRINT*, 'FAIL TO RUN COMMAND: ', TRIM(CMD_TCL)
STOP
ENDIF
INQUIRE (FILE = TRIM(FN_INP), EXIST=ex)
IF (.NOT. ex) THEN
IF (M .GE. 5) THEN
STOP
ELSE
PRINT*, 'FAIL TO GENERATE INP FILE: ', TRIM(FN_INP)
PRINT*, 'CHANGE MESH SIZE, TRYING .... '
STOP
ENDIF
GOTO 201
ENDIF
ENDIF
get mass -----
N_INP_OK = 1
OPEN(204,FILE=trim(FN_MASS),STATUS='OLD')
READ(205, '(F16.8)') TOTAL_MASS
CLOSE(204)
PRINT*, '-----TOTAL MASS -----'
PRINT*, TOTAL_MASS

N_PFEA = N_PFEA + 1
IF (IFLAG_Y .EQ. 1) THEN
Y = TOTAL_MASS
RETURN
ENDIF
888----- COPY PYTHON FILE -----
CMD_PY = "copy '//TRIM(PATH)//'\'/TRIM(FN_PY0)/* "
1 //TRIM(GDR)
l_result = systemq (TRIM(CMD_PY))
IF (.NOT. l_result) THEN
PRINT*, 'FAIL TO RUN COMMAND: ', TRIM(CMD_PY)
STOP
ENDIF
C----- CREATE PATH_X.DAT FILE -----
FND = TRIM(GDR)//'\'/PATH_X.DAT'
INQUIRE (FILE = TRIM(FND), EXIST=ex)
IF (.NOT. EXIST) THEN
CMD_PY = "del '//TRIM(FND)
l_result = systemq (TRIM(CMD_PY))
IF (.NOT. l_result) THEN
PRINT*, 'FAIL TO RUN COMMAND: ', TRIM(CMD_PY)
STOP
ENDIF
ENDIF
OPEN(203,FILE=trim(FND),STATUS='NEW')
WRITE(CTMP,*) I_SQP ---- FOR AIJK
WRITE(CTMP,*) IITER
JOBNAME = TRIM(ADJUSTL(CTMP))//'\'/TRIM(ADJUSTL(CTMP1))
WRITE(203, '(A)') TRIM(JOBNAME)
FN_INP = TRIM(GDR)//'\'/TRIM(FN_INP0)
FN_ODB = TRIM(GDR)//'\'/TRIM(JOBNAME)'.ODB'
FN_MISES = TRIM(GDR)//'\'/TRIM(FN_MISES0)
WRITE(203, '(A)') TRIM(ADJUSTL(FN_INP))
WRITE(203, '(A)') TRIM(ADJUSTL(FN_ODB))
WRITE(203, '(A)') TRIM(ADJUSTL(FN_MISES))
WRITE(203, '(A)') TRIM(ELSETNAME)
CLOSE(203)
INQUIRE (FILE = TRIM(FN_ODB), EXIST=ex)
IF (.NOT. EXIST) THEN
CMD_PY = "del '//TRIM(FN_ODB)
l_result = systemq (TRIM(CMD_PY))
IF (.NOT. l_result) THEN
PRINT*, 'FAIL TO RUN COMMAND: ', TRIM(CMD_PY)
STOP
ENDIF
ENDIF
run PYTHON FILE -----
FN_BAT = TRIM(GDR)//'\'/TRIM(FN_BAT0)
OPEN(202,FILE=trim(FN_BAT),STATUS='NEW')
WRITE(202,*) , 'cd '//TRIM(GDR)
1 TRIM(FN_TCL)
CLOSE(202)
CMD_PY = GDR(1:2)//'' & '//cd '//TRIM(GDR)//'' & '//
1 'TRIM(EXE_ABAQUS)//TRIM(FN_PY)
l_result = systemq (TRIM(CMD_PY))
IF (.NOT. l_result) THEN
PRINT*, 'FAIL TO RUN COMMAND: ', TRIM(CMD_PY)
STOP
ENDIF
INQUIRE (FILE = TRIM(FN_MISES), EXIST=ex)
IF (.NOT. ex) THEN
PRINT*, 'FAIL TO CREAT MISES FILE: ', TRIM(FN_MISES)
STOP
ENDIF
get max mises -----
OPEN(205,FILE=trim(FN_MISES),STATUS='OLD')
READ(205, '(F16.8)') MAX_PRIN
PRINT*, '-----MAX PRIN -----'
PRINT*, MAX_PRIN

READ(205, '(F16.8)') MAX_MISES
PRINT*, '-----MAX MISES -----'
PRINT*, MAX_MISES

READ(205, '(F16.8)') E(1)
PRINT*, '-----E1 -----'
PRINT*, E(1)

READ(205, '(F16.8)') E(2)
PRINT*, '-----E2 -----'
PRINT*, E(2)

READ(205, '(F16.8)') E(3)
PRINT*, '-----E3 -----'
PRINT*, E(3)

Y2(1) = MAX_PRIN
Y2(2) = MAX_MISES
Y2(3) = E(1)
Y2(4) = E(2)
Y2(5) = E(3)

CALL CMM(MAX_PRIN, E, R_NF, GDR)

Y = R_NF - R_NC
PRINT*, '-----y -----'
PRINT*, Y

N_FEA = N_FEA + 1

RETURN
END
```