```
C       =================================================================
c       Multi-point single-step PDD method for
C       EXAMPLE 3: shape design of a three-hole bracket
C
C
C
C
C       N = 9
C       Y0 = total mass of design
C     Y1 = Sy - Sigma_max
c     Sy            :   uniaxial yield strength
C     Sigma_max     :   maximum von Mises stress
c
C       X(1:9) -> TRUNCATED GAUSSIAN DISTRIBUTION
C
C     -------  command line parameters to run the code   -------------
C     PROGRAM.EXE N_FLAG_AIJK NBAS(<10) NGAUSS(<10)  NVAR_PDD(1,2,3) FN IBOX(0,1,2,...) PATH NAME
C     FOR EXAMPLE, UNIVARIATE, M=1 CASE USING THE FOLLOWING COMMAND LINE:
C               PROGRAM.EXE 1 1 2 1 FN 2 D:\3HB 3HB-S1M1-B2
C
C****************************************************************************************
c     Copyright (c) 2010-2014 [Xuchun Ren], All Rights Reserved.

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION X(200),XL(200),XU(200),G(200), OBJ(200), G1(200),
     *WK(800),IWK(200),RPRM(20),IPRM(20), XSIG_IN(200), X_IT(200,200),
     * OBJ_IT(200,200), CONS_IT(200,200),OBJ_IT_EXC(200,200),
     *  CONS_IT_EXC(200,200), XT(200), OBJ_EXC(200), G_EXC(200),
     * XLAST(200), X_IT1(200,200),X_FEASIBLE(200), X_SQP(2000,200),
     *   OBJ_SQP(2000,200), G_SQP(2000,200), X_CENTRAL(200),
     * OBJ_IT1(200,200), CONS_IT1(200,200),OBJ_IT_EXC1(200,200),
     *  CONS_IT_EXC1(200,200), XT1(200), OBJ_EXC1(200), G_EXC1(200),
     *  AA(2,10),XL0(200),XU0(200),X0(200),XL01(200),XU01(200),X00(200),
     *  DV_TRUST(200), DSA_SENS_C(200), SF_SENS_C_FEAS(200),
     *  G_FEAS(200), G_DSA1(200), G_DSA2(200),XL02(200),XU02(200),
     *  F_FEAS(200), X_UT(200), I_INCR_DBOX(200),IHISTORY_TR(2000,200),
     * G_AIJK_SQP(200),X_FEASIBLE_LAST(200),G_PDD_SQP(200),X_N(200),
     * G_SQP_AIJK(2000,200),X_UNTR(200),DELTA_X(200),DELTA_X2(200)

      REAL*8 XMU(20),SIG(20),NMU(20),NSIG(20),RTMP,RTMP2, X_DV_DSA,
     1        OBJ_AIJK_LAST,   OBJ_AIJK_LAST2

      DIMENSION X_AIJK_LAST(200), DBOX(200),X_AIJK_LAST2(200),
     1   X_AIJK_LAST2T(200),X_AIJK_LAST_T(200)
      COMMON /SR02/ XMU,SIG, NMU, NSIG, NYQ, N

      COMMON /XR01/ N_F, N_GF, N_GC, N_EF,N_EC, N_EF1, N_EC1,
     1       N_EF2, N_EC2
      INTEGER N_PREFEA, N_FEA, NO_GAUSS, N_FLAG_AIJK, I_EVAL, I_SQP,
     2     IITER_TOTAL, I_DSA, I_DSA_DV
      COMMON /XR05/ IITER, IITER1, N_PREFEA, N_FEA, IITER_TOTAL, I_SQP

      COMMON /XR04/ NBAS_R, NGAUSS_R, NVAR
      COMMON /XR06/ NO_GAUSS

        INTEGER(2) n1, n2, n3, n4, n5, n6, n7, n8,n9, status
        CHARACTER(320) buf, FN
        CHARACTER(80) MSG,MSG1

        CHARACTER(320) PATH, NAME
        CHARACTER(320) WORKFOLDER
      COMMON /XR07/ PATH, WORKFOLDER
      COMMON /XR09/ I_EVAL

      COMMON /XR10/ FN

        LOGICAL ex
        LOGICAL(4) l_result

        CHARACTER(320)  FN_TCL,FN_PY0
        INTEGER I_RECIPROCAL,I_FLAG_INCRD,I_GRD_NEG(200)
        COMMON /XR11/ XL01, XU01
        COMMON /XR20/ C1_SCAL, I_RECIPROCAL

        EPS_CONV = 0.1
        X_FEASIBLE = 0.0
        X_FEASIBLE_LAST = -10**6
        I_FLAG_INCRD = 0
        IHISTORY_TR = 0.0

        I_INCR_DBOX = 0
        X_DV_DSA = 0.0
        DV_TRUST = 0.0
        DSA_SENS_C = 0.0
        SF_SENS_C_FEAS = 0.0
        F_FEAS = 0.0
        G_FEAS = 0.0
        I_DSA = 0
        I_DSA_DV = 1
        DBOX_MIN = 2.0
        G_DSA1 = 0.0
        G_DSA2 = 0.0
        G_PDD_SQP = 0.0

        IWRITE = 401

        ISCAL = 0
        DABOBJ = 0.001*10
        DELOBJ = 0.01
        DOBJ1 = 0.05
        DOBJ2 = 0.01
        CT = -0.1
        CTMIN = 0.03

        RMVLMZ = 0.4

C       NO AIJK INVOLVED
        I_RECIPROCAL = 0
        C1_SCAL = 8.0
        I_EVAL = 0
C        N_FLAG_AIJK = 0

        FN_TCL = '3hole_bracket-17.tcl'
        FN_PY0 = 'submit-07.py'

        MSG = 'PROGRAM NBAS(<10) NGAUSS(<10) NVAR_PDD(1,2,3) FN
     1   IBOX(0,1,2,...) PATH NAME '
c        MSG1 = 'IID = 0(INDEPENDENT), 1(IID), 2(DEPENDNENT) '

        N_PREFEA = 0
        N_FEA = 0

        n1 = 1
        n2 = 2
        n3 = 3
        n4 = 4
        n5 = 5
        n6 = 6
        n7 = 7
        n8 = 8

        PI = 3.1415926

        CALL GETARG(n1, buf, status)
        IF (status .EQ. -1) THEN
        PRINT*, 'WRONG N_FLAG_AIJK!'
        PRINT*, MSG
C        PRINT*, MSG1
        STOP
        ENDIF
        Read( buf, '(i10)' )  N_FLAG_AIJK
        IF (N_FLAG_AIJK .NE. 0 .AND. N_FLAG_AIJK .NE. 1) THEN
        PRINT*, 'WRONG N_FLAG_AIJK = ', N_FLAG_AIJK
        STOP
        ENDIF
        PRINT*, 'N_FLAG_AIJK = '
        WRITE (*,*) N_FLAG_AIJK

        CALL GETARG(n2, buf, status)
        IF (status .EQ. -1) THEN
        PRINT*, 'WRONG NBAS!'
        PRINT*, MSG
C        PRINT*, MSG1
        STOP
        ENDIF
        Read( buf, '(i10)' )  NBAS_R
        PRINT*, 'NBAS = '
        WRITE (*,*) NBAS_R

        CALL GETARG(n3, buf, status )
        IF (status .EQ. -1) THEN
        PRINT*, 'WRONG NGAUSS!'
        PRINT*, MSG
C        PRINT*, MSG1
        STOP
        ENDIF
        Read( buf, '(i10)' )  NGAUSS_R
        PRINT*, 'NGAUSS = '
        WRITE (*,*) NGAUSS_R

        CALL GETARG(n4, buf, status)
        IF (status .EQ. -1) THEN
        PRINT*, 'WRONG NVAR_PDD!'
        PRINT*, MSG
C        PRINT*, MSG1
        STOP
        ENDIF
        Read( buf, '(i10)' )  NVAR
        PRINT*, 'NVAR_PDD = '
        WRITE (*,*) NVAR

        CALL GETARG(n5, FN, status )
        IF (status .EQ. -1) THEN
        PRINT*, 'WRONG FILE NAME!'
        PRINT*, MSG
C        PRINT*, MSG1
        STOP
        ENDIF
        PRINT*, 'FILE NAME IS ---'
        WRITE (*,*) TRIM(FN)

        CALL GETARG(n6, buf, status )
        IF (status .EQ. -1) THEN
        PRINT*, 'WRONG FILE NAME!'
        PRINT*, MSG
C        PRINT*, MSG1
        STOP
        ENDIF
        Read( buf, '(i10)' )  IBOX
        PRINT*, 'IBOX = '
        WRITE (*,*) IBOX

        IF (NBAS_R .GE. 10) THEN
        PRINT*, 'WRONG NBAS_R (SHOULD BE < 10)!'
        PRINT*, MSG
        WRITE (*,*) NBAS_R
        STOP
        ENDIF

        IF (NGAUSS_R .GE. 10) THEN
        PRINT*, 'WRONG NGAUSS_R (SHOULD BE < 10)!'
        PRINT*, MSG
        WRITE (*,*) NGAUSS_R
        STOP
        ENDIF

        IF (NVAR .NE. 1 .AND. NVAR .NE. 2 .AND. NVAR .NE. 3) THEN
        PRINT*, 'WRONG NVAR (SHOULD BE 1,2,3)!'
        PRINT*, MSG
        WRITE (*,*) NVAR
        STOP
        ENDIF

C        IF (IBOX .LT. 0 ) THEN
C        PRINT*, 'WRONG IBOX (SHOULD BE >= 0)!'
C        PRINT*, MSG
C        WRITE (*,*) IBOX
C        STOP
C        ENDIF

        CALL GETARG(n7, PATH, status )
        IF (status .EQ. -1) THEN
        PRINT*, 'WRONG PATH!'
        PRINT*, MSG
C        PRINT*, MSG1
        STOP
        ENDIF
        PRINT*, 'PATH IS ---'
        WRITE (*,*) TRIM(PATH)

        INQUIRE (DIRECTORY = TRIM(PATH), EXIST=ex)
        IF (.NOT. ex) THEN
            PRINT*, 'directory: ', TRIM(PATH), '  NOT EXIST! '
            STOP
         ENDIF
        INQUIRE (FILE = TRIM(PATH)//'\'//TRIM(FN_TCL), EXIST=ex)
        IF (.NOT. ex) THEN
            PRINT*, 'TCL FILE: ', TRIM(PATH)//'\'//TRIM(FN_TCL),
     1      '  NOT EXIST! '
            STOP
         ENDIF
        INQUIRE (FILE = TRIM(PATH)//'\'//TRIM(FN_PY0), EXIST=ex)
        IF (.NOT. ex) THEN
            PRINT*, 'PYTHON FILE: ', TRIM(PATH)//'\'//TRIM(FN_PY0),
     1      '  NOT EXIST! '
            STOP
         ENDIF

        CALL GETARG(n8, NAME, status )
        IF (status .EQ. -1) THEN
        PRINT*, 'WRONG NAME!'
        PRINT*, MSG
C        PRINT*, MSG1
        STOP
        ENDIF
        PRINT*, 'NAME IS ---'
        WRITE (*,*) TRIM(NAME)

        WORKFOLDER = TRIM(PATH)//"\" //TRIM(NAME)

        INQUIRE (DIRECTORY = TRIM(WORKFOLDER), EXIST=ex)
        IF (ex) THEN
            PRINT*, 'directory: ',TRIM(WORKFOLDER) , '  ALREADY EXIST! '
            STOP
        ELSE
          l_result = MAKEDIRQQ (TRIM(WORKFOLDER))
          IF (.NOT. l_result) THEN
             PRINT*, 'FAIL TO CREATE FOLDER :', TRIM(WORKFOLDER)
             STOP
          ENDIF

        ENDIF

      OPEN(109,FILE=trim(FN)//'-ITERATION_PROCESS.TXT')
      OPEN(108,FILE=trim(FN)//'-AIJK.TXT')
      OPEN(IWRITE,FILE=trim(FN)//'-DOT.TXT')
      OPEN(110,FILE=trim(FN)//'-DBOX.TXT')
      OPEN(111,FILE=trim(FN)//'-SQP.TXT')
      OPEN(112,FILE=trim(FN)//'-DV_TRUST.TXT')
      OPEN(113,FILE=trim(FN)//'-DSA_SEN.TXT')
      OPEN(114,FILE=trim(FN)//'-I_XU_XL.TXT')
      OPEN(115,FILE=trim(FN)//'-ERROR_FLAG.TXT')
      OPEN(116,FILE=trim(FN)//'-G_DSA_SQP.TXT')
      OPEN(117,FILE=trim(FN)//'-DEBUG_DAT.TXT')
      N_F = 0
      N_GF = 0
      N_GC = 0
      N_EF = 0
      N_EC = 0
      N_EF1 = 0
      N_EC1 = 0
      N_EF2 = 0
      N_EC2 = 0
      X_IT = 0.0
      OBJ_IT = 0.0
      CONS_IT = 0.0
      OBJ_IT_EXC = 0.0
      CONS_IT_EXC = 0.0
      XT = 0.0
      OBJ_EXC = 0.0
      G_EXC = 0.0
      XLAST = 0.0
      OBJ_AIJK_LAST = 100000.0
      OBJ_AIJK_LAST2 = 100000.0
      X_AIJK_LAST = 0.0
      X_AIJK_LAST2 = 0.0

C     DEFINE NRWK, NRIWK.
      NRWK=800
      NRIWK=200
C     ZERO RPRM AND IPRM.
      DO 10 I=1,20
        RPRM(I)=0.0
10    IPRM(I)=0
      IPRM(1) = 1
      IPRM(2) = ISCAL
C     SET
      RPRM(1) = CT
      RPRM(2) = CTMIN
      RPRM(3)=DABOBJ
      RPRM(4)=DELOBJ
      RPRM(11) = RMVLMZ
C ------  IWRITE ---
      IPRM(5) = IWRITE
C     DEFINE METHOD,NDV,NCON.
C     SEQUENTIAL QUADRATIC PROGRAMMING METHOD.
      METHOD=3
C     NINE DESIGN VARIABLES.
      NDV=9
      N=NDV
C     ONE CONSTRAINT
      NCON=1
      XL = 0.
      XU = 0.
      XL0 = 0.
      XU0 = 0.
C     DEFINE BOUNDS AND INITIAL DESIGN.
      DO 20 I=1,NDV
C     INITIAL VALUES.
      X(I)=1.0
C     LOWER BOUNDS.
      XL(I)=1
C     UPPER BOUNDS
20    XU(I)=10

      XL(1)= 0.0
      XL(2)= 17.0
      XL(3)= 10.0
      XL(4)= 30.0
      XL(5)= 12.0
      XL(6)= 12.0
      XL(7)= 50.0
      XL(8)= -15.0
      XL(9)= -8.0

      XU(1)=14.0
      XU(2)=35.0
      XU(3)=30.0
      XU(4)=40.0
      XU(5)=30.0
      XU(6)=30.0
      XU(7)=140.0
      XU(8)=10.0
      XU(9)=15.0

      X(1) = 0.0
      X(2) = 30.0
      X(3) = 10.0
      X(4) = 40.0
      X(5) = 20.0
      X(6) = 20.0
      X(7) = 75.0
      X(8) = 0.0
      X(9) = 0.0

C      X(1) =                         13.9999
C      X(2) =                         17.0000
C      X(3) =                         28.5645
C      X(4) =                         30.0000
C      X(5) =                         12.0000
C      X(6) =                         12.0000
C      X(7) =                         117.831
C      X(8) =                         -14.009
C      X(9) =                         14.9999

      XL01 = XL
      XU01 = XU
      X00 = X

      IITER_TOTAL = 0
      CALL TRANSLATE_X (XL,NDV,X0, XL01, XU01, 1)
      XL = X0
      CALL TRANSLATE_X (XU,NDV,X0, XL01, XU01, 1)
      XU = X0
      CALL TRANSLATE_X (X,NDV,X0, XL01, XU01, 1)
      X = X0

      XL0= XL
      XU0 = XU

      I_FEASIBLE = 0
      I_FEASIBLE_LAST = 0
      I_FEASIBLE_CUR = 1

      IF (IBOX .GT. 0) THEN
      DBOX = (XU-XL)/IBOX
      ELSE IF (IBOX .LT. 0) THEN
      DBOX = (XU-XL)/ABS(IBOX)
      ENDIF
C     DEFINE IPRINT, MINMAX, INFO.
C     PRINT CONTROL.
      IPRINT=3
C     MINIMIZE
      MINMAX=-1

      X_SQP = 0.0
      OBJ_SQP = 0.0
      G_SQP = 0.0
      G_SQP_AIJK = 0.0

      I_SQP = 0
      N_VIOLANCE = 0
      I_DBOX_SMALL = 0
C     INITIALIZE INFO TO ZERO.
99    INFO=0
      I_FEAS_LAST2 = 0
      NDV = 9
      NCON = 1
      MINMAX=-1
      IPRINT=3
      IWRITE = 401

      ISCAL = 0
      IF (N_FLAG_AIJK .EQ. 2) THEN
      DABOBJ = 0.001*10*10*0.5**I_SQP
      DELOBJ = 0.01*10*0.5**I_SQP
      DOBJ1 = 0.05*10*0.5**I_SQP
      DOBJ2 = 0.01*10*0.5**I_SQP
      CT = -0.1*10*0.5**I_SQP
      CTMIN = 0.03*10*0.5**I_SQP
      ELSE

      DABOBJ = 0.001*10
      DELOBJ = 0.01
      DOBJ1 = 0.05
      DOBJ2 = 0.01
      CT = -0.1
      CTMIN = 0.03
      ENDIF

      RMVLMZ = 0.4

C     OPTIMIZE.
      X_IT = 0.0
      OBJ_IT = 0.0
      CONS_IT = 0.0
      OBJ_IT_EXC = 0.0
      CONS_IT_EXC = 0.0
      XT = 0.0
      OBJ_EXC = 0.0
      G_EXC = 0.0
      XLAST = 0.0
C      X_AIJK_LAST = 0.0
      NRWK=800
      NRIWK=200

C     ZERO RPRM AND IPRM.
      RPRM=0.0
      IPRM= 0
      IPRM(1) = 1
      IPRM(2) = ISCAL
C     SET
      RPRM(1) = CT
      RPRM(2) = CTMIN
      RPRM(3)=DABOBJ
      RPRM(4)=DELOBJ
      RPRM(11) = RMVLMZ
      IPRINT=3
C     MINIMIZE
      MINMAX=-1
      IPRM(5) = IWRITE

      IITER= 0
      IITER1= 0

      IFCALL =0
      IGFCALL = 0
      IGCCALL = 0
      XLAST = 0.0

      I_SQP = I_SQP + 1
      IF (I_SQP .EQ. 1) THEN
      X_SQP(I_SQP,:) = X00
      ELSE
      X_SQP(I_SQP,:) = X0
      ENDIF
      IF (I_SQP .GE. 2) THEN
      WRITE(111,'(I8, F20.10, F20.10)'),
     2      I_SQP-1, OBJ_SQP(I_SQP-1,1), G_SQP(I_SQP-1,1)
      ENDIF

      WRITE(111,'(I8, F20.10, F20.10, F20.10, F20.10, F20.10,
     1      F20.10, F20.10, F20.10, F20.10, F20.10, F20.10)'),
     2      I_SQP, OBJ1, G1(1),X_SQP(I_SQP,1:NDV)

      I_MAX_CHANGE = 0
      I_MAX2_CHANGE = 0
      X_MAX_CHANGE = 0.
      X_MAX2_CHANGE = 0.
      X_CHANGE = 0.
      X_N = X
C
C     X_FEASIBLE IS TRANSFORMED; X_AIJK_LAST2, X_AIJK_LAST, X_IT ARE UNTRANSFORMED
C
      IF (IBOX .GT. 0) THEN
      CALL TRANSLATE_X (X_AIJK_LAST2,NDV,X_AIJK_LAST2T, XL01, XU01, 1)
      CALL TRANSLATE_X (X_AIJK_LAST,NDV,X_AIJK_LAST_T, XL01, XU01, 1)

          WRITE(117,'(I8,A40)'),I_SQP,'---------- END2   -------'
          WRITE(117,'(A20, F20.10, F20.10, F20.10, F20.10, F20.10,
     1     F20.10, F20.10, F20.10,F20.10)'),
     2      'X', X(1:NDV)
          WRITE(117,'(A20, F20.10, F20.10, F20.10, F20.10, F20.10,
     1     F20.10, F20.10, F20.10,F20.10)'),
     2      'XL', XL(1:NDV)
          WRITE(117,'(A20, F20.10, F20.10, F20.10, F20.10, F20.10,
     1     F20.10, F20.10, F20.10,F20.10)'),
     2      'XU', XU(1:NDV)
          WRITE(117,'(A20, F20.10, F20.10, F20.10, F20.10, F20.10,
     1     F20.10, F20.10, F20.10,F20.10)'),
     2      'XL0', XL0(1:NDV)
          WRITE(117,'(A20, F20.10, F20.10, F20.10, F20.10, F20.10,
     1     F20.10, F20.10, F20.10,F20.10)'),
     2      'XU0', XU0(1:NDV)
      IF (I_FEASIBLE_LAST .EQ. 1 .AND. I_FEASIBLE_CUR .EQ. 1 ) THEN
          X_MAX_CHANGE=MAXVAL(ABS(X_FEASIBLE - X_FEASIBLE_LAST))
        N_BOUND = 0
        DO I = 1,NDV
            IF (X_FEASIBLE(I) .EQ. XL(I)) THEN
                IF (ABS(XL(I) - XL0(I)) .GE. 0.01) THEN
                N_BOUND = N_BOUND + 1
                ENDIF
            ELSE IF(X_FEASIBLE(I) .EQ. XU(I) ) THEN
                IF (ABS(XU(I) - XU0(I)) .GE. 0.01) THEN
                N_BOUND = N_BOUND + 1
                ENDIF
            ENDIF
         ENDDO

          IF (X_MAX_CHANGE < EPS_CONV .AND. N_BOUND .EQ. 0) THEN
          IITER = 1
          STOP
          ENDIF

      ELSE IF (I_FEASIBLE_LAST .EQ. 0 .AND. I_FEASIBLE_CUR .EQ. 1 ) THEN
          X_MAX_CHANGE=MAXVAL(ABS(X_FEASIBLE_LAST-X_FEASIBLE))
        N_BOUND = 0
        DO I = 1,NDV
            IF (X_FEASIBLE(I) .EQ. XL(I)) THEN
                IF (ABS(XL(I) - XL0(I)) .GE. 0.01) THEN
                N_BOUND = N_BOUND + 1
                ENDIF
            ELSE IF(X_FEASIBLE(I) .EQ. XU(I) ) THEN
                IF (ABS(XU(I) - XU0(I)) .GE. 0.01) THEN
                N_BOUND = N_BOUND + 1
                ENDIF
            ENDIF
         ENDDO

          IF (X_MAX_CHANGE .LT. 0.5*EPS_CONV .AND. N_BOUND .EQ. 0) THEN
          IITER = 1
          STOP
          ENDIF
      ENDIF

      IF (I_DSA .EQ. 1 .AND. I_DBOX_SMALL .EQ. 0) THEN
        IF (I_DSA_DV .EQ. 0) THEN
            I_DSA_DV = 1
            PRINT*, 'WRONG I_DSA_DV!'
c            STOP
        ENDIF
        IF (X_DV_DSA .EQ. 0.0) THEN
            X_DV_DSA = 1.0
            PRINT*, 'WRONG X_DV_DSA!'
c            STOP
        ENDIF

        IF (I_FEAS_LAST2 .EQ. 0) THEN
          DSA_SENS_C(I_DSA_DV) = (G_DSA2(1) - G_FEAS(1))/X_DV_DSA

        ELSE IF (I_FEAS_LAST2 .EQ. 1) THEN
          DSA_SENS_C(I_DSA_DV) = (G_DSA2(1) - G_DSA1(1))/X_DV_DSA

        ELSE
        PRINT*, 'WRONG I_FEAS_LAST2 = ', I_FEAS_LAST2
        STOP
        ENDIF

        IF (DSA_SENS_C(I_DSA_DV)*G_FEAS(I_DSA_DV+1) .GT. 0) THEN
            DV_TRUST(I_DSA_DV) = 1
        ELSE
            DV_TRUST(I_DSA_DV) = -1
        ENDIF
        I_DSA_DV = 1
        I_DSA = 0
        X_DV_DSA = 0.0
      ENDIF

      WRITE(112,'(I8, F20.10, F20.10, F20.10, F20.10, F20.10,
     1      F20.10, F20.10, F20.10, F20.10)'),
     2      I_SQP, DV_TRUST(1:NDV)

      WRITE(113,'(I8, F20.10, F20.10, F20.10, F20.10, F20.10,
     1      F20.10, F20.10, F20.10, F20.10)'),
     2      I_SQP, DSA_SENS_C(1:NDV)

      WRITE(116,'(I8,A40)'),I_SQP, '-------------------------'
      WRITE(116,'(A20, F20.10, F20.10, F20.10, F20.10, F20.10,
     1      F20.10, F20.10, F20.10, F20.10)'),'G_SQP_PDD',
     1      G_SQP(MAX(1,I_SQP-1),2:NDV+1)

      WRITE(116,'(A20, F20.10, F20.10, F20.10, F20.10, F20.10,
     1      F20.10, F20.10, F20.10, F20.10)'),'G_SQP_AIJK',
     2      G_SQP_AIJK(MAX(1,I_SQP-2),2:NDV+1)

      X_MAX_CHANGE = 0.
      I_GRD_MAX = 0
      G_GRD_MAX = 0.
      G_GRD = 0.
      I_GRD_NEG = 0
      I_GRD_MAX_RE = 0
      G_GRD_MAX_RE = 0.
      G_GRD_RE = 0.
      I_GRD_MAX_ABS_CHANGE = 0
      G_GRD_MAX_ABS_CHANGE = 0.
      G_GRD_ABS_CHANGE = 0.

      X_UNTR = X

          IF (I_FEASIBLE .EQ. 1 .AND. I_FEASIBLE_CUR .EQ. 1
     1      .AND. I_SQP .GT. 2) THEN
            DO I = 1, NDV
               X_CHANGE = ABS(X_AIJK_LAST2T(I)-X_FEASIBLE(I))
               G_GRD = ABS(G_SQP(I_SQP-1, I+1))
               G_GRD_RE = MAX(ABS(G_SQP(I_SQP-1, I+1)/
     1          G_SQP_AIJK(I_SQP-2,I+1)),
     1          ABS(G_SQP_AIJK(I_SQP-2,I+1)/G_SQP(I_SQP-1, I+1)))

             G_GRD_ABS_CHANGE = ABS(G_SQP(I_SQP-1, I+1)-
     1        G_SQP_AIJK(I_SQP-2,I+1))
               IF (G_GRD_MAX_ABS_CHANGE .LT. G_GRD_ABS_CHANGE) THEN
               G_GRD_MAX_ABS_CHANGE = G_GRD_ABS_CHANGE
               I_GRD_MAX_ABS_CHANGE = I
               ENDIF

               IF (G_SQP(I_SQP-1, I+1)*G_SQP_AIJK(I_SQP-2,I+1)
     1          .LT. 0.0) THEN
               I_GRD_NEG(I) = 1
               ENDIF

               IF (G_GRD_MAX_RE .LT. G_GRD_RE) THEN
                 G_GRD_MAX_RE = G_GRD_RE
                 I_GRD_MAX_RE = I
               ENDIF

               IF (DV_TRUST(I) .LT. 0) X_CHANGE = 1.0* X_CHANGE
               IF (X_CHANGE .GE.X_MAX_CHANGE) THEN
                  X_MAX_CHANGE = X_CHANGE
                  I_MAX_CHANGE = I
               ENDIF
               IF (G_GRD_MAX .LT. G_GRD) THEN
                 G_GRD_MAX = G_GRD
                 I_GRD_MAX = I
               ENDIF
               IF (I_GRD_MAX .EQ. 0) THEN
               PRINT*, G_DSA1(1:10)
               STOP
               ENDIF
             ENDDO
             CTMP1 = 1
             IF (G_GRD_MAX_ABS_CHANGE .GE. 20.0) THEN
             CTMP1 = 0.9
             ELSE IF (G_GRD_MAX_ABS_CHANGE .GE. 5.0) THEN
             CTMP1 = 0.95
             ENDIF
c           DBOX(I_GRD_MAX_ABS_CHANGE) = 0.618*DBOX(I_GRD_MAX_ABS_CHANGE)
          X(I_GRD_MAX_ABS_CHANGE) = X_FEASIBLE(I_GRD_MAX_ABS_CHANGE)+
     1    CTMP1 *(X(I_GRD_MAX_ABS_CHANGE)-
     2    X_FEASIBLE(I_GRD_MAX_ABS_CHANGE))

C          DELTA_X = X - X_FEASIBLE
C          DO II = 1,NDV
C          IF(DELTA_X(II)*OBJ_SQP(I_SQP-1,II+1) .GT. 0.0) THEN
C          X(II) = X_FEASIBLE(II)
C          ENDIF
C          ENDDO

          CALL TRANSLATE_X (X,NDV,X0, XL01, XU01, -1)

          X_AIJK_LAST = X0

          WRITE(111,'(A48, F20.10, F20.10, F20.10, F20.10, F20.10,
     1      F20.10, F20.10, F20.10, F20.10)'),
     2      'TR', X0(1:N)

          ENDIF

          X_MAX_CHANGE = -0.1
          I_C1_NE_MAX_CHANGE1 = 0
          I_C1_PO_MAX_CHANGE1 = 0
          I_C1_NE_MAX_CHANGE2 = 0
          I_C1_PO_MAX_CHANGE2 = 0
          C1_NE_MAX_CHANGE1 = -0.1
          C1_PO_MAX_CHANGE1 = -0.1
          C1_NE_MAX_CHANGE2 = -0.1
          C1_PO_MAX_CHANGE2 = -0.1

          IF (I_FEASIBLE .EQ. 1 .AND. I_FEASIBLE_CUR .EQ. 0 ) THEN
            DELTA_X = X_AIJK_LAST2T - X_FEASIBLE
            C_V = G_SQP(I_SQP-1, 1)
            DO I = 1,NDV
                DDX = DELTA_X(I)
                IF(OBJ_SQP(I_SQP-1,I+1)*DELTA_X(I) .GT. 0) THEN
                      X(I) = X_FEASIBLE(I)+0.1*DDX
                ELSE
                   IF (ABS(DDX) .GT. X_MAX_CHANGE) THEN
                      X_MAX_CHANGE = ABS(DDX)
                      I_MAX_CHANGE = I
                   ENDIF
                ENDIF
            ENDDO
            DO I = 1, NDV
                G_P = G_SQP(I_SQP-1, I+1)
                G_A = G_SQP_AIJK(I_SQP-2, I+1)
                DDX = DELTA_X(I)
                IF(OBJ_SQP(I_SQP-1,I+1)*DELTA_X(I) .GT. 0) THEN
                      X(I) = X_FEASIBLE(I)+0.1*DDX
                ELSE
                   IF (I .NE.I_MAX_CHANGE) THEN
                   IF (G_P*G_A .LT. 0) THEN
                     IF (G_P * DDX .LT. 0) THEN
                        C1_NE_CHANGE1 = ABS(G_P*G_A)
                        IF(C1_NE_MAX_CHANGE1 .LT. C1_NE_CHANGE1) THEN
                        C1_NE_MAX_CHANGE1 = C1_NE_CHANGE1
                        I_C1_NE_MAX_CHANGE1 = I
                        ENDIF
                     ELSE
                        IF (G_P*DDX .GT. C_V/9.0) THEN
                        X(I) = X_FEASIBLE(I)+0.8*DDX
                        DBOX(I) = 0.95*DBOX(I)
                        ENDIF
                        C1_PO_CHANGE1 = ABS(G_P*G_A)
                        IF(C1_PO_MAX_CHANGE1 .LT. C1_PO_CHANGE1) THEN
                        C1_PO_MAX_CHANGE1 = C1_PO_CHANGE1
                        I_C1_PO_MAX_CHANGE1 = I
                        ENDIF
                     ENDIF
                   ELSE
                     IF (G_P * DDX .LT. 0) THEN
                        C1_NE_CHANGE2 = ABS(G_P*G_A)
                        IF(C1_NE_MAX_CHANGE2 .LT. C1_NE_CHANGE2) THEN
                        C1_NE_MAX_CHANGE2 = C1_NE_CHANGE2
                        I_C1_NE_MAX_CHANGE2 = I
                        ENDIF
                     ELSE
                        IF (G_P*DDX .GT. C_V/9.0) THEN
                        X(I) = X_FEASIBLE(I)+0.8*DDX
                        DBOX(I) = 0.95*DBOX(I)
                        ENDIF
                        C1_PO_CHANGE2 = ABS(G_P*G_A)
                        IF(C1_PO_MAX_CHANGE2 .LT. C1_PO_CHANGE2) THEN
                        C1_PO_MAX_CHANGE2 = C1_PO_CHANGE2
                        I_C1_PO_MAX_CHANGE2 = I
                        ENDIF
                     ENDIF
                ENDIF
              ENDIF
             ENDIF
           ENDDO

C   --   IF     I_C1_PO_MAX_CHANGEI > 2 * C_V  THEN *0.618
C   ---  I_MAX_CHANGE BOX*0.618 IF AND ONLY IF > 5 * C_v

            X(I_MAX_CHANGE) =  X_FEASIBLE(I_MAX_CHANGE) +
     1          0.618*DELTA_X(I_MAX_CHANGE)
            G_P = G_SQP(I_SQP-1, I_MAX_CHANGE+1)
            DDX = DELTA_X(I_MAX_CHANGE)
            IF (G_P*DDX .GT. 5 * C_V) THEN
                DBOX(I_MAX_CHANGE)= 0.618*DBOX(I_MAX_CHANGE)
            ELSE IF (G_P*DDX .GT. 2 * C_V) THEN
                DBOX(I_MAX_CHANGE)= 0.8*DBOX(I_MAX_CHANGE)
            ELSE
                DBOX(I_MAX_CHANGE)= 0.9*DBOX(I_MAX_CHANGE)
            ENDIF

            IF (I_C1_NE_MAX_CHANGE1 .NE. 0) THEN
                IF (C1_NE_MAX_CHANGE1 .GE. 2.0*C_V) THEN
                X(I_C1_NE_MAX_CHANGE1) = X_FEASIBLE(I_C1_NE_MAX_CHANGE1)
     1              + 0.618*DELTA_X(I_C1_NE_MAX_CHANGE1)
                DBOX(I_C1_NE_MAX_CHANGE1) = 0.618*
     1                  DBOX(I_C1_NE_MAX_CHANGE1)
                ELSE IF (C1_NE_MAX_CHANGE1 .GE. 2.0*C_V/9.0) THEN
                X(I_C1_NE_MAX_CHANGE1) = X_FEASIBLE(I_C1_NE_MAX_CHANGE1)
     1              + 0.8*DELTA_X(I_C1_NE_MAX_CHANGE1)
                DBOX(I_C1_NE_MAX_CHANGE1) = 0.9*
     1                  DBOX(I_C1_NE_MAX_CHANGE1)
                ELSE
                X(I_C1_NE_MAX_CHANGE1) = X_FEASIBLE(I_C1_NE_MAX_CHANGE1)
     1              + 0.9*DELTA_X(I_C1_NE_MAX_CHANGE1)
                DBOX(I_C1_NE_MAX_CHANGE1) = 0.95*
     1                  DBOX(I_C1_NE_MAX_CHANGE1)
                ENDIF
            ENDIF
            IF (I_C1_PO_MAX_CHANGE1 .NE. 0) THEN
                IF (C1_PO_MAX_CHANGE1 .GE. 1.0*C_V) THEN
                X(I_C1_PO_MAX_CHANGE1) = X_FEASIBLE(I_C1_PO_MAX_CHANGE1)
     1              + 0.618*DELTA_X(I_C1_PO_MAX_CHANGE1)
                DBOX(I_C1_PO_MAX_CHANGE1) = 0.618*
     1                  DBOX(I_C1_PO_MAX_CHANGE1)
                ELSE IF (C1_PO_MAX_CHANGE1 .GE. 1.0*C_V/9.0) THEN
                X(I_C1_PO_MAX_CHANGE1) = X_FEASIBLE(I_C1_PO_MAX_CHANGE1)
     1              + 0.618*DELTA_X(I_C1_PO_MAX_CHANGE1)
                DBOX(I_C1_PO_MAX_CHANGE1) = 0.8*
     1                  DBOX(I_C1_PO_MAX_CHANGE1)
                ELSE
                X(I_C1_PO_MAX_CHANGE1) = X_FEASIBLE(I_C1_PO_MAX_CHANGE1)
     1              + 0.9*DELTA_X(I_C1_PO_MAX_CHANGE1)
                DBOX(I_C1_PO_MAX_CHANGE1) = 0.95*
     1                  DBOX(I_C1_PO_MAX_CHANGE1)
                ENDIF
            ENDIF

            IF (I_C1_NE_MAX_CHANGE2 .NE. 0) THEN
                IF (C1_NE_MAX_CHANGE2 .GE. 2.0*C_V) THEN
                X(I_C1_NE_MAX_CHANGE2) = X_FEASIBLE(I_C1_NE_MAX_CHANGE2)
     1              + 0.618*DELTA_X(I_C1_NE_MAX_CHANGE2)
                DBOX(I_C1_NE_MAX_CHANGE2) = 0.618*
     1                  DBOX(I_C1_NE_MAX_CHANGE2)
                ELSE IF (C1_NE_MAX_CHANGE2 .GE. 2.0*C_V/9.0) THEN
                X(I_C1_NE_MAX_CHANGE2) = X_FEASIBLE(I_C1_NE_MAX_CHANGE2)
     1              + 0.8*DELTA_X(I_C1_NE_MAX_CHANGE2)
                DBOX(I_C1_NE_MAX_CHANGE2) = 0.9*
     1                  DBOX(I_C1_NE_MAX_CHANGE2)
                ELSE
                X(I_C1_NE_MAX_CHANGE2) = X_FEASIBLE(I_C1_NE_MAX_CHANGE2)
     1              + 0.9*DELTA_X(I_C1_NE_MAX_CHANGE2)
                DBOX(I_C1_NE_MAX_CHANGE2) = 0.95*
     1                  DBOX(I_C1_NE_MAX_CHANGE2)
                ENDIF
            ENDIF
            IF (I_C1_PO_MAX_CHANGE2 .NE. 0) THEN
                IF (C1_PO_MAX_CHANGE2 .GE. 1.0*C_V) THEN
                X(I_C1_PO_MAX_CHANGE2) = X_FEASIBLE(I_C1_PO_MAX_CHANGE2)
     1              + 0.618*DELTA_X(I_C1_PO_MAX_CHANGE2)
                DBOX(I_C1_PO_MAX_CHANGE2) = 0.618*
     1                  DBOX(I_C1_PO_MAX_CHANGE2)
                ELSE IF (C1_PO_MAX_CHANGE2 .GE. 1.0*C_V/9.0) THEN
                X(I_C1_PO_MAX_CHANGE2) = X_FEASIBLE(I_C1_PO_MAX_CHANGE2)
     1              + 0.618*DELTA_X(I_C1_PO_MAX_CHANGE2)
                DBOX(I_C1_PO_MAX_CHANGE2) = 0.8*
     1                  DBOX(I_C1_PO_MAX_CHANGE2)
                ELSE
                X(I_C1_PO_MAX_CHANGE2) = X_FEASIBLE(I_C1_PO_MAX_CHANGE2)
     1              + 0.9*DELTA_X(I_C1_PO_MAX_CHANGE2)
                DBOX(I_C1_PO_MAX_CHANGE2) = 0.95*
     1                  DBOX(I_C1_PO_MAX_CHANGE2)
                ENDIF
            ENDIF
            DELTA_X2 = X - X_FEASIBLE
            DO II = 1,NDV
            IF (X(II) .EQ. X_N(II)) THEN
                X(II) = X_FEASIBLE(II) + 0.9*DELTA_X(II)
            ENDIF

            IF(DELTA_X2(II)*OBJ_SQP(I_SQP-1,II+1) .GT. 0.0) THEN
            X(II) = X_FEASIBLE(II)
            ENDIF
            ENDDO
            DO I = 1, NDV
            IF(ABS(X_N(I)-X_FEASIBLE(I)) .LT. ABS(X(I)-X_FEASIBLE(I))
     1           ) THEN
            X(I) = X_N(I)
            ENDIF
            ENDDO

            WRITE(115,'(I8,A40)'), I_SQP, '---------------------------'
C            WRITE(115,'(A20, I8, I8, I8, I8, I8,
C     1      I8, I8, I8, I8)'), 'I_GRD_NEG',
C     2      I_GRD_NEG(1:NDV)
            WRITE(115,'(A20, F20.10, F20.10, F20.10, F20.10, F20.10,
     1      F20.10, F20.10, F20.10, F20.10)'), 'X_N',
     2      X_N(1:NDV)
            WRITE(115,'(A20, I8, A30,F20.10)'), 'I_C1_NE_MAX_CHANGE1=',
     1       I_C1_NE_MAX_CHANGE1, 'C1_NE_MAX_CHANGE1 = ',
     1       C1_NE_MAX_CHANGE1
            WRITE(115,'(A20, I8, A30,F20.10)'), 'I_C1_PO_MAX_CHANGE1=',
     1       I_C1_PO_MAX_CHANGE1, 'C1_PO_MAX_CHANGE1 = ',
     1       C1_PO_MAX_CHANGE1
            WRITE(115,'(A20, I8, A30,F20.10)'), 'I_C1_NE_MAX_CHANGE2=',
     1       I_C1_NE_MAX_CHANGE2, 'C1_NE_MAX_CHANGE2 = ',
     1       C1_NE_MAX_CHANGE2
            WRITE(115,'(A20, I8, A30,F20.10)'), 'I_C1_PO_MAX_CHANGE2=',
     1       I_C1_PO_MAX_CHANGE2, 'C1_PO_MAX_CHANGE2 = ',
     1       C1_PO_MAX_CHANGE2

            DBOX_MAX = MAXVAL(DBOX)
            N_VIOLANCE =  N_VIOLANCE + 1
c            RPRM(11) = RMVLMZ*0.618**N_VIOLANCE
c            IF (DBOX(I_MAX_CHANGE) .LT. 2.0 .AND. I_DBOX_SMALL .LE. 4)
            IF (DBOX(I_MAX_CHANGE) .LT. 2.0 .AND. DBOX_MAX .GE.
              IF (DBOX(I_MAX_CHANGE) .LT. 2.0 .AND. DBOX_MAX .GE.
     1      DBOX(I_MAX_CHANGE)*10) THEN
C            DBOX = 4.0
C            DBOX(I_MAX_CHANGE) = 2.0
            TMP3 = DBOX(I_MAX_CHANGE)
            DO I = 1, NDV
            IF (DBOX(I) .GT. 3.0*TMP3) THEN
            DBOX(I) = 0.8*DBOX(I)
            ENDIF
            ENDDO
            DBOX(I_MAX_CHANGE) = TMP3

            ENDIF

            DO I = 1,NDV
                IF (DBOX(I) .LT. 0.2) THEN
                    DBOX(I) = 0.2
                ENDIF
                IF (DBOX(I) .LT. X_MAX_CHANGE) THEN
                    IF (X_MAX_CHANGE .LT. 5.0 .AND.
     1               X_MAX_CHANGE .GE. 0.0) THEN
                    DBOX(I) = DBOX(I_MAX_CHANGE)
                    ENDIF
                ENDIF
            ENDDO

            I_FEAS_LAST2 = 0
            X_DV_DSA = -0.618*TMP

            IF (X_DV_DSA .EQ. 0.0) THEN
            PRINT*, 'XDV_DSA = 0.0'
            ENDIF

c            I_FEASIBLE_LAST = 0

            I_DSA_DV = I_MAX_CHANGE

          CALL TRANSLATE_X (X,NDV,X0, XL01, XU01, -1)

          X_AIJK_LAST = X0

          WRITE(111,'(A48, F20.10, F20.10, F20.10, F20.10, F20.10,
     1      F20.10, F20.10, F20.10, F20.10)'),
     2      'TR', X0(1:N)
          X_CENTRAL = X
        ELSE
          X_CENTRAL = X
        ENDIF
C
C  -----  IHISTORY_TR(J,191:199) IF COMPONENT INCREASED SET TO 1,
C  -----  IHISTORY_TR(J,1:NDV) SAVE CORDINATE OF X
C  ------ IHISTORY_TR(2000,200) SAVE TOTAL NUMBER OF INCREASING TIMES

      DO I = 1,NDV
        I_FLAG_INCRD = 0
C        IF (I_FEASIBLE_CUR .EQ. 1) THEN
        IF ((X_UNTR(I) .EQ. XL(I) .AND. XL(I) - XL0(I) .NE. 0.0)
     1    .OR. (X_UNTR(I) .EQ. XU(I) .AND. XU(I) - XU0(I) .NE. 0.0))THEN
        I_TMP = IHISTORY_TR(2000,200)
        DO J = 1, I_TMP
        RTMP = 0.0
        DO K = 1, NDV
        RTMP = RTMP + (X_UNTR(K)-IHISTORY_TR(J,K))**2.0D0
        ENDDO
        IF(DSQRT(RTMP)/9.0 .LE. 0.4 .AND. IHISTORY_TR(J,190+I) .EQ. 1)
     1   THEN
        IF (ABS(X_UNTR(I)-IHISTORY_TR(J,I)) .LT. 2.0*DBOX(I)) THEN
        I_FLAG_INCRD = 1
        EXIT
        ENDIF

        ENDIF
        ENDDO

        IF (DBOX(I) .LT. 2.0 .AND. I_FLAG_INCRD .EQ. 0 .AND.
     1  XL(I) - XL0(I) .NE. 0.0 .AND. XU(I) - XU0(I) .NE. 0.0
     2   ) THEN
        DBOX(I) =1/0.618*DBOX(I)
        IHISTORY_TR(I_TMP+1,1:NDV) = X_UNTR(1:NDV)
        IHISTORY_TR(I_TMP+1,190+I) = 1
        IHISTORY_TR(2000,200) = IHISTORY_TR(2000,200) + 1
C        I_INCR_DBOX(I) = 1

        ENDIF
C        ENDIF
        DBOX_I = DBOX(I)
        IF (DBOX_I .LT. 2.0) THEN
        TMP1 = ABS(X_CENTRAL(I)-XL(I))
        TMP2 = ABS(X_CENTRAL(I)-XU(I))
        IF (TMP1 .GT. TMP2) THEN
            XL(I) = XL(I)+0.5*DBOX(I)
            XU(I) = XU(I)+0.5*DBOX(I)
        ELSE IF (TMP1 .LT. TMP2)THEN
            XL(I) = XL(I)-0.5*DBOX(I)
            XU(I) = XU(I)-0.5*DBOX(I)
        ENDIF
        ELSE
            XL(I) = X_CENTRAL(I)-0.5*DBOX(I)
            XU(I) = X_CENTRAL(I)+0.5*DBOX(I)
        ENDIF
        ELSE
        XL(I) = X_CENTRAL(I) - 0.5*DBOX(I)
        XU(I) = X_CENTRAL(I) + 0.5*DBOX(I)
        ENDIF
        IF (XL(I) .LT. XL0(I)) XL(I) = XL0(I)
        IF (XU(I) .GT. XU0(I)) XU(I) = XU0(I)
        IF (XL(I) .GE. XU0(I)) XL(I) = XU0(I)  - 2.0
        IF (XU(I) .LE. XL0(I)) XU(I) = XL0(I)  + 2.0

      ENDDO

      WRITE(110,'(I8, F20.10, F20.10, F20.10, F20.10, F20.10,
     1      F20.10, F20.10, F20.10, F20.10)'),
     2      I_SQP, DBOX(1:N)

      ENDIF

      IF (IBOX .LT. 0) THEN
      DO I = 1,NDV
C         TMP1 = ABS(X(I)-XL(I))
C         TMP2 = ABS(X(I)-XU(I))
C         IF (TMP1 .GT. TMP2) THEN
C             XL(I) = XL(I)+0.5*DBOX(I)
C             XU(I) = XU(I)+0.5*DBOX(I)
C         ELSE IF (TMP1 .LT. TMP2)THEN
C             XL(I) = XL(I)-0.5*DBOX(I)
C             XU(I) = XU(I)-0.5*DBOX(I)
C         ENDIF
C         IF (I .EQ. I_MAX_CHANGE) THEN
C             XL(I) = X(I)-0.5*DBOX(I)
C             XU(I) = X(I)+0.5*DBOX(I)
C         ENDIF
C
        XL(I) = X(I) - 0.5*DBOX(I)
        XU(I) = X(I) + 0.5*DBOX(I)
        IF (XL(I) .LT. XL0(I)) XL(I) = XL0(I)
        IF (XU(I) .GT. XU0(I)) XU(I) = XU0(I)
        IF (XL(I) .GE. XU0(I)) XL(I) = XU0(I)  - 2.0
        IF (XU(I) .LE. XL0(I)) XU(I) = XL0(I)  + 2.0

      ENDDO
C       N_VIOLANCE =  N_VIOLANCE + 1
C       RPRM(11) = RMVLMZ*0.618**N_VIOLANCE
      ENDIF

      IF (IBOX .EQ. 0) THEN
        XL = XL0
        XU = XU0
C         N_VIOLANCE =  N_VIOLANCE + 1
C         RPRM(11) = RMVLMZ*0.618**N_VIOLANCE
      ENDIF

      CALL TRANSLATE_X (XL,NDV,XL02, XL01, XU01, -1)
      CALL TRANSLATE_X (XU,NDV,XU02, XL01, XU01, -1)
      WRITE(114,'(I8, F20.10, F20.10, F20.10, F20.10, F20.10,
     1      F20.10, F20.10, F20.10, F20.10)'),
     2      I_SQP, XL02(1:N)
      WRITE(114,'(A8, F20.10, F20.10, F20.10, F20.10, F20.10,
     1      F20.10, F20.10, F20.10, F20.10)'),
     2      'XU', XU02(1:N)

100   CALL DOT (INFO,METHOD,IPRINT,NDV,NCON,X,XL,XU,
     1 OBJ1,MINMAX,G1,RPRM,IPRM,WK,NRWK,IWK,NRIWK)
C     FINISHED?
      CALL TRANSLATE_X (X,NDV,X0, XL01, XU01, -1)

      PRINT *, '*****************************************************'
      PRINT *, '*****************************************************'
      PRINT *, '*****************************************************'
      PRINT *, 'X=',X0(1),X0(2),X0(3),X0(4),X0(5),X0(6),X0(7),X0(8),
     1           X0(9)
      PRINT *, '*****************************************************'
      PRINT *, '*****************************************************'
      PRINT *, '*****************************************************'
      CALL SLEEP(1)
C       PAUSE 1

      IF (IITER .GT. 190) THEN
        IITER_TOTAL = IITER_TOTAL + 1
        WRITE(109,*), 'NUMBER OF OBJECTIVE FUNCTION CALLS
     1    N_F = ', N_F
C         PRINT *, 'N_GF = ', N_GF
C         PRINT *, 'N_GC = ', N_GC
        WRITE(109,*), 'TOTAL NUMBER OF ROUGH INTERATION IITER = ', IITER
        WRITE(109,*), 'NUMBER OF OBJECTIVE FUNCTION CALLS FOR EVALUATION
     1   IFCALL = ', IFCALL
        WRITE(109,*), 'NUMBER OF OBJECTIVE FUNCTION CALLS FOR GRADIENT
     1      IGFCALL = ', IGFCALL
        WRITE(109,*), 'NUMBER OF CONSTRAINT FUNCTION CALLS FOR GRADIENT
     1      IGCCALL = ', IGCCALL
        WRITE(109,*), 'NUMBER OF FUNCTION F EVALUATION N_EF = ', N_EF
        WRITE(109,*), 'NUMBER OF FUNCTION G_I EVALUATION N_EC = ', N_EC
        WRITE(109,*), 'N_EF1 = ', N_EF1
        WRITE(109,*), 'N_EC1 = ', N_EC1
        WRITE(109,*), 'EVALUATION FOR BIVARIATE OF F: N_EF2 = ', N_EF2 +
     1  N_EF1
        WRITE(109,*), 'EVALUATION FOR BIVARIATE OF C: N_EC2 = ', N_EC2 +
     1  N_EC1
        WRITE(109,*),'NUMBER OF PREPROCESS BY HM: N_PREFEA = ', N_PREFEA
        WRITE(109,*),'NUMBER OF JOB BY ABAQUS: N_FEA = ', N_FEA

        WRITE(109,*) '---------------ITERATION PROCESS----------------'
        WRITE(109,*) 'X(1)      X(2)        X(3)      X(4)        X(5)
     1      X(6)      X(7)        X(8)      X(9)
     2   OBJ    CONS1 '

        DO I = 1,IITER-1
            I1 = I+1
C             PRINT *, I, X_IT(I,1), X_IT(I,2),OBJ_IT(I1,1),OBJ_IT(I1,2)
C     1        ,OBJ_IT(I1,3), CONS_IT(I1,1), CONS_IT(I1,2), CONS_IT(I1,3)
            WRITE(109,'(I8, F20.10, F20.10, F20.10, F20.10, F20.10,
     1      F20.10, F20.10, F20.10, F20.10, F20.10, F20.10)'),
     1    I, X_IT(I,1), X_IT(I,2),X_IT(I,3), X_IT(I,4),X_IT(I,5),
     1      X_IT(I,6), X_IT(I,7),X_IT(I,8), X_IT(I,9),
     1  OBJ_IT(I1,1), CONS_IT(I1,1)

        ENDDO
C     -----------  WRITE REDUCED ITERATION PROCESS --------
        WRITE(109,*) '----------REDUCED ITERATION PROCESS-------------'
        WRITE(109,*) 'X(1)      X(2)        X(3)      X(4)        X(5)
     1      X(6)      X(7)        X(8)      X(9)
     2   OBJ    CONS1 '

        DO I = 1,IITER1
            I1 = I
            PRINT *,
     1    I, X_IT(I,1), X_IT(I,2),X_IT(I,3),X_IT(I,4),X_IT(I,5),
     1      X_IT(I,6), X_IT(I,7),X_IT(I,8), X_IT(I,9)
            WRITE(109,'(I8, F20.10, F20.10, F20.10, F20.10, F20.10,
     1      F20.10, F20.10, F20.10, F20.10, F20.10, F20.10)'),
     1    I, X_IT(I,1), X_IT(I,2),X_IT(I,3),X_IT(I,4),X_IT(I,5),
     1      X_IT(I,6), X_IT(I,7),X_IT(I,8), X_IT(I,9),
     1  OBJ_IT1(I1,1), CONS_IT1(I1,1)

        ENDDO
       XLAST(:) = X_IT(IITER-1,:)
       IITER = 1
       IITER1 = 1

       ENDIF

      IF (IITER .GE. 1) THEN
      X_IT(IITER,:) = X0
      OBJ_IT(IITER,:) = OBJ
      CONS_IT(IITER,:) = G
      ENDIF

      IF (IITER .GE. 2) THEN
      XLAST(:) = X_IT(IITER-1,:)
      ENDIF

      IITER = IITER + 1
      IF(INFO .EQ. 0) THEN
        IITER_TOTAL = IITER_TOTAL +1
        PRINT *, 'NUMBER OF OBJECTIVE FUNCTION CALLS
     1    N_F = ', N_F
C         PRINT *, 'N_GF = ', N_GF
C         PRINT *, 'N_GC = ', N_GC
        PRINT *, 'TOTAL NUMBER OF ROUGH INTERATION IITER = ', IITER
        PRINT *, 'NUMBER OF OBJECTIVE FUNCTION CALLS FOR EVALUATION
     1   IFCALL = ', IFCALL
        PRINT *, 'NUMBER OF OBJECTIVE FUNCTION CALLS FOR GRADIENT
     1      IGFCALL = ', IGFCALL
        PRINT *, 'NUMBER OF CONSTRAINT FUNCTION CALLS FOR GRADIENT
     1      IGCCALL = ', IGCCALL
        PRINT *, 'NUMBER OF FUNCTION F EVALUATION N_EF = ', N_EF
        PRINT *, 'NUMBER OF FUNCTION G_I EVALUATION N_EC = ', N_EC
        PRINT *, 'EVALUATION FOR UNIVARIATE OF F: N_EF1 = ', N_EF1
        PRINT *, 'EVALUATION FOR UNIVARIATE OF C: N_EC1 = ', N_EC1
        PRINT *, 'EVALUATION FOR BIVARIATE OF F: N_EF2 = ', N_EF2+N_EF1
        PRINT *, 'EVALUATION FOR BIVARIATE OF C: N_EC2 = ', N_EC2+N_EC1
        PRINT *, 'NUMBER OF PREPROCESS BY HM: N_PREFEA = ', N_PREFEA
        PRINT *, 'NUMBER OF JOB BY ABAQUS: N_FEA = ', N_FEA
        PRINT *, '---------------ITERATION PROCESS------------------'
        PRINT *, 'K      X(1)      X(2)        OBJ     OBJ_G1     OBJ_G2
     1      CONS    CONS_G1     CONS_G2 '

        WRITE(109,*), 'NUMBER OF OBJECTIVE FUNCTION CALLS
     1    N_F = ', N_F
C         PRINT *, 'N_GF = ', N_GF
C         PRINT *, 'N_GC = ', N_GC
        WRITE(109,*), 'TOTAL NUMBER OF ROUGH INTERATION IITER = ', IITER
        WRITE(109,*), 'NUMBER OF OBJECTIVE FUNCTION CALLS FOR EVALUATION
     1   IFCALL = ', IFCALL
        WRITE(109,*), 'NUMBER OF OBJECTIVE FUNCTION CALLS FOR GRADIENT
     1      IGFCALL = ', IGFCALL
        WRITE(109,*), 'NUMBER OF CONSTRAINT FUNCTION CALLS FOR GRADIENT
     1      IGCCALL = ', IGCCALL
        WRITE(109,*), 'NUMBER OF FUNCTION F EVALUATION N_EF = ', N_EF
        WRITE(109,*), 'NUMBER OF FUNCTION G_I EVALUATION N_EC = ', N_EC
        WRITE(109,*), 'N_EF1 = ', N_EF1
        WRITE(109,*), 'N_EC1 = ', N_EC1
        WRITE(109,*), 'EVALUATION FOR BIVARIATE OF F: N_EF2 = ', N_EF2 +
     1  N_EF1
        WRITE(109,*), 'EVALUATION FOR BIVARIATE OF C: N_EC2 = ', N_EC2 +
     1  N_EC1
        WRITE(109,*),'NUMBER OF PREPROCESS BY HM: N_PREFEA = ', N_PREFEA
        WRITE(109,*),'NUMBER OF JOB BY ABAQUS: N_FEA = ', N_FEA

        WRITE(109,*) '---------------ITERATION PROCESS----------------'
        WRITE(109,*) 'X(1)      X(2)        X(3)      X(4)        X(5)
     1      X(6)      X(7)        X(8)      X(9)
     2   OBJ    CONS1 '

        DO I = 1,IITER-1
            I1 = I+1
C             PRINT *, I, X_IT(I,1), X_IT(I,2),OBJ_IT(I1,1),OBJ_IT(I1,2)
C     1        ,OBJ_IT(I1,3), CONS_IT(I1,1), CONS_IT(I1,2), CONS_IT(I1,3)
            WRITE(109,'(I8, F20.10, F20.10, F20.10, F20.10, F20.10,
     1      F20.10, F20.10, F20.10, F20.10, F20.10, F20.10)'),
     1    I, X_IT(I,1), X_IT(I,2),X_IT(I,3), X_IT(I,4),X_IT(I,5),
     1      X_IT(I,6), X_IT(I,7),X_IT(I,8), X_IT(I,9),
     1  OBJ_IT(I1,1), CONS_IT(I1,1)

        ENDDO
C     -----------  WRITE REDUCED ITERATION PROCESS --------
        WRITE(109,*) '----------REDUCED ITERATION PROCESS-------------'
        WRITE(109,*) 'X(1)      X(2)        X(3)      X(4)        X(5)
     1      X(6)      X(7)        X(8)      X(9)
     2   OBJ    CONS1 '

        DO I = 1,IITER1
            I1 = I
            PRINT *,
     1    I, X_IT1(I,1), X_IT1(I,2),X_IT1(I,3), X_IT1(I,4),
     2      X_IT1(I,5),
     1      X_IT1(I,6), X_IT1(I,7),X_IT1(I,8), X_IT1(I,9)
            WRITE(109,'(I8, F20.10, F20.10, F20.10, F20.10, F20.10,
     1      F20.10, F20.10, F20.10, F20.10, F20.10, F20.10)'),
     1    I, X_IT1(I,1), X_IT1(I,2),X_IT1(I,3), X_IT1(I,4),
     2      X_IT1(I,5),
     1      X_IT1(I,6), X_IT1(I,7),X_IT1(I,8), X_IT1(I,9),
     1  OBJ_IT1(I1,1), CONS_IT1(I1,1)

        ENDDO

C       --  PRINT EXACT SOLUTION
C        WRITE(109,*) '----------EXACT OF OBJ AND CONS--------------'
C        DO I = 1,IITER-1
C            WRITE(109,'(I8, F20.10, F20.10, F20.10, F20.10, F20.10,
C     1        F20.10, F20.10, F20.10)')
C     1    I, X_IT(I,1), X_IT(I,2),OBJ_IT_EXC(I,1),OBJ_IT_EXC(I,2)
C     2        ,OBJ_IT_EXC(I,3), CONS_IT_EXC(I,1)
C     3       , CONS_IT_EXC(I,2), CONS_IT_EXC(I,3)
C
C        ENDDO

        RTMP = 0.0
        DO I = 1, 9
        RTMP = RTMP + (X_FEASIBLE(I)-X_FEASIBLE_LAST(I))**2.0D0
        ENDDO
        DX = DSQRT(RTMP)

        I_BOUND_LOCAL = 0
        DO I = 1, NDV
            RTMP = ABS(X_FEASIBLE(I)-X_FEASIBLE_LAST(I))
            I_TMP = 0
            IF (ABS(X_FEASIBLE(I)-XL(I)) .LE. 0.001) THEN
                IF (XL(I) - XL0(I) .NE. 0.0) THEN
                I_TMP = 1
                ENDIF
            ELSE IF(ABS(X_FEASIBLE(I)-XU(I)) .LE. 0.001) THEN
                IF (XU(I) - XU0(I) .NE. 0.0) THEN
                I_TMP = 1
                ENDIF
            ENDIF
            IF (DX .LT. EPS_CONV .AND. I_TMP .EQ. 1 ) THEN
                I_BOUND_LOCAL = I_BOUND_LOCAL +1
            ENDIF
        ENDDO
        WRITE(117,'(I8,A40)'),I_SQP,'---------- END1   -------'
          WRITE(117,'(A20, F20.10, F20.10, F20.10, F20.10, F20.10,
     1     F20.10, F20.10, F20.10,F20.10)'),
     2      'X', X(1:NDV)
          WRITE(117,'(A20, F20.10, F20.10, F20.10, F20.10, F20.10,
     1     F20.10, F20.10, F20.10,F20.10)'),
     2      'XL', XL(1:NDV)
          WRITE(117,'(A20, F20.10, F20.10, F20.10, F20.10, F20.10,
     1     F20.10, F20.10, F20.10,F20.10)'),
     2      'XU', XU(1:NDV)
          WRITE(117,'(A20, F20.10, F20.10, F20.10, F20.10, F20.10,
     1     F20.10, F20.10, F20.10,F20.10)'),
     2      'XL0', XL0(1:NDV)
          WRITE(117,'(A20, F20.10, F20.10, F20.10, F20.10, F20.10,
     1     F20.10, F20.10, F20.10,F20.10)'),
     2      'XU0', XU0(1:NDV)

        OBJ_AIJK_LAST2 = OBJ_AIJK_LAST
        OBJ_AIJK_LAST = OBJ1
        X_AIJK_LAST2 = X_AIJK_LAST
        X_AIJK_LAST = X0

C     BEEP
        IF (DX .LE. EPS_CONV .AND. I_BOUND_LOCAL .EQ. 0) THEN
          WRITE(111,'(I8, F20.10, F20.10)'),
     2      I_SQP, OBJ_SQP(I_SQP,1), G_SQP(I_SQP,1)
          WRITE(111,'(I8, F20.10, F20.10, F20.10, F20.10, F20.10,
     1      F20.10, F20.10, F20.10, F20.10, F20.10, F20.10)'),
     2      I_SQP+1, OBJ1, G1(1),X0(1:NDV)

        CLOSE(108)
        CLOSE(109)
        CLOSE(110)
        CLOSE(111)
        CLOSE(112)
        CLOSE(113)
        CLOSE(114)
        CLOSE(115)
        CLOSE(116)
        CLOSE(117)
        PRINT *, CHAR(7)
        PRINT *, CHAR(7)
        PRINT *, CHAR(7)
        STOP
        ELSE
        G_AIJK_SQP = G
        G_SQP_AIJK(I_SQP,:) = G
        GO TO 99
        ENDIF
CC     EVALUATE OBJECTIVE AND CONSTRAINT.
      ELSE IF (INFO .EQ. 1) THEN
          IF (X0(1) .NE. XLAST(1) .OR. X0(2) .NE. XLAST(2) .OR.
     1         X0(3) .NE. XLAST(3) .OR. X0(4) .NE. XLAST(4) .OR.
     2         X0(5) .NE. XLAST(5) .OR. X0(6) .NE. XLAST(6) .OR.
     2         X0(7) .NE. XLAST(7) .OR. X0(8) .NE. XLAST(8) .OR.
     2         X0(9) .NE. XLAST(9)  ) THEN
              IF (IITER .EQ. 1) THEN
              CALL EVAL(OBJ,X,G,0,0)
              OBJ_SQP(I_SQP,:) = OBJ
              G_SQP(I_SQP,:) = G
              G_PDD_SQP = G
              IF (I_FEASIBLE_LAST .EQ. 1 .AND.I_FEASIBLE_CUR .EQ. 0)THEN
                G_DSA2 = G
                I_DSA = 1
              ENDIF
              IF (G(1) .LE. 0.0) THEN
              I_FEASIBLE =  1
              I_FEASIBLE_LAST = I_FEASIBLE_CUR
              I_FEASIBLE_CUR = 1
              X_FEASIBLE_LAST = X_FEASIBLE
              X_FEASIBLE = X
              F_FEAS = OBJ
              G_FEAS = G
              SF_SENS_C_FEAS(1:9) = G(2:10)
              ELSE
c              I_FEASIBLE =  0
              I_FEASIBLE_LAST = I_FEASIBLE_CUR
              I_FEASIBLE_CUR = 0
              ENDIF
              IF (I_FEASIBLE_LAST .EQ. 1 .AND.I_FEASIBLE_CUR .EQ. 0)THEN
                G_DSA1 = G
              ENDIF
              ELSEIF (IITER .GT. 1) THEN
              CALL EVAL(OBJ,X,G,0,N_FLAG_AIJK)
              IF (N_FLAG_AIJK .EQ. 0) THEN
              OBJ_SQP(I_SQP,:) = OBJ
              G_SQP(I_SQP,:) = G
              IF (I_FEASIBLE_LAST .EQ. 1 .AND.I_FEASIBLE_CUR .EQ. 0)THEN
                G_DSA2 = G
                I_DSA = 1
              ENDIF
              IF (G(1) .LE. 0) THEN
              I_FEASIBLE =  1
              I_FEASIBLE_LAST = I_FEASIBLE_CUR
              I_FEASIBLE_CUR = 1
              X_FEASIBLE_LAST = X_FEASIBLE
              X_FEASIBLE = X
              F_FEAS = OBJ
              G_FEAS = G
              SF_SENS_C_FEAS(1:9) = G(2:10)
              ELSE
c              I_FEASIBLE =  0
              I_FEASIBLE_LAST = I_FEASIBLE_CUR
              I_FEASIBLE_CUR = 0
              ENDIF
              IF (I_FEASIBLE_LAST .EQ. 1 .AND.I_FEASIBLE_CUR .EQ. 0)THEN
                G_DSA1 = G
              ENDIF
              ENDIF
              ENDIF

              IITER1  = IITER1 + 1
              IF (IITER1 .GE. 1) THEN
              X_IT1(IITER1,:) = X0
              OBJ_IT1(IITER1,:) = OBJ
              CONS_IT1(IITER1,:) = G
C        EXCTACT SOLUTION
              XT1 = X_IT1(IITER1,:)
c              CALL EVAL_EXC(OBJ_EXC1,XT1,G_EXC1)
              OBJ_IT_EXC1(IITER1,:) = OBJ_EXC1
              CONS_IT_EXC1(IITER1,:) = G_EXC1
              ENDIF

          ENDIF
          OBJ1 = OBJ(1)
          G1(1) = G(1)

          IFCALL = IFCALL +1

C     EVALUATE GRADIENT
      ELSE IF (INFO .EQ. 2) THEN

          IF (X0(1) .NE. XLAST(1) .OR. X0(2) .NE. XLAST(2) .OR.
     1         X0(3) .NE. XLAST(3) .OR. X0(4) .NE. XLAST(4) .OR.
     2         X0(5) .NE. XLAST(5) .OR. X0(6) .NE. XLAST(6) .OR.
     2         X0(7) .NE. XLAST(7) .OR. X0(8) .NE. XLAST(8) .OR.
     2         X0(9) .NE. XLAST(9)  ) THEN
               IITER1  = IITER1 + 1
              IF (IITER .EQ. 1) THEN
              CALL EVAL(OBJ,X,G,0,0)
              OBJ_SQP(I_SQP,:) = OBJ
              G_SQP(I_SQP,:) = G
              G_PDD_SQP = G
              IF (I_FEASIBLE_LAST .EQ. 1 .AND.I_FEASIBLE_CUR .EQ. 0)THEN
                G_DSA2 = G
                I_DSA = 1
              ENDIF
              IF (G(1) .LE. 0) THEN
              I_FEASIBLE =  1
              I_FEASIBLE_LAST = I_FEASIBLE_CUR
              I_FEASIBLE_CUR = 1
              X_FEASIBLE_LAST = X_FEASIBLE
              X_FEASIBLE = X
              F_FEAS = OBJ
              G_FEAS = G
              SF_SENS_C_FEAS(1:9) = G(2:10)
              ELSE
c              I_FEASIBLE =  0
              I_FEASIBLE_LAST = I_FEASIBLE_CUR
              I_FEASIBLE_CUR = 0
              ENDIF
              IF (I_FEASIBLE_LAST .EQ. 1 .AND.I_FEASIBLE_CUR .EQ. 0)THEN
                G_DSA1 = G
              ENDIF

              ELSEIF (IITER .GT. 1) THEN
              CALL EVAL(OBJ,X,G,0,N_FLAG_AIJK)
              IF (N_FLAG_AIJK .EQ. 0) THEN
              OBJ_SQP(I_SQP,:) = OBJ
              G_SQP(I_SQP,:) = G
              IF (I_FEASIBLE_LAST .EQ. 1 .AND.I_FEASIBLE_CUR .EQ. 0)THEN
                G_DSA2 = G
                I_DSA = 1
              ENDIF
              IF (G(1) .LE. 0) THEN
              I_FEASIBLE =  1
              I_FEASIBLE_LAST = I_FEASIBLE_CUR
              I_FEASIBLE_CUR = 1
              X_FEASIBLE_LAST = X_FEASIBLE
              X_FEASIBLE = X
              F_FEAS = OBJ
              G_FEAS = G
              SF_SENS_C_FEAS(1:9) = G(2:10)
              ELSE
c              I_FEASIBLE =  0
              I_FEASIBLE_LAST = I_FEASIBLE_CUR
              I_FEASIBLE_CUR = 0
              ENDIF
              IF (I_FEASIBLE_LAST .EQ. 1 .AND.I_FEASIBLE_CUR .EQ. 0)THEN
                G_DSA1 = G
              ENDIF
              ENDIF
              ENDIF

              IF (IITER1 .GE. 1) THEN
              X_IT1(IITER1,:) = X0
              OBJ_IT1(IITER1,:) = OBJ
              CONS_IT1(IITER1,:) = G
              ENDIF

         ENDIF

        IGFCALL = IGFCALL +1
C        CALL EVAL (OBJ, X, G, 0, 1)
        DO I = 1,NDV
            WK(I) = OBJ(I+1)
        ENDDO

        NGT = IPRM(20)
C     CONSTRAINT GRADIENTS

        IF (NGT .NE. 0) THEN
         IGCCALL = IGCCALL +1
        DO K = 1, NGT
            J=IWK(K)
            AA(K,1) = G((J-1)*10+2)
            AA(K,2) = G((J-1)*10+3)
            AA(K,3) = G((J-1)*10+4)
            AA(K,4) = G((J-1)*10+5)
            AA(K,5) = G((J-1)*10+6)
            AA(K,6) = G((J-1)*10+7)
            AA(K,7) = G((J-1)*10+8)
            AA(K,8) = G((J-1)*10+9)
            AA(K,9) = G((J-1)*10+10)

        ENDDO
        DO K = 1, NGT
        DO I = 1, NDV
            WK(I+(K)*NDV)=AA(K,I)
        ENDDO
        ENDDO
        ENDIF

      ELSE
        PRINT *, 'UNEXPECTED INFO NUMBER, INFO = ', INFO
        STOP
      ENDIF

C     GO CONTINUE WITH OPTIMIZATION.
      GO TO 100
      END

C      SUBROUTINE EVAL (OBJ,X,G,IMCS,IOBJ_CONS)
       SUBROUTINE EVAL (OBJ,X,G,IMCS,IFLAG_AIJK)
CC    SUBROUTINE TO EVALUATE THE OBJECTIVE FUNCTION AND CONSTRAINTS
C     FOR THE BOX DESIGN PROBLEM.
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
        DOUBLE PRECISION X(200),G(200),MOMSEN_OUT(10,2),XMU_IN(200),
     1   OBJ(200), XSIG_IN(200),X0(200),XL01(200),XU01(200)

        REAL*8 XMU(20),SIG(20),NMU(20),NSIG(20)
        COMMON /SR02/ XMU,SIG, NMU, NSIG, NYQ, N
        INTEGER NBAS(20), NGAUSS(20),NGAUSS_SM(20),NBAS_SM(20), IFLAG_Y,
     1       I_EVAL

        COMMON /XR01/ N_F, N_GF, N_GC, N_EF,N_EC, N_EF1, N_EC1,
     1     N_EF2, N_EC2
         COMMON /XR04/ NBAS_R, NGAUSS_R, NVAR

         COMMON /XR09/ I_EVAL
         CHARACTER(80) FN
       COMMON /XR10/ FN
         COMMON /XR11/ XL01, XU01
         COMMON /XR20/ C1_SCAL, I_RECIPROCAL
      INTEGER IITER, IITER1, N_PREFEA, N_FEA, IITER_TOTAL, I_SQP
      COMMON /XR05/ IITER, IITER1, N_PREFEA, N_FEA, IITER_TOTAL, I_SQP

        IF (I_EVAL .EQ. 0) THEN
        OPEN(301,FILE=trim(FN)//'-OBJECT_CONS_SENS.TXT',STATUS='NEW')
        OPEN(302,FILE=trim(FN)//'-F_G1_MEAN_SD.TXT',STATUS='NEW')
        WRITE(301,*) 'I_EVAL       OBJ/CONS      O1/C1        O2/C2
     1          O3/C3      O4/C4       O5/C5      O6/C6        O7/C7
     2      O8/C8        O9/C9 '
        WRITE(302,*) 'I_EVAL       OBJ/CONS      MEAN        SD'
        ELSE
        OPEN(301,FILE=trim(FN)//'-OBJECT_CONS_SENS.TXT', STATUS='OLD',
     1   ACCESS='APPEND')
        OPEN(302,FILE=trim(FN)//'-F_G1_MEAN_SD.TXT',STATUS='OLD',
     1   ACCESS='APPEND')
        ENDIF
         CALL TRANSLATE_X (X,N,X0, XL01, XU01, -1)

        I_EVAL = I_EVAL + 1
        SMU0 = 0.34130
        SSIG0 = 0.0014037

        N_F = N_F +1
        N_GF = N_GF +1
        N_GC = N_GC +1

c        N=2
        NSAMP = 0
        IID = 0
        XSIG_IN = 0.2
        DO I = 1, N
        XMU_IN(I) = X0(I)
        ENDDO

C     EVLUATE OBJECTIVE FUNCTION AND ITS SENSITIVITY
C        IF (IOBJ_CONS .EQ. 1 .OR. IOBJ_CONS .EQ. 0) THEN
        NBAS = NBAS_R
c        NBAS(2) = 2
        NBAS_SM = 3
        NGAUSS = NGAUSS_R
c        NGAUSS(2)= 3
        NGAUSS_SM = 4
        NGAUSS_GQ = 5

        CALL PDD_SEN (X0, XMU_IN, XSIG_IN, NSAMP,NBAS, NBAS_SM, NGAUSS,
     1 NGAUSS_SM, IID,  NGAUSS_GQ, 1, IMCS, MOMSEN_OUT, IFLAG_AIJK,
     2   NVAR)

        TMP = DSQRT((MOMSEN_OUT(1,2)-MOMSEN_OUT(1,1)**2))
        PRINT*, '-----------MU0, SIG0--------------'
        PRINT*, MOMSEN_OUT(1,1), TMP
        CALL SLEEP(2)
        WRITE(302,'(I8, F20.10, F20.10)'),
     1    I_EVAL, MOMSEN_OUT(1,1), TMP

        OBJ(1) = 0.5*TMP/SSIG0 + 0.5*MOMSEN_OUT(1,1)/SMU0
        OBJ(2) = 0.5*0.5*(MOMSEN_OUT(2,2) -  2.0D0* MOMSEN_OUT(1,1)
     1   *MOMSEN_OUT(2,1))/(TMP* SSIG0) + 0.5*MOMSEN_OUT(2,1)/SMU0
        OBJ(3) = 0.5*0.5* (MOMSEN_OUT(3,2) -  2.0D0* MOMSEN_OUT(1,1)
     1   *MOMSEN_OUT(3,1))/(TMP * SSIG0)  + 0.5*MOMSEN_OUT(3,1)/SMU0
        OBJ(4) = 0.5*0.5* (MOMSEN_OUT(4,2) -  2.0D0* MOMSEN_OUT(1,1)
     1   *MOMSEN_OUT(4,1))/(TMP * SSIG0)  + 0.5*MOMSEN_OUT(4,1)/SMU0
        OBJ(5) = 0.5*0.5*(MOMSEN_OUT(5,2) -  2.0D0* MOMSEN_OUT(1,1)
     1   *MOMSEN_OUT(5,1))/(TMP* SSIG0) + 0.5*MOMSEN_OUT(5,1)/SMU0
        OBJ(6) = 0.5*0.5* (MOMSEN_OUT(6,2) -  2.0D0* MOMSEN_OUT(1,1)
     1   *MOMSEN_OUT(6,1))/(TMP * SSIG0)  + 0.5*MOMSEN_OUT(6,1)/SMU0
        OBJ(7) = 0.5*0.5* (MOMSEN_OUT(7,2) -  2.0D0* MOMSEN_OUT(1,1)
     1   *MOMSEN_OUT(7,1))/(TMP * SSIG0)  + 0.5*MOMSEN_OUT(7,1)/SMU0
        OBJ(8) = 0.5*0.5*(MOMSEN_OUT(8,2) -  2.0D0* MOMSEN_OUT(1,1)
     1   *MOMSEN_OUT(8,1))/(TMP* SSIG0) + 0.5*MOMSEN_OUT(8,1)/SMU0
        OBJ(9) = 0.5*0.5* (MOMSEN_OUT(9,2) -  2.0D0* MOMSEN_OUT(1,1)
     1   *MOMSEN_OUT(9,1))/(TMP * SSIG0)  + 0.5*MOMSEN_OUT(9,1)/SMU0
        OBJ(10) = 0.5*0.5* (MOMSEN_OUT(10,2) -  2.0D0* MOMSEN_OUT(1,1)
     1   *MOMSEN_OUT(10,1))/(TMP * SSIG0)  + 0.5*MOMSEN_OUT(10,1)/SMU0

        OBJ = OBJ * 100.0

C        WRITE(301,*) '-----I_EVAL =', I_EVAL,'   OBJ/G_OBJ---------'

        IF (IFLAG_AIJK .EQ. 0) THEN
        WRITE(301,'(I8,A40)' ),
     1    I_SQP,'-------------------------------------------------'
        ENDIF

        WRITE(301,'(I8, F20.10, F20.10, F20.10, F20.10, F20.10,
     1       F20.10, F20.10, F20.10, F20.10, F20.10)'),
     1    I_EVAL, OBJ(1),OBJ(2),OBJ(3),OBJ(4),OBJ(5),OBJ(6),OBJ(7),
     2       OBJ(8),OBJ(9),OBJ(10)

C        PAUSE 1

C     EVLUATE CONSTRAINT 1 AND ITS SENSITIVITY
C      ELSEIF (IOBJ_CONS .EQ. 2 .OR. IOBJ_CONS .EQ. 0) THEN
c        NBAS = 4
        NBAS = NBAS_R
c        NBAS(4) = 1
        NBAS_SM = 3
c        NGAUSS = 5
        NGAUSS = NGAUSS_R
c        NGAUSS(4)= 2
        NGAUSS_SM = 4
        NGAUSS_GQ = 5

        CALL PDD_SEN (X0, XMU_IN, XSIG_IN, NSAMP,NBAS, NBAS_SM, NGAUSS,
     1 NGAUSS_SM, IID,  NGAUSS_GQ, 2, IMCS, MOMSEN_OUT, IFLAG_AIJK,
     2   NVAR)

        TMP = DSQRT((MOMSEN_OUT(1,2)-MOMSEN_OUT(1,1)**2))
        WRITE(302,'(A8, F20.10, F20.10)'),
     1    'G1', MOMSEN_OUT(1,1), TMP

      IF  (I_RECIPROCAL .EQ. 0) THEN

        G(1) =  3.0D0*  TMP - MOMSEN_OUT(1,1)
        TMP1 =  MOMSEN_OUT(2,2) -  2.0D0* MOMSEN_OUT(1,1)
     1   *MOMSEN_OUT(2,1)
        G(2) =  3.0D0*0.5* TMP1/TMP - MOMSEN_OUT(2,1)

        TMP1 =  MOMSEN_OUT(3,2) -  2.0D0* MOMSEN_OUT(1,1)
     1   *MOMSEN_OUT(3,1)
        G(3) = 3.0D0*0.5* TMP1/TMP - MOMSEN_OUT(3,1)

        TMP1 =  MOMSEN_OUT(4,2) -  2.0D0* MOMSEN_OUT(1,1)
     1   *MOMSEN_OUT(4,1)
        G(4) = 3.0D0*0.5* TMP1/TMP - MOMSEN_OUT(4,1)
        TMP1 =  MOMSEN_OUT(5,2) -  2.0D0* MOMSEN_OUT(1,1)
     1   *MOMSEN_OUT(5,1)
        G(5) = 3.0D0*0.5* TMP1/TMP - MOMSEN_OUT(5,1)
        TMP1 =  MOMSEN_OUT(6,2) -  2.0D0* MOMSEN_OUT(1,1)
     1   *MOMSEN_OUT(6,1)
        G(6) = 3.0D0*0.5* TMP1/TMP - MOMSEN_OUT(6,1)
        TMP1 =  MOMSEN_OUT(7,2) -  2.0D0* MOMSEN_OUT(1,1)
     1   *MOMSEN_OUT(7,1)
        G(7) = 3.0D0*0.5* TMP1/TMP - MOMSEN_OUT(7,1)
        TMP1 =  MOMSEN_OUT(8,2) -  2.0D0* MOMSEN_OUT(1,1)
     1   *MOMSEN_OUT(8,1)
        G(8) = 3.0D0*0.5* TMP1/TMP - MOMSEN_OUT(8,1)
        TMP1 =  MOMSEN_OUT(9,2) -  2.0D0* MOMSEN_OUT(1,1)
     1   *MOMSEN_OUT(9,1)
        G(9) = 3.0D0*0.5* TMP1/TMP - MOMSEN_OUT(9,1)
        TMP1 =  MOMSEN_OUT(10,2) -  2.0D0* MOMSEN_OUT(1,1)
     1   *MOMSEN_OUT(10,1)
        G(10) = 3.0D0*0.5* TMP1/TMP - MOMSEN_OUT(10,1)

        G = G/C1_SCAL
      ELSE IF (I_RECIPROCAL .EQ. 1) THEN
         G(1) =  - 1.0/(3.0D0*  TMP) + 1.0/MOMSEN_OUT(1,1)
        TMP1 =  MOMSEN_OUT(2,2) -  2.0D0* MOMSEN_OUT(1,1)
     1   *MOMSEN_OUT(2,1)
        G(2) =  -TMP/(3.0D0*0.5* TMP1) + 1.0/MOMSEN_OUT(2,1)

        TMP1 =  MOMSEN_OUT(3,2) -  2.0D0* MOMSEN_OUT(1,1)
     1   *MOMSEN_OUT(3,1)
        G(3) = -TMP/(3.0D0*0.5* TMP1) + 1.0/MOMSEN_OUT(3,1)

        TMP1 =  MOMSEN_OUT(4,2) -  2.0D0* MOMSEN_OUT(1,1)
     1   *MOMSEN_OUT(4,1)
        G(4) = -TMP/(3.0D0*0.5* TMP1) + 1.0/MOMSEN_OUT(4,1)
        TMP1 =  MOMSEN_OUT(5,2) -  2.0D0* MOMSEN_OUT(1,1)
     1   *MOMSEN_OUT(5,1)
        G(5) = -TMP/(3.0D0*0.5* TMP1) + 1.0/MOMSEN_OUT(5,1)
        TMP1 =  MOMSEN_OUT(6,2) -  2.0D0* MOMSEN_OUT(1,1)
     1   *MOMSEN_OUT(6,1)
        G(6) = -TMP/(3.0D0*0.5* TMP1) + 1.0/MOMSEN_OUT(6,1)
        TMP1 =  MOMSEN_OUT(7,2) -  2.0D0* MOMSEN_OUT(1,1)
     1   *MOMSEN_OUT(7,1)
        G(7) = -TMP/(3.0D0*0.5* TMP1) + 1.0/MOMSEN_OUT(7,1)
        TMP1 =  MOMSEN_OUT(8,2) -  2.0D0* MOMSEN_OUT(1,1)
     1   *MOMSEN_OUT(8,1)
        G(8) = -TMP/(3.0D0*0.5* TMP1) + 1.0/MOMSEN_OUT(8,1)
        TMP1 =  MOMSEN_OUT(9,2) -  2.0D0* MOMSEN_OUT(1,1)
     1   *MOMSEN_OUT(9,1)
        G(9) = -TMP/(3.0D0*0.5* TMP1) + 1.0/MOMSEN_OUT(9,1)
        TMP1 =  MOMSEN_OUT(10,2) -  2.0D0* MOMSEN_OUT(1,1)
     1   *MOMSEN_OUT(10,1)
        G(10) = -TMP/(3.0D0*0.5* TMP1) + 1.0/MOMSEN_OUT(10,1)

        G = G*C1_SCAL

      ELSE
        PRINT*, 'WRONG I_RECIPROCAL!'
        STOP
      ENDIF

        WRITE(301,'(A8, F20.10, F20.10, F20.10, F20.10, F20.10,
     1       F20.10, F20.10, F20.10, F20.10, F20.10)')
     1    'CONS', G(1),G(2),G(3),G(4),G(5),G(6),G(7),
     2       G(8),G(9),G(10)

        CLOSE(301)
        CLOSE(302)

        RETURN
      END

       SUBROUTINE TRANSLATE_X (X,N,X_O, XL, XU, I_FLAG)
CC     ----------SUBROUTINE TO TRANSLATE X
CC    -----  I_FLAG = 1 TO NORMALIZE, = -1 TO UNNORMALIZE
CC    -----  I_FLAG = 2 TO NORMALIZE TO (0,2), = -2 TO UNNORMALIZE

        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
        DOUBLE PRECISION X(200), X_O(200), XL(200), XU(200), XI(200)

        X_MIN = MINVAL(XL)
        X_MAX = MAXVAL(XU)
        TRAN  = ABS(X_MIN) + ABS(X_MAX)
        IF (I_FLAG .EQ. 1) THEN
            X_O(1:N) = X(1:N)  + TRAN
        ELSE IF (I_FLAG .EQ. -1) THEN
            X_O(1:N) = X(1:N)  - TRAN
        ELSE IF (I_FLAG .EQ. 2) THEN
            X_O(1:N) = (X(1:N)  + TRAN)/(0.1*TRAN)
        ELSE IF (I_FLAG .EQ. -2) THEN
            X_O(1:N) = X(1:N)*(0.1*TRAN)  - TRAN
        ELSE
            PRINT*, 'WRONG I_FLAG FOR TRANLATE_X()!'
            STOP
        ENDIF

        RETURN
      END
```