```
C       ===================================================================
c       Sensitivity analysis for
C       EXAMPLE 2: a cubic polynomial function
C
C
C
C
C
C       N = 4
C       Y = 500 - (X1+X2)^3 + X1 - X2 -X3 + X1*X2*X3-X4
c
C     XI ~ WEIBULL(LAMDA, K)   LAMDA =  1, K=0.5,  INCONSISTENT POLYNOMIAL HERMITE
C     GAUSS-HERMITE QUADRATURE RULE
c
C
C     CALCULATE SENSITIVITY WITH RESPECT TO LAMDA (NOT TO MEAN)
C
C
C
C
C     -------  command line parameters to run the code   -------------
C     PROGRAM.EXE NSAMP NBAS NBAS_SM NGAUSS NGAUSS_SM FN IID  D_LAMBDA
C
C*********************************************************************************************
c     Copyright (c) 2010-2014 [Xuchun Ren], All Rights Reserved.



        USE RNNOF_INT
        USE RNSET_INT
        USE GQRUL_INT
        USE ERF_INT
        USE UMACH_INT


        USE DFLIB
        INCLUDE "link_fnl_shared.h"

        IMPLICIT REAL*8 (A-H,O-Z)
        REAL*8 IND, D_LAMBDA, A_LAMBDA,A_K
        REAL*8 X(200),XMU(200),SIG(200),XX(50),QX(50),QW(50),QXFIX(2),
        1          NMU(200), NSIG(200), SYUI(200), TMPA1(200), TMPA2(200)
      REAL*8 XX_SM(50),QX_SM(50),QW_SM(50),QXFIX_SM(2),X1(200),XDL(200)
        REAL*8 YR1(:,:),YR2(:,:,:,:),YR3(:,:,:,:,:,:),H(:,:,:)
        REAL*8 SMR1(:,:),SMR2(:,:,:,:),SMR3(:,:,:,:,:,:),SMH(:,:,:),
     1      SGR1(:,:),SGR2(:,:,:,:),SGR3(:,:,:,:,:,:),SGH(:,:,:)
        REAL*8 ALFA(:,:),PSIF(:,:),BITA(:,:,:,:),GAMA(:,:,:,:,:,:)
        REAL*8 ALFA_SM(:,:),PSIF_SM(:,:),BITA_SM(:,:,:,:),
        1          GAMA_SM(:,:,:,:,:,:),
     2      ALFA_SG(:,:),PSIF_SG(:,:),BITA_SG(:,:,:,:),
        1          GAMA_SG(:,:,:,:,:,:),
     3      ALFA_SMIID(:,:),ALFA_SGIID(:,:),SCMU(:,:,:),SCSIG(:,:,:),
     4      YPDD(:,:,:), H_GQ(:,:,:)
      ALLOCATABLE YR1,YR2,YR3,H,ALFA,PSIF,BITA,GAMA,SMR1,SMR2,SMR3,SMH,
     1              ALFA_SM,PSIF_SM,BITA_SM,GAMA_SM,
     2      SGR1,SGR2,SGR3,SGH,
     3              ALFA_SG,PSIF_SG,BITA_SG,GAMA_SG,
     4          ALFA_SGIID, ALFA_SMIID,SCMU,SCSIG,YPDD,H_GQ





      INTEGER N
        COMMON   /XR01/ XMU,SIG, NMU, NSIG, NYQ, N, NGAUSS_GQ,  NBAS, A_LAMBDA
        1  ,A_K

        DOUBLE PRECISION TMP(1), YLIM_SEN_K_U(71),YLIM_SEN_K_B(71),
     1  YLIM_SEN_K_T(71), YLIM_SEN_L_U(71),YLIM_SEN_L_B(71),
     1  YLIM_SEN_L_T(71),YLIM_SEN_L_MCS(71),
     &  YLIM_SEN_K_MCS(71),YLIM_PF_MCS(71),YLIM_PF_MCS2(71)
C


        INTEGER(2) n1, n2, n3, n4, n5, n6, n7, n8,n9, status
        CHARACTER(80) buf, FN
        CHARACTER(80) MSG,MSG1
        MSG = 'PROGRAM NSAMP NBAS NBAS_SM NGAUSS NGAUSS_SM FN IID
     1          D_LAMBDA'
        MSG1 = 'IID = 0(INDEPENDENT), 1(IID)'

       YLIM_SEN_K_U = 0.
       YLIM_SEN_K_B = 0.
       YLIM_SEN_K_T = 0.
       YLIM_SEN_L_U = 0.
       YLIM_SEN_L_B = 0.
       YLIM_SEN_L_T = 0.
       YLIM_SEN_K_MCS = 0.
       YLIM_SEN_L_MCS = 0.

       A_LAMBDA = 1.0
       A_K  = 0.5

        n1 = 1
        n2 = 2
        n3 = 3
        n4 = 4
        n5 = 5
        n6 = 6
        n7 = 7
        n8 = 8
        n9 = 9
        n10 = 10

        DO I =1,20
          X1(I)=0.
        ENDDO
```

```
        CALL GETARG(n1, buf, status)
        IF (status .EQ. -1) THEN
        PRINT*, 'WRONG NSAMP!'
        PRINT*, MSG
        PRINT*, MSG1
        STOP
        ENDIF
        Read( buf, '(i10)' )  NSAMP
        PRINT*, 'NSAMP = '
        WRITE (*,*) NSAMP



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

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



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


        CALL GETARG(n5, buf, status)
        IF (status .EQ. -1) THEN
        PRINT*, 'WRONG NGAUSS!'
        PRINT*, MSG
        PRINT*, MSG1
        STOP
        ENDIF
        Read( buf, '(i10)' )  NGAUSS_SM
        PRINT*, 'NGAUSS_SM = '
        WRITE (*,*) NGAUSS_SM


        CALL GETARG(n6, FN, status )
        IF (status .EQ. -1) THEN
        PRINT*, 'WRONG FILE NAME!'
        PRINT*, MSG
        PRINT*, MSG1
        STOP
        ENDIF
        PRINT*, 'FILE NAME IS ---'
        WRITE (*,*) FN


        CALL GETARG(n7, buf, status )
        IF (status .EQ. -1) THEN
        PRINT*, 'WRONG WHEN READING IID!'
        PRINT*, MSG
        PRINT*, MSG1
        STOP
        ENDIF
        Read( buf, '(i10)' )  IID
        PRINT*, 'IID = '
        WRITE (*,*) IID



        CALL GETARG(n8, buf, status )
        IF (status .EQ. -1) THEN
        PRINT*, 'WRONG WHEN READING d_lambda (FOR MCS2)!'
        PRINT*, MSG
        PRINT*, MSG1
        STOP
        ENDIF
        Read( buf, '(F10.6)' )  D_LAMBDA
        PRINT*, 'D_LAMBDA = '
        WRITE (*,*) D_LAMBDA





      OPEN(10, FILE=trim(FN)//'_U.dat', STATUS='UNKNOWN')
      OPEN(11, FILE=trim(FN)//'_U_sen_K.dat', STATUS='UNKNOWN')
      OPEN(12, FILE=trim(FN)//'_U_sen_LAMBDA.dat', STATUS='UNKNOWN')
      OPEN(20, FILE=trim(FN)//'_B.dat', STATUS='UNKNOWN')
      OPEN(21, FILE=trim(FN)//'_B_sen_K.dat', STATUS='UNKNOWN')
      OPEN(22, FILE=trim(FN)//'_B_sen_LAMBDA.dat', STATUS='UNKNOWN')
      OPEN(30, FILE=trim(FN)//'_T.dat', STATUS='UNKNOWN')
      OPEN(31, FILE=trim(FN)//'_T_sen_K.dat', STATUS='UNKNOWN')
      OPEN(32, FILE=trim(FN)//'_T_sen_LAMBDA.dat', STATUS='UNKNOWN')
      OPEN(40, FILE=trim(FN)//'_MCS.dat', STATUS='UNKNOWN')
      OPEN(41, FILE=trim(FN)//'_MCS_sen_K.dat', STATUS='UNKNOWN')
      OPEN(411, FILE=trim(FN)//'_MCS_sen_LAMBDA.dat', STATUS='UNKNOWN')
      OPEN(42, FILE=trim(FN)//'_MCS_sen2.dat', STATUS='UNKNOWN')
      OPEN(43, FILE=trim(FN)//'_YLIM_PF_MCS.dat', STATUS='UNKNOWN')
      OPEN(44, FILE=trim(FN)//'_YLIM_PF_MCS2.dat', STATUS='UNKNOWN')
C
C       GET THE PARAMETERS FOR EACH VARIABLE/DISTRIBUTION
C
C       READ INPUT
```

```
C
C    PRINT*, 'ENTER SAMPLE SIZE FOR MCS IN THE DECOMPOSITION METHOD'
C    READ*, NSAMP
C    PRINT*, 'ENTER NOS. OF HERMITE POLY BASIS W/O ZERO-ORDER (1-9)'
C    READ*, NBAS
C    PRINT*, 'ENTER ORDER OF GAUSS-HERMITE INTEGRATION'
C    READ*, NGAUSS
       NGAUSS_GQ = 10

         N = 4
         PI = 4.*DATAN(1.0D0)
         DO I = 1,N
         XMU(I) = 0.
         SIG(I) = 1.
         ENDDO

       DO I = 1,N
             NMU(I) = 0.
               NSIG(I) = 1.
         ENDDO


         IF (NGAUSS .EQ. 0 .OR. NGAUSS_SM .EQ. 0) STOP
         CALL DGQRUL (NGAUSS,4,0.,0.,0,QXFIX,QX,QW)
         CALL DGQRUL (NGAUSS_SM,4,0.,0.,0,QXFIX_SM,QX_SM,QW_SM)
         DO I = 1,NGAUSS
         XX(I) = QX(I)*DSQRT(2.D0)
         ENDDO
         DO I = 1,NGAUSS_SM
         XX_SM(I)  = QX_SM(I)*DSQRT(2.D0)
         ENDDO


         ALLOCATE( YR1(N,NGAUSS),YR2(N,N,NGAUSS,NGAUSS),H(N,NBAS,NGAUSS),
      &          ALFA(N,NBAS),PSIF(N,NBAS),BITA(N,N,NBAS,NBAS),
      &          YR3(N,N,N,NGAUSS,NGAUSS,NGAUSS),
      &          GAMA(N,N,N,NBAS,NBAS,NBAS),
      &   SMR1(N,NGAUSS_SM),SMR2(N,N,NGAUSS_SM,NGAUSS_SM),
      &   SMH(N,NBAS_SM,NGAUSS_SM),ALFA_SM(N,NBAS_SM),PSIF_SM(N,NBAS_SM),
      &   BITA_SM(N,N,NBAS_SM,NBAS_SM),
      &          SMR3(N,N,N,NGAUSS_SM,NGAUSS_SM,NGAUSS_SM),
      &          GAMA_SM(N,N,N,NBAS_SM,NBAS_SM,NBAS_SM),
      &   SGR1(N,NGAUSS_SM),SGR2(N,N,NGAUSS_SM,NGAUSS_SM),
      &   SGH(N,NBAS_SM,NGAUSS_SM),ALFA_SG(N,NBAS_SM),PSIF_SG(N,NBAS_SM),
      &   BITA_SG(N,N,NBAS_SM,NBAS_SM),
      &          SGR3(N,N,N,NGAUSS_SM,NGAUSS_SM,NGAUSS_SM),
      &          GAMA_SG(N,N,N,NBAS_SM,NBAS_SM,NBAS_SM),
      &    ALFA_SGIID(N,NBAS_SM),ALFA_SMIID(N,NBAS_SM) )

       ALLOCATE(H_GQ(N,NBAS,NGAUSS_GQ),SCMU(NGAUSS_GQ,
      1  NGAUSS_GQ,NGAUSS_GQ),SCSIG(NGAUSS_GQ,NGAUSS_GQ,NGAUSS_GQ),
      1          YPDD(NGAUSS_GQ,NGAUSS_GQ,NGAUSS_GQ))


C
C       INITIALIZE COEFFICIENTS
C



         Y0 = 0.
         SM0 = 0.
         SG0 = 0.

       NSCORV = N
       IF (IID .EQ. 1) THEN
         IF (IID .EQ. 1 .AND. (NGAUSS .NE. NGAUSS_SM .OR.
      1           NBAS .NE. NBAS_SM)) THEN
         PRINT*,' NGAUSS != NGAUSS_SM  or NBAS != NBAS_SM '
         STOP
         ENDIF
       NSCORV = 1
       ENDIF
       NS = NSCORV

         DO I = 1,N
          DO J = 1,NBAS
          ALFA(I,J) = 0.
          ENDDO
         ENDDO
         DO I = 1,N
          DO J = 1,NBAS_SM
          ALFA_SM(I,J) = 0.
          ALFA_SG(I,J) = 0.
          ENDDO
         ENDDO

         DO I1 = 1,N
          DO I2 = 1,N
           DO J1 = 1,NBAS
            DO J2 = 1,NBAS
            BITA(I1,I2,J1,J2) = 0.
            ENDDO
           ENDDO
          ENDDO
         ENDDO

         DO I1 = 1,N
          DO I2 = 1,N
           DO J1 = 1,NBAS_SM
            DO J2 = 1,NBAS_SM
            BITA_SM(I1,I2,J1,J2) = 0.
            BITA_SG(I1,I2,J1,J2) = 0.
            ENDDO
           ENDDO
          ENDDO
         ENDDO

         DO I1 = 1,N
          DO I2 = 1,N
           DO I3 = 1,N
            DO J1 = 1,NBAS
             DO J2 = 1,NBAS
              DO J3 = 1,NBAS
              GAMA(I1,I2,I3,J1,J2,J3) = 0.
              ENDDO
             ENDDO
            ENDDO
```

```
           ENDDO
          ENDDO
         ENDDO


         DO I1 = 1,N
          DO I2 = 1,N
           DO I3 = 1,N
            DO J1 = 1,NBAS_SM
             DO J2 = 1,NBAS_SM
              DO J3 = 1,NBAS_SM
              GAMA_SM(I1,I2,I3,J1,J2,J3) = 0.
              GAMA_SG(I1,I2,I3,J1,J2,J3) = 0.
              ENDDO
             ENDDO
            ENDDO
           ENDDO
          ENDDO
         ENDDO



C
C        CALCULATE/STORE RESPONSE AT MEAN INPUT (YM  SSM)
C
         DO I = 1,N
         X(I) = 0.
         ENDDO
         CALL RESPON (X,YM)
         CALL SCOR (X,SMM,SGM)
C
C        CALCULATE/STORE UNIVARIATE RESPONSES AT GAUSS POINTS (YR1)
C
         DO IX = 1,N
          DO I = 1,N
          X(I) = 0.
          ENDDO
           DO IGAUSS = 1,NGAUSS
           X(IX) = XX(IGAUSS)
           CALL RESPON (X,Y)
           YR1(IX,IGAUSS) = Y
           ENDDO
         ENDDO



         DO IX = 1,N
          DO I = 1,N
          X(I) = 0.
          ENDDO
           DO IGAUSS = 1,NGAUSS_SM
           X(IX) = XX_SM(IGAUSS)
C          X1 = X(IX)
           CALL SCOR (X,S_MU, S_SIG)
           SMR1(IX,IGAUSS) = S_MU
           SGR1(IX,IGAUSS) = S_SIG
           ENDDO
         ENDDO



C
C        CALCULATE/STORE BIVARIATE RESPONSES AT GAUSS POINTS (YR2)
C
         DO IX = 1,N-1
          DO JX = IX+1,N
           DO I = 1,N
           X(I) = 0.
           ENDDO
C                  HERE CALCULATE THE VALUE OF Y AT THE CASE ONLY VARIABLE X(IX) AND X(JX) NONZERO  IN NGAUSS X NGAUSS INTEGRAL POINTS
            DO IGAUSS = 1,NGAUSS
             DO JGAUSS = 1,NGAUSS
             X(IX) = XX(IGAUSS)
             X(JX) = XX(JGAUSS)
             CALL RESPON (X,Y)
             YR2(IX,JX,IGAUSS,JGAUSS) = Y
             ENDDO
            ENDDO
          ENDDO
         ENDDO

         DO IX = 1,N-1
          DO JX = IX+1,N
           DO I = 1,N
           X(I) = 0.
           ENDDO
            DO IGAUSS = 1,NGAUSS_SM
             DO JGAUSS = 1,NGAUSS_SM
             X(IX) = XX_SM(IGAUSS)
             X(JX) = XX_SM(JGAUSS)
C            X1 = X(IX)
             CALL SCOR (X,S_MU, S_SIG)
             SMR2(IX,JX,IGAUSS,JGAUSS) = S_MU
             SGR2(IX,JX,IGAUSS,JGAUSS) = S_SIG
             ENDDO
            ENDDO
          ENDDO
         ENDDO



C
C
C        CALCULATE/STORE TRIVARIATE RESPONSES AT GAUSS POINTS (YR3)
C
         DO IX = 1,N-2
          DO JX = IX+1,N-1
           DO KX = JX+1,N
            DO I = 1,N
            X(I) = 0.
            ENDDO
             DO IGAUSS = 1,NGAUSS
              DO JGAUSS = 1,NGAUSS
               DO KGAUSS = 1,NGAUSS
               X(IX) = XX(IGAUSS)
               X(JX) = XX(JGAUSS)
               X(KX) = XX(KGAUSS)
               CALL RESPON (X,Y)
               YR3(IX,JX,KX,IGAUSS,JGAUSS,KGAUSS) = Y
               ENDDO
              ENDDO
             ENDDO
```

```
          ENDDO
         ENDDO
        ENDDO

        DO IX = 1,N-2
         DO JX = IX+1,N-1
          DO KX = JX+1,N
           DO I = 1,N
           X(I) = 0.
           ENDDO
            DO IGAUSS = 1,NGAUSS_SM
             DO JGAUSS = 1,NGAUSS_SM
              DO KGAUSS = 1,NGAUSS_SM
              X(IX) = XX_SM(IGAUSS)
              X(JX) = XX_SM(JGAUSS)
              X(KX) = XX_SM(KGAUSS)
C           X1 = X(IX)
              CALL SCOR (X,S_MU,S_SIG)
              SMR3(IX,JX,KX,IGAUSS,JGAUSS,KGAUSS) = S_MU
              SGR3(IX,JX,KX,IGAUSS,JGAUSS,KGAUSS) = S_SIG
              ENDDO
             ENDDO
            ENDDO
          ENDDO
         ENDDO
        ENDDO


C
C       CALCULATE/STORE BASIS FUNCTIONS AT GAUSS POINTS (H)
C
        DO I = 1,N
         DO J = 1,NBAS
          DO L = 1,N
          X(L) = 0.
          ENDDO
           DO IGAUSS = 1,NGAUSS
           X(I) = XX(IGAUSS)
           CALL BASIS (X,PSIF,N,NBAS)
         H(I,J,IGAUSS) = PSIF(I,J)
           ENDDO
         ENDDO
        ENDDO

        DO I = 1,N
         DO J = 1,NBAS_SM
          DO L = 1,N
          X(L) = 0.
          ENDDO
           DO IGAUSS = 1,NGAUSS_SM
           X(I) = XX(IGAUSS)
           CALL BASIS (X,PSIF_SM,N,NBAS_SM)
           CALL BASIS (X,PSIF_SG,N,NBAS_SM)
         SMH(I,J,IGAUSS) = PSIF_SM(I,J)
         SGH(I,J,IGAUSS) = PSIF_SG(I,J)
           ENDDO
         ENDDO
        ENDDO

C
C       CALCULATE COEFFICIENT Y0
C

        YU = 0.
        DO IX = 1,N
         DO IGAUSS = 1,NGAUSS
         YU = YU + YR1(IX,IGAUSS)*QW(IGAUSS)/DSQRT(PI)
       ENDDO
        ENDDO



        SMU = 0.
        SGU = 0.
        DO IX = 1,N
         DO IGAUSS = 1,NGAUSS_SM
         SMU = SMU + SMR1(IX,IGAUSS)*QW_SM(IGAUSS)/DSQRT(PI)
         SGU = SGU + SGR1(IX,IGAUSS)*QW_SM(IGAUSS)/DSQRT(PI)
       ENDDO
        ENDDO

C
        YB = 0.
        DO IX = 1,N-1
         DO JX = IX+1,N
          DO IGAUSS = 1,NGAUSS
           DO JGAUSS = 1,NGAUSS
           YB = YB + YR2(IX,JX,IGAUSS,JGAUSS)*(QW(IGAUSS)/DSQRT(PI))*
     &        (QW(JGAUSS)/DSQRT(PI))
         ENDDO
          ENDDO
         ENDDO
        ENDDO

        SMB = 0.
        SGB = 0.
        DO IX = 1,N-1
         DO JX = IX+1,N
          DO IGAUSS = 1,NGAUSS_SM
           DO JGAUSS = 1,NGAUSS_SM
           SMB = SMB + SMR2(IX,JX,IGAUSS,JGAUSS)*(QW_SM(IGAUSS)/DSQRT(PI))*
     &        (QW_SM(JGAUSS)/DSQRT(PI))
           SGB = SGB + SGR2(IX,JX,IGAUSS,JGAUSS)*(QW_SM(IGAUSS)/DSQRT(PI))*
     &        (QW_SM(JGAUSS)/DSQRT(PI))

         ENDDO
          ENDDO
         ENDDO
        ENDDO
```

```fortran
C
        YT = 0.
        DO IX = 1,N-2
         DO JX = IX+1,N-1
          DO KX = JX+1,N
           DO IGAUSS = 1,NGAUSS
            DO JGAUSS = 1,NGAUSS
             DO KGAUSS = 1,NGAUSS
             YT = YT + YR3(IX,JX,KX,IGAUSS,JGAUSS,KGAUSS)*
     &          (QW(IGAUSS)/DSQRT(PI))*(QW(JGAUSS)/DSQRT(PI))*
     &          (QW(KGAUSS)/DSQRT(PI))
           ENDDO
            ENDDO
           ENDDO
          ENDDO
         ENDDO
        ENDDO


        SMT = 0.
        SGT = 0.
        DO IX = 1,N-2
         DO JX = IX+1,N-1
          DO KX = JX+1,N
           DO IGAUSS = 1,NGAUSS_SM
            DO JGAUSS = 1,NGAUSS_SM
             DO KGAUSS = 1,NGAUSS_SM
             SMT = SMT + SMR3(IX,JX,KX,IGAUSS,JGAUSS,KGAUSS)*
     &          (QW_SM(IGAUSS)/DSQRT(PI))*(QW_SM(JGAUSS)/DSQRT(PI))*
     &          (QW_SM(KGAUSS)/DSQRT(PI))
             SGT = SGT + SGR3(IX,JX,KX,IGAUSS,JGAUSS,KGAUSS)*
     &          (QW_SM(IGAUSS)/DSQRT(PI))*(QW_SM(JGAUSS)/DSQRT(PI))*
     &          (QW_SM(KGAUSS)/DSQRT(PI))

           ENDDO
            ENDDO
           ENDDO
          ENDDO
         ENDDO
        ENDDO

C
C
        Y0 = YT - DFLOAT(N-3)*YB + DFLOAT((N-2)*(N-3))*YU/2.
     &     - DFLOAT((N-1)*(N-2)*(N-3))*YM/6.

C      SM0 = SMU - (N-1)*SMM
C      SG0 = SGU - (N-1)*SGM

      NS = NSCORV

      SM0 = SMU - (N-1.)*SMM
      SG0 = SGU - (N-1.)*SGM

C   SM0 = SMT - DFLOAT(N-3)*SMB + DFLOAT((N-2)*(N-3))*SMU/2.
C     &     - DFLOAT((N-1)*(N-2)*(N-3))*SMM/6.
C   SG0 = SGT - DFLOAT(N-3)*SGB + DFLOAT((N-2)*(N-3))*SGU/2.
C     &     - DFLOAT((N-1)*(N-2)*(N-3))*SGM/6.
C

C
C       CALCULATE COEFFICIENTS ALFA(I,J)
C
        DO I = 1,N
         DO J = 1,NBAS
C
         YYU = 0.
          DO IGAUSS = 1,NGAUSS
          YYU = YYU + YR1(I,IGAUSS)*H(I,J,IGAUSS)*QW(IGAUSS)/DSQRT(PI)
        ENDDO
C
          YYB = 0.
           DO 10 IX = 1,N-1
           DO 10 JX = IX+1,N
           IF (IX .NE. I .AND. JX .NE. I) GO TO 10
            DO IGAUSS = 1,NGAUSS
             DO JGAUSS = 1,NGAUSS
             IF (IX .EQ. I) KGAUSS = IGAUSS
             IF (JX .EQ. I) KGAUSS = JGAUSS
             YYB = YYB + YR2(IX,JX,IGAUSS,JGAUSS)*H(I,J,KGAUSS)*
     &           (QW(IGAUSS)/DSQRT(PI))*(QW(JGAUSS)/DSQRT(PI))
             ENDDO
            ENDDO
10         CONTINUE
C
C

          YYT = 0.
           DO 20 IX = 1,N-2
           DO 20 JX = IX+1,N-1
           DO 20 KX = JX+1,N
           IF (IX .NE. I .AND. JX .NE. I .AND. KX .NE. I) GO TO 20
            DO IGAUSS = 1,NGAUSS
             DO JGAUSS = 1,NGAUSS
              DO KGAUSS = 1,NGAUSS
              IF (IX .EQ. I) MGAUSS = IGAUSS
              IF (JX .EQ. I) MGAUSS = JGAUSS
              IF (KX .EQ. I) MGAUSS = KGAUSS
              YYT = YYT + YR3(IX,JX,KX,IGAUSS,JGAUSS,KGAUSS)*H(I,J,MGAUSS)
     &          *(QW(IGAUSS)/DSQRT(PI))*(QW(JGAUSS)/DSQRT(PI))*
     &          (QW(KGAUSS)/DSQRT(PI))
                  ENDDO
             ENDDO
            ENDDO
20         CONTINUE
C
C   --------------  IJNME PDD EQ.(26) (23)
           ALFA(I,J) = YYT - DFLOAT(N-3)*YYB + DFLOAT((N-2)*(N-3))*YYU/2.
C
         ENDDO
        ENDDO

C
```

```fortran
C        CALCULATE COEFFICIENTS ALFA_SM(I,J), ALFA_SG(I,J)
C
         DO I = 1,N
          DO J = 1,NBAS_SM
C
          SSMU = 0.
          SSGU = 0.
           DO IGAUSS = 1,NGAUSS_SM
           SSMU = SSMU + SMR1(I,IGAUSS)*SMH(I,J,IGAUSS)*QW_SM(IGAUSS)/DSQRT(PI)
           SSGU = SSGU + SGR1(I,IGAUSS)*SGH(I,J,IGAUSS)*QW_SM(IGAUSS)/DSQRT(PI)
         ENDDO
C
           SSMB = 0.
           SSGB = 0.
            DO 110 IX = 1,N-1
            DO 110 JX = IX+1,N
            IF (IX .NE. I .AND. JX .NE. I) GO TO 110
             DO IGAUSS = 1,NGAUSS_SM
              DO JGAUSS = 1,NGAUSS_SM
              IF (IX .EQ. I) KGAUSS = IGAUSS
              IF (JX .EQ. I) KGAUSS = JGAUSS
              SSMB = SSMB + SMR2(IX,JX,IGAUSS,JGAUSS)*SMH(I,J,KGAUSS)*
     &           (QW_SM(IGAUSS)/DSQRT(PI))*(QW_SM(JGAUSS)/DSQRT(PI))
              SSGB = SSGB + SGR2(IX,JX,IGAUSS,JGAUSS)*SGH(I,J,KGAUSS)*
     &           (QW_SM(IGAUSS)/DSQRT(PI))*(QW_SM(JGAUSS)/DSQRT(PI))

              ENDDO
             ENDDO
110         CONTINUE
C
C
           SSMT = 0.
           SSGT = 0.
            DO 120 IX = 1,N-2
            DO 120 JX = IX+1,N-1
            DO 120 KX = JX+1,N
            IF (IX .NE. I .AND. JX .NE. I .AND. KX .NE. I) GO TO 120
             DO IGAUSS = 1,NGAUSS_SM
              DO JGAUSS = 1,NGAUSS_SM
               DO KGAUSS = 1,NGAUSS_SM
               IF (IX .EQ. I) MGAUSS = IGAUSS
               IF (JX .EQ. I) MGAUSS = JGAUSS
               IF (KX .EQ. I) MGAUSS = KGAUSS
               SSMT = SSMT + SMR3(IX,JX,KX,IGAUSS,JGAUSS,KGAUSS)*SMH(I,J,MGAUSS)
     &           *(QW_SM(IGAUSS)/DSQRT(PI))*(QW_SM(JGAUSS)/DSQRT(PI))*
     &           (QW_SM(KGAUSS)/DSQRT(PI))
               SSGT = SSGT + SGR3(IX,JX,KX,IGAUSS,JGAUSS,KGAUSS)*SGH(I,J,MGAUSS)
     &           *(QW_SM(IGAUSS)/DSQRT(PI))*(QW_SM(JGAUSS)/DSQRT(PI))*
     &           (QW_SM(KGAUSS)/DSQRT(PI))
                   ENDDO
              ENDDO
             ENDDO
120         CONTINUE
C
            ALFA_SM(I,J) = SSMT - DFLOAT(N-3)*SSMB + DFLOAT((N-2)*(N-3))*SSMU/2.
            ALFA_SG(I,J) = SSGT - DFLOAT(N-3)*SSGB + DFLOAT((N-2)*(N-3))*SSGU/2.
C
          ENDDO
         ENDDO

C  ----------    FORMULA 14   OF LR.PDF   ----------
       IF (IID .EQ. 1) THEN
             DO I = 1,N
              DO J = 1,NBAS_SM
                ALFA_SMIID(I,J) = ALFA_SM(1,J)
                ALFA_SGIID(I,J) = ALFA_SG(1,J)
              ENDDO
             ENDDO
          END IF




C
C        CALCULATE COEFFICIENTS BITA(I1,I2,J1,J2)
C
         DO I1 = 1,N-1
          DO I2 = I1+1,N
           DO J1 = 1,NBAS
            DO J2 = 1,NBAS
C
           YYYB = 0.
             DO IGAUSS = 1,NGAUSS
              DO JGAUSS = 1,NGAUSS
              YYYB = YYYB +
     &      YR2(I1,I2,IGAUSS,JGAUSS)*H(I1,J1,IGAUSS)*H(I2,J2,JGAUSS)*
     &      (QW(IGAUSS)/DSQRT(PI))*(QW(JGAUSS)/DSQRT(PI))
              ENDDO
             ENDDO
C
           YYYT = 0.
            DO 30 IIX = 1,N-2
            DO 30 JJX = IIX+1,N-1
            DO 30 KKX = JJX+1,N
            IF (IIX .NE. I1 .AND. IIX .NE. I2 .AND.
     &       JJX .NE. I1 .AND. JJX .NE. I2 .AND.
     &       KKX .NE. I1 .AND. KKX .NE. I2) GO TO 30
            IF (IIX .EQ. I1 .AND. (JJX .EQ. I2 .OR. KKX .EQ. I2)) GO TO 31
            IF (JJX .EQ. I1 .AND. (IIX .EQ. I2 .OR. KKX .EQ. I2)) GO TO 31
            IF (KKX .EQ. I1 .AND. (IIX .EQ. I2 .OR. JJX .EQ. I2)) GO TO 31
            GO TO 30
31         DO IGAUSS = 1,NGAUSS
             DO JGAUSS = 1,NGAUSS
               DO KGAUSS = 1,NGAUSS
               IF (IIX .EQ. I1) MGAUSS1 = IGAUSS
               IF (JJX .EQ. I1) MGAUSS1 = JGAUSS
               IF (KKX .EQ. I1) MGAUSS1 = KGAUSS
               IF (IIX .EQ. I2) MGAUSS2 = IGAUSS
               IF (JJX .EQ. I2) MGAUSS2 = JGAUSS
               IF (KKX .EQ. I2) MGAUSS2 = KGAUSS
               YYYT = YYYT + YR3(IIX,JJX,KKX,IGAUSS,JGAUSS,KGAUSS)*
     &       H(I1,J1,MGAUSS1)*H(I2,J2,MGAUSS2)*(QW(IGAUSS)/DSQRT(PI))*
     &       (QW(JGAUSS)/DSQRT(PI))*(QW(KGAUSS)/DSQRT(PI))
                   ENDDO
              ENDDO
             ENDDO
30          CONTINUE
C
            BITA(I1,I2,J1,J2) = YYYT - DFLOAT(N-3)*YYYB
```

```
C       WRITE(*,*) I1, I2, J1, J2,   BITA(I1,I2,J1,J2)
C
           ENDDO
          ENDDO
         ENDDO
        ENDDO



C
C       CALCULATE COEFFICIENTS BITA_SM(I1,I2,J1,J2), BITA_SG(I1,I2,J1,J2)
C
        DO I1 = 1,N-1
         DO I2 = I1+1,N
          DO J1 = 1,NBAS_SM
           DO J2 = 1,NBAS_SM
C
          SSSMB = 0.
          SSSGB = 0.
            DO IGAUSS = 1,NGAUSS_SM
             DO JGAUSS = 1,NGAUSS_SM
             SSSMB = SSSMB +
     &     SMR2(I1,I2,IGAUSS,JGAUSS)*SMH(I1,J1,IGAUSS)*
     &      SMH(I2,J2,JGAUSS)*
     &     (QW_SM(IGAUSS)/DSQRT(PI))*(QW_SM(JGAUSS)/DSQRT(PI))
             SSSGB = SSSGB +
     &     SGR2(I1,I2,IGAUSS,JGAUSS)*SGH(I1,J1,IGAUSS)*
     &      SGH(I2,J2,JGAUSS)*
     &     (QW_SM(IGAUSS)/DSQRT(PI))*(QW_SM(JGAUSS)/DSQRT(PI))


             ENDDO
            ENDDO
C
           SSSMT = 0.
           SSSGT = 0.
           DO 130 IIX = 1,N-2
           DO 130 JJX = IIX+1,N-1
           DO 130 KKX = JJX+1,N
           IF (IIX .NE. I1 .AND. IIX .NE. I2 .AND.
     &       JJX .NE. I1 .AND. JJX .NE. I2 .AND.
     &       KKX .NE. I1 .AND. KKX .NE. I2) GO TO 130
           IF (IIX .EQ. I1 .AND. (JJX .EQ. I2 .OR. KKX .EQ. I2)) GO TO 131
           IF (JJX .EQ. I1 .AND. (IIX .EQ. I2 .OR. KKX .EQ. I2)) GO TO 131
           IF (KKX .EQ. I1 .AND. (IIX .EQ. I2 .OR. JJX .EQ. I2)) GO TO 131
           GO TO 130
131        DO IGAUSS = 1,NGAUSS_SM
             DO JGAUSS = 1,NGAUSS_SM
              DO KGAUSS = 1,NGAUSS_SM
              IF (IIX .EQ. I1) MGAUSS1 = IGAUSS
              IF (JJX .EQ. I1) MGAUSS1 = JGAUSS
              IF (KKX .EQ. I1) MGAUSS1 = KGAUSS
              IF (IIX .EQ. I2) MGAUSS2 = IGAUSS
              IF (JJX .EQ. I2) MGAUSS2 = JGAUSS
              IF (KKX .EQ. I2) MGAUSS2 = KGAUSS
              SSSMT = SSSMT + SMR3(IIX,JJX,KKX,IGAUSS,JGAUSS,KGAUSS)*
     &      SMH(I1,J1,MGAUSS1)*SMH(I2,J2,MGAUSS2)*
     &      (QW_SM(IGAUSS)/DSQRT(PI))*
     &      (QW_SM(JGAUSS)/DSQRT(PI))*(QW_SM(KGAUSS)/DSQRT(PI))

              SSSGT = SSSGT + SGR3(IIX,JJX,KKX,IGAUSS,JGAUSS,KGAUSS)*
     &      SGH(I1,J1,MGAUSS1)*SGH(I2,J2,MGAUSS2)*
     &      (QW_SM(IGAUSS)/DSQRT(PI))*
     &      (QW_SM(JGAUSS)/DSQRT(PI))*(QW_SM(KGAUSS)/DSQRT(PI))

                  ENDDO
             ENDDO
            ENDDO
130        CONTINUE
C
           BITA_SM(I1,I2,J1,J2) = SSSMT - DFLOAT(N-3)*SSSMB
           BITA_SG(I1,I2,J1,J2) = SSSGT - DFLOAT(N-3)*SSSGB

C       WRITE(*,*) I1, I2, J1, J2,   BITA(I1,I2,J1,J2)
C
           ENDDO
          ENDDO
         ENDDO
        ENDDO




c       CALCULATE COEFFICIENTS GAMA(I1,I2,I3,J1,J2,J3)

      DO I1 = 1,N-2
      DO I2 = I1+1,N-1
      DO I3 = I2+1,N
      DO J1 = 1,NBAS
       DO J2 = 1,NBAS
       DO J3 = 1,NBAS

        GAMA(I1,I2,I3,J1,J2,J3) = 0.
           DO IGAUSS = 1,NGAUSS
           DO JGAUSS = 1,NGAUSS
           DO KGAUSS = 1,NGAUSS
              GAMA(I1,I2,I3,J1,J2,J3) = GAMA(I1,I2,I3,J1,J2,J3) +
     &         YR3(I1,I2,I3,IGAUSS,JGAUSS,KGAUSS)*
     &        H(I1,J1,IGAUSS)*H(I2,J2,JGAUSS)*H(I3,J3,KGAUSS)*
     &       (QW(IGAUSS)/DSQRT(PI))*
     &       (QW(JGAUSS)/DSQRT(PI))*(QW(KGAUSS)/DSQRT(PI))
           ENDDO
          ENDDO
         ENDDO
CC
        ENDDO
       ENDDO
      ENDDO
      ENDDO
      ENDDO
      ENDDO
```

```
C
C
C       CALCULATE COEFFICIENTS GAMA_SM(I1,I2,I3,J1,J2,J3), GAMA_SG(I1,I2,I3,J1,J2,J3)
C
C   DO I1 = 1,N-2
C    DO I2 = I1+1,N-1
C     DO I3 = I2+1,N
C      DO J1 = 1,NBAS_SM
C       DO J2 = 1,NBAS_SM
C        DO J3 = 1,NBAS_SM
CC
C        GAMA_SM(I1,I2,I3,J1,J2,J3) = 0.
C        GAMA_SG(I1,I2,I3,J1,J2,J3) = 0.
C            DO IGAUSS = 1,NGAUSS_SM
C          DO JGAUSS = 1,NGAUSS_SM
C           DO KGAUSS = 1,NGAUSS_SM
C              GAMA_SM(I1,I2,I3,J1,J2,J3) = GAMA_SM(I1,I2,I3,J1,J2,J3) +
C     &         SMR3(I1,I2,I3,IGAUSS,JGAUSS,KGAUSS)*
C     &        SMH(I1,J1,IGAUSS)*SMH(I2,J2,JGAUSS)*SMH(I3,J3,KGAUSS)*
C     &       (QW_SM(IGAUSS)/DSQRT(PI))*
C     &       (QW_SM(JGAUSS)/DSQRT(PI))*(QW_SM(KGAUSS)/DSQRT(PI))
C
C              GAMA_SG(I1,I2,I3,J1,J2,J3) = GAMA_SG(I1,I2,I3,J1,J2,J3) +
C     &         SGR3(I1,I2,I3,IGAUSS,JGAUSS,KGAUSS)*
C     &        SGH(I1,J1,IGAUSS)*SGH(I2,J2,JGAUSS)*SGH(I3,J3,KGAUSS)*
C     &       (QW_SM(IGAUSS)/DSQRT(PI))*
C     &       (QW_SM(JGAUSS)/DSQRT(PI))*(QW_SM(KGAUSS)/DSQRT(PI))
C
C           ENDDO
C          ENDDO
C         ENDDO
CC
C        ENDDO
C       ENDDO
C      ENDDO
C     ENDDO
C    ENDDO
C   ENDDO
C
```

```
C


C
C       NOW CALCULATE THE SENSIVITY OF Q-TH MOMENT BY ANALYTICAL EXPRESSION
C
        PRINT*, 'THE SENSITIVITY OF q-th MOMENT  OF Y  IS  ....'
C

      IF (IID .EQ. 1) THEN
          PRINT*,  'INDEPENDENT IDENTICAL DISTRIBUTION ...'
          NBAS_SEN = MIN(NBAS, NBAS_SM)
          SENSM = N* SM0*Y0
          SENSG = N*SG0*Y0
          DO I = 1,N
              DO J = 1,NBAS_SEN
                SENSM =  SENSM + ALFA(I,J)*ALFA_SMIID(I,J)
                SENSG =  SENSG + ALFA(I,J)*ALFA_SGIID(I,J)
              ENDDO
          ENDDO

          WRITE(*,*) 'SENSM = ', SENSM
          WRITE(*,*) 'SENSG = ', SENSG
      ELSE IF (IID .EQ. 0) THEN
          PRINT*,  'INDEPENDENT BUT NOT INDENTICAL DISTRIBUTION
     1              NOT READY YET ... '
C          NBAS_SEN = MIN(NBAS, NBAS_SM)
C          SENSM = N* SM0
C          SENSG = N*SG0
C          DO I = 1,N
C              DO J = 1,NBAS_SEN
C                SENSM =  SENSM + ALFA(I,J)*ALFA_SM(I,J)
C                SENSG =  SENSG + ALFA(I,J)*ALFA_SG(I,J)
C              ENDDO
C          ENDDO
C
      ELSE IF (IID .EQ. 2) THEN
          PRINT*,  'DEPENDENT DISTRIBUTION
     1              NOT READY YET ... '
C          NBAS_SEN = MIN(NBAS, NBAS_SM)
C          SENSM = N* SM0
C          SENSG = N*SG0
C          DO I = 1,N
C              DO J = 1,NBAS_SEN
C                SENSM =  SENSM + ALFA(I,J)*ALFA_SM(I,J)
C                SENSG =  SENSG + ALFA(I,J)*ALFA_SG(I,J)
C              ENDDO
C          ENDDO
C
      END IF


C
C       NOW START SIMULATION FOR UNIVARIATE/BIVARIATE/DIRECT MCS
C
        PRINT*, 'STARTING SIMULATION ....'
C
C     PFU = FAULURE PROBABLITY OF UNIVARIATE APPROX,
C     PFB =                       BIVARIATE
C     PFT =                       TRIVARIATE
C     PFMCS =                     DIRECT MCS
C     SMOMU = MOMENT CALC BY UNIVARIATE APPROX
C     SMUSU = SENSITIVITY TO MU OF MOMENT CALC BY UNIVARIATE APPROX
C     SSIGSU = SENSITIVITY TO SIG OF MOMENT CALC BY UNIVARIATE APPROX
C     SMOMB = MOMENT CALC BY BIVARIATE APPROX
C     SMUSB = SENSITIVITY TO MU OF MOMENT CALC BY BIVARIATE APPROX
C     SSIGSB = SENSITIVITY TO SIG OF MOMENT CALC BY BIVARIATE APPROX
C     SMOMT = MOMENT CALC BY TRIVARIATE APPROX
C     SMUST = SENSITIVITY TO MU OF MOMENT CALC BY TRIVARIATE APPROX
C     SSIGST = SENSITIVITY TO SIG OF MOMENT CALC BY TRIVARIATE APPROX

C     SPFU_MU =  SENSITIVITY OF PROBABILITY OF FAILURE TO MU BY UNIVARITE METHOD
C     SPFU_SIG =  SENSITIVITY OF PROBABILITY OF FAILURE TO SIG BY UNIVARITE METHOD
C     SPFB_MU =  SENSITIVITY OF PROBABILITY OF FAILURE TO MU BY BIVARITE METHOD
C     SPFB_SIG =  SENSITIVITY OF PROBABILITY OF FAILURE TO SIG BY BIVARITE METHOD
C     SPFT_MU =  SENSITIVITY OF PROBABILITY OF FAILURE TO MU BY TRIVARITE METHOD
C     SPFT_SIG =  SENSITIVITY OF PROBABILITY OF FAILURE TO SIG BY TRIVARITE METHOD







      SPFU_MU =  0.0
      SPFU_SIG =  0.0
      SPFB_MU =  0.0
      SPFB_SIG =  0.0
      SPFT_MU =  0.0
      SPFT_SIG =  0.0



        PFU = 0.
        PFB = 0.
        PFT = 0.
        PFMCS = 0.
        SMOMU = 0.
        SMUSU = 0.
        SSIGSU = 0.
        SMOMB = 0.
        SMUSB = 0.
        SSIGSB = 0.
        SMOMT = 0.
        SMUST = 0.
        SSIGST = 0.


        Y_EXC_M1 = 0.
        Y_EXC_M2 = 0.
        Y_EXC_M3 = 0.
        Y_EXC_MU = 0.
        Y_EXC_SIG = 0.



        YLIM_PF_MCS = 0
        YLIM_PF_MCS2 = 0
```

```
C
        KK = 1
      KSTEP = NSAMP/10
        ISEED = 123457
        CALL RNSET(ISEED)
        OPEN(108,FILE='BIVARIATE_GHI2.TXT')
        DO 100 ISAMP = 1,NSAMP
C
        DO I = 1,N
C     This function generates a pseudorandom number from a standard normal distribution.
        X(I) = DRNNOF()
        ENDDO
c
        XDL = X/(1.0+D_LAMBDA*NMU(1))

        CALL BASIS (X,PSIF,N,NBAS)
c       CALL FX_FU_JU(X, Y_FFJ)
C
C       UNIVARIATE APPROXIMATION
C
      SEN_F = 1

        SYU = Y0
        DO I = 1,N
         DO J = 1,NBAS
         SYU = SYU + ALFA(I,J)*PSIF(I,J)
         ENDDO
        ENDDO
        IND = 0.
        IF (SYU .LE. 0.) IND = 1.
        PFU = PFU + IND
        CALL SCOR3_WEIBULL (X,S_MU, S_SIG)
        SPFU_MU = SPFU_MU + IND * S_MU
        SPFU_SIG = SPFU_SIG + IND * S_SIG
        IF (SEN_F .EQ. 1) THEN
        DO JJ = 10,-60,-1
           IND = 0.
           IF (SYU .LE. JJ*50) THEN
           IND = 1.
           YLIM_SEN_K_U(JJ+61) = YLIM_SEN_K_U(JJ+61) + IND * S_MU
           YLIM_SEN_L_U(JJ+61) = YLIM_SEN_L_U(JJ+61) + IND * S_SIG
           ELSE
           exit
           ENDIF
        ENDDO
        ENDIF

C   SYU = 0.
C
C   DO I = 1,N
C    SYUI(I) = 0.
C    DO J = 1,NBAS
C    SYUI(I) = SYUI(I)+ ALFA(I,J)*PSIF(I,J)
C    ENDDO
C    SYU = SYU + SYUI(I)
C   ENDDO
C   SYU = SYU + Y0
C   IND = 0.
C   IF (SYU .LE. 0.) IND = 1.
C   PFU = PFU + IND
C
C
C       UNIVARIATE APPROXIMATION OF MOMENT AND SENSITIVITY
C     SMUU  ---- UNIVIATE APPROXIMATION/EXACT VALUE OF SCORE FUNCTION OF MU
C     SSIGU  ---- UNIVIATE APPROXIMATION/EXACT VALUE OF SCORE FUNCTION OF SIG  (MAY CALL SCOR () TO GET THE EXACT VALUE)
C     FOR IDD CASE
        SMUU =  SM0
        SSIGU = SG0

       DO I = 1,N
           DO J = 1,NBAS
            SMUU = SMUU + ALFA_SMIID(I,J)*PSIF(I,J)
            SSIGU = SSIGU + ALFA_SGIID(I,J)*PSIF(I,J)
        ENDDO
           ENDDO

c       CALL SCOR2 (X,SMUU, SSIGU)

       CALL SCOR3_WEIBULL (X,SMUU_EX, SSIGU_EX)
       IF (SMUU_EX .NE. SMUU .OR. SSIGU_EX .NE. SSIGU) THEN
C         PRINT*, 'SMUU_EX != SMUU .OR. SSIGU_EX != SSIGU'
C       WRITE (*,*) SMUU_EX, SMUU, SSIGU_EX, SSIGU
       ENDIF

c       SMUU =  SMUU_EX
c       SSIGU = SSIGU_EX





      IF (IID .NE. 1) THEN
        PRINT*, 'NOT READY FOR IID != 1 SIMULATION'
        STOP
      ENDIF
      CALL PDF_IID(X,YPDF)
      SMOMU = SMOMU + SYU
      SMUSU = SMUSU + SMUU*SYU
      SSIGSU = SSIGSU + SSIGU*SYU


C
C       BIVARIATE APPROXIMATION
C
        SYB = SYU
        DO I1 = 1,N-1
         DO I2 = I1+1,N
          DO J1 = 1,NBAS
           DO J2 = 1,NBAS
           SYB = SYB + BITA(I1,I2,J1,J2)*PSIF(I1,J1)*PSIF(I2,J2)
           ENDDO
          ENDDO
```

```
         ENDDO
        ENDDO
        IND = 0.
        IF (SYB .LE. 0.) IND = 1.
        PFB = PFB + IND
        SPFB_MU = SPFB_MU + IND * S_MU
        SPFB_SIG = SPFB_SIG + IND * S_SIG
        IF (SEN_F .EQ. 1) THEN
        DO JJ = 10,-60,-1
           IND = 0.
           IF (SYB .LE. JJ*50) THEN
           IND = 1.
           YLIM_SEN_K_B(JJ+61) = YLIM_SEN_K_B(JJ+61) + IND * S_MU
           YLIM_SEN_L_B(JJ+61) = YLIM_SEN_L_B(JJ+61) + IND * S_SIG
           ELSE
           exit
           ENDIF
        ENDDO
        ENDIF


C   IF (SYB .LE. 0.) THEN
C     WRITE(*,*) 'BINGO'
C   ENDIF


        WRITE(108,'(I8,F18.6)') ISAMP , SYB
C
C     BIVARIATE APPROXIMATE FOR MOMENT AND SENSITIVITY
C
      SMOMB = SMOMB + SYB
      SMUSB = SMUSB + SMUU*SYB
      SSIGSB = SSIGSB + SSIGU*SYB


C
C
C       TRIVARIATE APPROXIMATION
C
         SYT = SYB
         DO I1 = 1,N-2
         DO I2 = I1+1,N-1
         DO I3 = I2+1,N
         DO J1 = 1,NBAS
         DO J2 = 1,NBAS
         DO J3 = 1,NBAS
         SYT = SYT + GAMA(I1,I2,I3,J1,J2,J3)*PSIF(I1,J1)*PSIF(I2,J2)*
     &           PSIF(I3,J3)
         ENDDO
         ENDDO
         ENDDO
         ENDDO
         ENDDO
         ENDDO
        IND = 0.
        IF (SYT .LE. 0.) IND = 1.
        PFT = PFT + IND
        SPFT_MU = SPFT_MU + IND * S_MU
        SPFT_SIG = SPFT_SIG + IND * S_SIG
        IF (SEN_F .EQ.1) THEN
        DO JJ = 10,-60,-1
           IND = 0.
           IF (SYT .LE. JJ*50) THEN
           IND = 1.
           YLIM_SEN_K_T(JJ+61) = YLIM_SEN_K_T(JJ+61) + IND * S_MU
           YLIM_SEN_L_T(JJ+61) = YLIM_SEN_L_T(JJ+61) + IND * S_SIG
           ELSE
           exit
           ENDIF
        ENDDO
        ENDIF


C
C     BIVARIATE APPROXIMATE FOR MOMENT AND SENSITIVITY
C
      SMOMT = SMOMT + SYT
      SMUST = SMUST + SMUU*SYT
      SSIGST = SSIGST + SSIGU*SYT



C
C       DIRECT MCS
C

        IND = 0.
        CALL RESPON (X,SYMCS)
        CALL RESPON (XDL,SYMCS2)
        Y_EXC_M1 = Y_EXC_M1 + SYMCS
        Y_EXC_M2 = Y_EXC_M2 + SYMCS**2
        Y_EXC_M3 = Y_EXC_M3 + SYMCS**3
        Y_EXC_MU = Y_EXC_MU + SYMCS * SMUU
        Y_EXC_SIG = Y_EXC_SIG + SYMCS * SSIGU
        IF (SYMCS .LE. 0.) IND = 1.
        PFMCS = PFMCS + IND
        PFMCS_MU = PFMCS_MU + IND*S_MU
C
      IF (SEN_F .EQ. 1) THEN
        DO JJ = 10,-60,-1
           IND = 0.
           IF (SYMCS .LE. JJ*50) THEN
           IND = 1.
           YLIM_PF_MCS(JJ+61) = YLIM_PF_MCS(JJ+61) + 1
           YLIM_SEN_K_MCS(JJ+61) = YLIM_SEN_K_MCS(JJ+61) + IND  * S_MU
           YLIM_SEN_L_MCS(JJ+61) = YLIM_SEN_L_MCS(JJ+61) + IND  * S_SIG
           ELSE
           exit
           ENDIF
        ENDDO
        DO JJ = 10,-60,-1
           IF (SYMCS2 .LE. JJ*50) THEN
           YLIM_PF_MCS2(JJ+61) = YLIM_PF_MCS2(JJ+61) + 1
           ELSE
           exit
           ENDIF
        ENDDO

        ENDIF
```

```fortran
          WRITE (10,*) SYU
          WRITE (20,*) SYB
          WRITE (30,*) SYT
          WRITE (40,*) SYMCS
C         PRINT*, Y0,SYU,SYB,SYT,SYMCS
C
      IF (ISAMP .EQ. KSTEP*KK) GO TO 28
      GO TO 100
28    KKK = KSTEP*KK
          PRINT*, KKK, ' SAMPLES .....'
      KK = KK + 1
C
100       CONTINUE
C
C
          IF (SEN_F .EQ. 1) THEN
          DO JJ = 1,71
          WRITE (11,*) YLIM_SEN_K_U(JJ)/DFLOAT(NSAMP)
          WRITE (12,*) YLIM_SEN_L_U(JJ)/DFLOAT(NSAMP)
          WRITE (21,*) YLIM_SEN_K_B(JJ)/DFLOAT(NSAMP)
          WRITE (22,*) YLIM_SEN_L_B(JJ)/DFLOAT(NSAMP)
          WRITE (31,*) YLIM_SEN_K_T(JJ)/DFLOAT(NSAMP)
          WRITE (32,*) YLIM_SEN_L_T(JJ)/DFLOAT(NSAMP)
          WRITE (41,*) YLIM_SEN_K_MCS(JJ)/DFLOAT(NSAMP)
          WRITE (411,*) YLIM_SEN_L_MCS(JJ)/DFLOAT(NSAMP)
          WRITE (42,*) (YLIM_PF_MCS2(JJ)-YLIM_PF_MCS(JJ))/
     1  (DFLOAT(NSAMP)*D_LAMBDA)
          WRITE (43,*) YLIM_PF_MCS(JJ)
          WRITE (44,*) YLIM_PF_MCS2(JJ)

          ENDDO
          ENDIF

          PFU = PFU/DFLOAT(NSAMP)
          PFB = PFB/DFLOAT(NSAMP)
          PFT = PFT/DFLOAT(NSAMP)
          PFMCS = PFMCS/DFLOAT(NSAMP)
          PFMCS_MU = PFMCS_MU/DFLOAT(NSAMP)

          SPFU_MU = SPFU_MU/DFLOAT(NSAMP)
          SPFU_SIG = SPFU_SIG/DFLOAT(NSAMP)
          SPFB_MU = SPFB_MU/DFLOAT(NSAMP)
          SPFB_SIG = SPFB_SIG/DFLOAT(NSAMP)
          SPFT_MU = SPFT_MU/DFLOAT(NSAMP)
          SPFT_SIG = SPFT_SIG/DFLOAT(NSAMP)



      SMOMU = SMOMU/(DFLOAT(NSAMP))
      SMUSU = SMUSU/(DFLOAT(NSAMP))
      SSIGSU = SSIGSU/(DFLOAT(NSAMP))

      SMOMB = SMOMB/(DFLOAT(NSAMP))
      SMUSB = SMUSB/(DFLOAT(NSAMP))
      SSIGSB = SSIGSB/(DFLOAT(NSAMP))


      SMOMT = SMOMT/(DFLOAT(NSAMP))
      SMUST = SMUST/(DFLOAT(NSAMP))
      SSIGST = SSIGST/(DFLOAT(NSAMP))

          Y_EXC_M1 = Y_EXC_M1/(DFLOAT(NSAMP))
          Y_EXC_M2 = Y_EXC_M2/(DFLOAT(NSAMP))
          Y_EXC_M3 = Y_EXC_M3/(DFLOAT(NSAMP))

          Y_EXC_MU = Y_EXC_MU/(DFLOAT(NSAMP))
          Y_EXC_SIG = Y_EXC_SIG/(DFLOAT(NSAMP))


C
C     PRINT STATISTICS
C
          PRINT*,'EXAMPLE:'
          PRINT*
          PRINT*, 'UNIVARIATE:'
          PRINT*
          PRINT*, 'PF = ', PFU
          PRINT*, 'SPF_MU = ', SPFU_MU
          PRINT*, 'SPF_SIG = ', SPFU_SIG
          PRINT*
C
          PRINT*, 'BIVARIATE:'
          PRINT*
          PRINT*, 'PF = ', PFB
          PRINT*, 'SPF_MU = ', SPFB_MU
          PRINT*, 'SPF_SIG = ', SPFB_SIG
          PRINT*
C
          PRINT*, 'TRIVARIATE:'
          PRINT*
          PRINT*, 'PF = ', PFT
          PRINT*, 'SPF_MU = ', SPFT_MU
          PRINT*, 'SPF_SIG = ', SPFT_SIG
          PRINT*
C
```

```
C      PRINT*, 'SMOMU = '
C      WRITE (*,*) SMOMU
C      PRINT*, 'SMUSU = '
C      WRITE (*,*) SMUSU
C      PRINT*, 'SSIGSU = '
C      WRITE (*,*) SSIGSU
C
C      PRINT*, 'SMOMB = '
C      WRITE (*,*) SMOMB
C      PRINT*, 'SMUSB = '
C      WRITE (*,*) SMUSB
C      PRINT*, 'SSIGSB = '
C      WRITE (*,*) SSIGSB
C
C      PRINT*, 'SMOMT = '
C      WRITE (*,*) SMOMT
C      PRINT*, 'SMUST = '
C      WRITE (*,*) SMUST
C      PRINT*, 'SSIGST = '
C      WRITE (*,*) SSIGST
C
C      PRINT*, 'Y_EXC_M1  = '
C      WRITE (*,*) Y_EXC_M1
C      PRINT*, 'Y_EXC_M2  = '
C      WRITE (*,*) Y_EXC_M2
C      PRINT*, 'Y_EXC_M3  = '
C      WRITE (*,*) Y_EXC_M3
C
C      PRINT*, 'Y_EXC_MU  = '
C      WRITE (*,*) Y_EXC_MU
C      PRINT*, 'Y_EXC_SIG  = '
C      WRITE (*,*) Y_EXC_SIG



        PRINT*, 'DIRECT MCS:'
        PRINT*
        PRINT*, 'PFMCS = ', PFMCS
        PRINT*
C
        PRINT*,'U.dat, B.dat, T.dat, MCS.dat OPENED FOR SAMPLE VALUES'
        PRINT*, 'OF UNIVARIATE, BIVARIATE, TRIVARIATE, AND MCS SOLUTIONS'


      STOP
      END

C
C       ---------------------------------------------------------
        SUBROUTINE RESPON(X,Y)
C
C       CALCULATES Y FOR A SAMPLE OF X
C


        USE DFLIB
        USE ERF_INT
        USE UMACH_INT
        INCLUDE "link_fnl_shared.h"
      IMPLICIT REAL*8 (A-H,O-Z)
       REAL*8 X(200),Z(200),Z1(200), XMU(200),SIG(200), NMU(200),NSIG(200)
      REAL*8 Y
      INTEGER NYQ, N, NGAUSS_GQ, NBAS
        COMMON   /XR01/ XMU,SIG, NMU, NSIG, NYQ, N, NGAUSS_GQ,  NBAS, A_LAMBDA
     1  ,A_K
C

      DO I = 1,N
       Z1(I) = NMU(I) + NSIG(I)*X(I)

      ENDDO

      DO I = 1,4
       Z(I) = A_LAMBDA*( -log(0.5D0*(1-DERF(X(I)/DSQRT(2.0D0))) ) )
     1  **(1.0/A_K)
      ENDDO


      Y= 500.0-(Z(1)+Z(2))**3+Z(1)-Z(2)-Z(3)+Z(1)*Z(2)*Z(3)-Z(4)
c      Y= 500.0-(X(1)+X(2))**3+X(1)-X(2)-X(3)+X(1)*X(2)*X(3)-X(4)
c      Y = X(1) + X(2)**3  + (X(1)+X(2))**3+X(1)*X(2)*X(3)

C      Z(1) = NMU(1) + NSIG(1)*X(1)
C  Z(2) = NMU(2) + NSIG(2)*X(2)
C  Z(3) = NMU(3)+ NSIG(3)*X(3)
C
C
C  Y = (Z(1)**2 + Z(2)**2 + Z(3)**2 + Z(1)*Z(2)
C     1      + 2*Z(2)*Z(3) + 4*Z(3)*Z(1))**NYQ
CC
        RETURN
        END

C
C-----------------------------------------------------------
        SUBROUTINE SCOR(X,S_MU, S_SIG)
C
C       CALCULATES SCORE FUNCTION FOR A SAMPLE OF X
```

```
C
      IMPLICIT REAL*8 (A-H,O-Z)
        REAL*8 X(200),Z(200),XMU(200),SIG(200), NMU(200),NSIG(200)
      REAL*8 S_MU, S_SIG
      INTEGER NYQ, N, NGAUSS_GQ, NBAS
        COMMON   /XR01/ XMU,SIG, NMU, NSIG, NYQ, N, NGAUSS_GQ,  NBAS, A_LAMBDA
     1  ,A_K
C
C   DO I = 1,4
C   Z(I) = X(I)
C   ENDDO
       Z(1) = NMU(1) + NSIG(1)*X(1)
C   Z(2) = NMU(2) + NSIG(2)*X(2)
C   Z(3) = NMU(3) + NSIG(3)*X(3)
C
      S_MU =  1./NSIG(1) * ((Z(1)-NMU(1))/NSIG(1))
C     1       + 1/NSIG(2) * ((Z(2)-NMU(2))/NSIG(2))
C     2       + 1/NSIG(3) * ((Z(3)-NMU(3))/NSIG(3))
      S_SIG =  1./NSIG(1) * (((Z(1)-NMU(1))/NSIG(1))**2  - 1)
C     1        + 1/NSIG(2) * (((Z(2)-NMU(2))/NSIG(2))**2  - 1)
C     2        + 1/NSIG(3) * (((Z(3)-NMU(3))/NSIG(3))**2  - 1)
C

        RETURN
        END

C
C--------------------------------------------------------------
        SUBROUTINE SCOR2(X,S_MU, S_SIG)
C
C       CALCULATES SCORE FUNCTION FOR A SAMPLE OF X  FOR IID
C
      IMPLICIT REAL*8 (A-H,O-Z)
        REAL*8 X(200),Z(200), XMU(200),SIG(200), NMU(200),NSIG(200)
      REAL*8 S_MU, S_SIG
      INTEGER NYQ, N, NGAUSS_GQ, NBAS
        COMMON   /XR01/ XMU,SIG, NMU, NSIG, NYQ, N, NGAUSS_GQ,  NBAS, A_LAMBDA
     1  ,A_K

C


      S_MU =0.
      S_SIG =0.
      DO I =1,N
        Z(I) = NMU(I) + NSIG(I)*X(I)
        S_MU = S_MU + 1./NSIG(I) * ((Z(I)-NMU(I))/NSIG(I))
        S_SIG = S_SIG + 1./NSIG(I) * (((Z(I)-NMU(I))/NSIG(I))**2  - 1.0)
      ENDDO

        RETURN
        END

C
C--------------------------------------------------------------
        SUBROUTINE SCOR3_WEIBULL(UX,S_LAMBDA, S_K)
C
C       CALCULATES SCORE FUNCTION FOR A SAMPLE OF  U (need to transform to X)  FOR IID
C
        USE DFLIB
        USE ERF_INT
        USE UMACH_INT
        INCLUDE "link_fnl_shared.h"

      IMPLICIT REAL*8 (A-H,O-Z)
        REAL*8 UX(200),X(200),Z(200), XMU(200),SIG(200), NMU(200),NSIG(200)
      REAL*8 S_MU, S_SIG
      INTEGER NYQ, N, NGAUSS_GQ, NBAS
      COMMON /XR01/ XMU,SIG, NMU, NSIG, NYQ, N, NGAUSS_GQ, NBAS,A_LAMBDA
     1  ,A_K

C


      S_LAMBDA =0.
      S_K  =0.
      DO I = 1,N
        X(I) = A_LAMBDA*( -log(0.5D0*( 1-DERF(UX(I)/DSQRT(2.0D0)) ) ) )
     1  **(1.0/A_K)
      ENDDO

      DO I =1,N
C        S_MU = S_MU -1.0/NMU(I) + X(I)/NMU(I)**2.0
        S_LAMBDA = S_LAMBDA+ A_K/A_LAMBDA *
     1   ( (X(I)/A_LAMBDA)**A_K  - 1.0)
        S_K = S_K+ 1.0D0/A_K + (LOG(X(I)) - LOG(A_LAMBDA))*
     1   (1.0- (X(I)/A_LAMBDA)**A_K )

      ENDDO

        RETURN
        END

C
C--------------------------------------------------------------
        SUBROUTINE FX_FU_JU(UX, Y_FFJ)
C
C       CALCULATES SCORE FUNCTION FOR A SAMPLE OF X  FOR IID
C
        USE DFLIB
        USE ERF_INT
        USE UMACH_INT
        INCLUDE "link_fnl_shared.h"

      IMPLICIT REAL*8 (A-H,O-Z)
        REAL*8 UX(200),X(200), Z(200), XMU(200),SIG(200), NMU(200),NSIG(200)
      REAL*8 Y_FFJ, detJ
      INTEGER NYQ, N, NGAUSS_GQ, NBAS
        COMMON   /XR01/ XMU,SIG, NMU, NSIG, NYQ, N, NGAUSS_GQ,  NBAS, A_LAMBDA
     1  ,A_K

        PI = 4.*DATAN(1.0D0)
C

      DO I = 1,N
        X(I) = A_LAMBDA*( -log(0.5D0*(1-DERF(UX(I)/DSQRT(2.0D0))) ) )
     1  **(1.0/A_K)
      ENDDO
```

```
      detJ = A_LAMBDA**N /DSQRT(2.0*PI)**N
      DO I =1,N
        detJ = detJ * DEXP(-0.5*UX(I)**2)
     1   *(- log(0.5D0*(1-DERF(UX(I)/DSQRT(2.0D0)))) )**(1.0/A_K - 1.0)
     2   /(0.5D0*(1-DERF(UX(I)/DSQRT(2.0D0))))
      ENDDO

      Y_FFJ = detJ
      DO I = 1,N
        Y_FFJ = Y_FFJ* A_K/A_LAMBDA * (X(I)/A_LAMBDA)**(A_K - 1.0)*
     1    DEXP(- (X(I)/A_LAMBDA)**A_K )*DEXP(-A_LAMBDA*X(I))*
     1    DSQRT(2.0*PI)/DEXP(-0.5*UX(I)**2)
      ENDDO


        RETURN
        END



C
C-----------------------------------------------------------------
        SUBROUTINE PDF_IID(X, Y)
C
C       CALCULATES PDF VALUE FOR A SAMPLE OF X
C
      IMPLICIT REAL*8 (A-H,O-Z)
        REAL*8 X(200),Z(200),XMU(200),SIG(200), NMU(200),NSIG(200)
      REAL*8 S_MU, S_SIG
      INTEGER NYQ, N, NGAUSS_GQ, NBAS
        COMMON   /XR01/ XMU,SIG, NMU, NSIG, NYQ, N, NGAUSS_GQ,  NBAS, A_LAMBDA
     1  ,A_K

      Y = 1.
      DO I = 1,N
        Z(I) = NMU(I) + NSIG(I)*X(I)
        Y = Y * 1./(NSIG(I)*DSQRT(2*PI)) * EXP(-0.5*
     1       ((Z(I)-NMU(I))/NSIG(I))**2)

      ENDDO

        RETURN
        END




C
C       ------------------------------------------------------------
        SUBROUTINE BASIS(X,PSIF,N,NBAS)
C
C       DEFINES HERMITE POLYNOMIALS - NORMALIZED W.R.T. N(0,1) PDF
C       DO NOT INCLUDE THE 0-ORDER, BECAUSE THE 0-ORDER GIVES MEAN
C     ALREADY INCLUDED IN Y0
C
      IMPLICIT REAL*8 (A-H,O-Z)
        REAL*8 X(N),PSIF(N,NBAS)
C
        DO I = 1,N
C       IF (0 .LE. NBAS) PSIF(I,1) = 1.  ----------THIS IS WHY Y0 COME IN
        IF (1 .LE. NBAS) PSIF(I,1) = X(I)
        IF (2 .LE. NBAS) PSIF(I,2) = (X(I)**2 - 1.)/DSQRT(2.D0)
        IF (3 .LE. NBAS) PSIF(I,3) = (X(I)**3 - 3.*X(I))/
     &                               DSQRT(6.D0)
        IF (4 .LE. NBAS) PSIF(I,4) = (X(I)**4 - 6.*X(I)**2 + 3.)/
     &                               DSQRT(24.D0)
        IF (5 .LE. NBAS) PSIF(I,5) = (X(I)**5 - 10.*X(I)**3 + 15.*X(I))/
     &                               DSQRT(120.D0)
        IF (6 .LE. NBAS) PSIF(I,6) = (X(I)**6 - 15.*X(I)**4 + 45.*X(I)**2
     &                              -15.)/DSQRT(720.D0)
        IF (7 .LE. NBAS) PSIF(I,7) = (X(I)**7 - 21.*X(I)**5 + 105.*X(I)**3
     &                              -105.*X(I))/DSQRT(5040.D0)
        IF (8 .LE. NBAS) PSIF(I,8) = (X(I)**8 - 28.*X(I)**6 + 210.*X(I)**4
     &                              -420.*X(I)**2 + 105.)/DSQRT(40320.D0)
        IF (9 .LE. NBAS) PSIF(I,9) = (X(I)**9 - 36.*X(I)**7 + 378.*X(I)**5
     &               -1260.*X(I)**3 + 945.*X(I))/DSQRT(362880.D0)
        ENDDO
C
        RETURN
        END
```