

```

=====
Sensitivity analysis for
EXAMPLE 4: a function of non-Gaussian variables (Lognormally distributed)
METHOD: PDD-SPA option 1
MAIN CODE

N = 6
Y = X1 + 2*X2 + 2*X_3 + X4 - 5*X5 - 5*X6

X(I) are lognormally distributed, mu1 ... mu4 = 120, mu5 = 50, mu6 = 40, COV = c \in [0.1, 0.7] STIELTJES PROCEDURE
GAUSS QUADRATURE RULE BY STIELTJES PROCEDURE

----- command line parameters to run the code -----
PROGRAM.EXE NGAUSS_SPA
*****
Copyright (c) 2010-2014 [Xuchun Ren], All Rights Reserved.

IMPLICIT REAL*8 (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
INTEGEB 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
INTEGEB IITER, IITER1, N_PREFEA, N_FEA, IITER_TOTAL, I_SQP
COMMON /XR05/ IITER, IITER1, N_PREFEA, N_FEA, IITER_TOTAL, I_SQP

REAL*8 PF, PF_MCS, PF_SEN(10,10), PF_SEN_MCS(10,10), TS_OUT,
1 Y0, YU(10,10), WU(10,10)

INTEGEB NBAS_SPA, NGAUSS_SPA ,status

COMMON /SPA01/ Y0, YU,WU, NBAS_SPA, NGAUSS_SPA
CHARACTER(80) MSG,MSG1
CHARACTER(320) buf
INTEGEB IW1
COMMON /XR07/ IW1
n1 = 1
n2 = 2
n3 = 3
n4 = 4
n5 = 5
n6 = 6
n7 = 7
n8 = 8

IW1 = 0

MSG = 'PROGRAM NGAUSS_SPA'

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

NBAS_SPA = 1
NGAUSS_SPA = 3

c0 = 1.0

IFLAG_AIJK = 0
N = 6
IMCS = 0
NVAR = 1

X0(1:4) = 120.0
X0(5) = 50.0
X0(6) = 40.0

NSAMP = 10**6
IID = 0

DO I = 1, N
XMU_IN(I) = X0(I)
ENDDO

NBAS_R = 1

NBAS = 1
NBAS_SM = 1
NGAUSS = 2
NGAUSS_SM = 2
NGAUSS_GQ = 2

-- modify main
dcov = 0.02
ncov = (0.7-0.1)/dcov
ncum = 8
ng_cum = 8
ng_cum_sen = 10
ntheta = 2

if (mod(ncum,2) .ne. 0) then
print*, 'ncum should be even!'
stop
endif

do i = 0, ncov+1
cov = 0.1 + i*dcov
CALL PDD_SEN_PF_SPA (X0, XMU_IN, cov, NSAMP,NBAS,
1 NBAS_SM, NGAUSS,
1 NGAUSS_SM, IID, NGAUSS_GQ, 1, IMCS, TS_OUT, PF,
2 PF_SEN, PF_MCS, PF_SEN_MCS,
3 IFLAG_AIJK, NVAR, ncum, ng_cum, ng_cum_sen, ntheta)

enddo

PRINT*, "PF = ", PF
PRINT*, "PF_SEN = ", PF_SEN(1:3,1:2)

PRINT*, "PF_MCS = ", PF_MCS
PRINT*, "PF_SEN_MCS = ", PF_SEN_MCS(1:3,1:2)

PRINT*, "TS_OUT = ", TS_OUT

END

```