```
*==> global.f <==
****************************************************************************
*                                                                          *
*  FILES                                                                   *
*                                                                          *
*  evol.rg   :   Output numbers from routine out.f                         *
*  susyin    :   I.C. at Mgut                                              *
*  log.true  :   Useful quantities (logs etc) at True min.                 *
*  true_min  :   True minimum configuration                                *
*  sigma     :   Tr(s) , Tr(|s|) one loop corrections to m1, m2            *
*                                                                          *
*                                                                          *
*  COMMON /RENTR/    :  Renormalization scale for TRUE min. (fixed)        *
*                       trn,qsl is the scale which appears explicitly      *
*                       in logarithms. This common is initialized          *
*                       in block data subpr.                               *
*  COMMON /SOFCOM/   :  Soft parameters and "corrected" vevs at Qrn        *
*  COMMON /NIT/      :  Communicate I.C. to evalsoft routine.              *
*  COMMON /QUEUE/    : Pass the value of logical lpr into function ftr2    *
*                      where the effective potential is calculated.        *
*                      (lpr controls printing facilities)                  *
*  COMMON /TRMIN/    : True minimum                                        *
*                                                                          *
*                                                                          *
*  idata    :   # of data points                                           *
*  ao       :   A_0                                                        *
*  amoo     :   M_0                                                        *
*  am12     :   M_1/2                                                      *
*  efapto   :   tanb                                                       *
*  ttop     :   Pole mass of top quark                                     *
*  reject   :   Rejecting unphysical sitations (Logical)                   *
*                                                                          *
*                                                                          *
*          1   mu > 0                                                      *
*        /                                                                 *
*   j ---                                                                  *
*        \                                                                 *
*          2   mu < 0                                                      *
*                                                                          *
*            1   No thres.                                                 *
*          /                                                               *
*   nai ---                                                                *
*          \                                                               *
*            2   Thresholds                                                *
*                                                                          *
*   MODULES                                                                *
*                                                                          *
*  initcnd  -->  Prints I.C. at Mgut in all files                          *
*  signmu   -->  Prints headers for mu>0 or mu<0                           *
*  xbrs     -->  Prints the renormal. scale for TRUE min.                  *
*                Mgut is computed inside evalsoft so we have to call       *
*                xbrs after calling evalsoft.                              *
****************************************************************************
      PROGRAM global
      IMPLICIT DOUBLE PRECISION (a-h,o-z)
      PARAMETER(idata=6,NU=39)
      DOUBLE PRECISION xcom(NU),vev(2),ftr2
      DOUBLE PRECISION hyps(NU)
      LOGICAL reject,lpr
      COMMON /SOFCOM/ xcom
      COMMON /RENTR/ trn,qsl
      COMMON /NIT/ ao,amoo,am12
```



```
      COMMON /SCALE/ amgut,susy,amz,amw
      COMMON /TRMIN/ yb,pb(2)
      COMMON /QUEUE/ lpr
      EXTERNAL ftr2
      DATA reject /.false./, naim /1/   !  naim always 1 due to PZZ corr.
      DATA nai,ttop /1, 174.d0/
      DATA kun,temp,it,stval /97, 0.d0, 80, 100.d0/
C-------------------------------------------------
C
C   Evolution output file
C
C     open(57,file='evol.rg')
C-------------------------------------------------
C      + is for mu>0  and - is for mu<0
C         write headings for ast.out
C-------------------------------------------------
C     OPEN(14,FILE='TAU81')
C     OPEN(21,FILE='RADY81')
C     OPEN(13,FILE='BEMU81')
C     OPEN(17,FILE='LOG81')
C     open(76,file='EXP')
C     OPEN(11,FILE='susyout')
C     OPEN(25,FILE='ast+.out')
C     OPEN(23,FILE='LLT+')
C     OPEN(26,FILE='ast-.out')
C     OPEN(27,FILE='LLT-')
C     OPEN(45,FILE='sin^2+')
C     OPEN(46,FILE='sin^2-')
C     OPEN(47,FILE='bounds+')
C     OPEN(48,FILE='bounds-')
C
      open(20,file='SUSY')    !  Write only in the 3-rd loop
      open(22,file='SIN')     !              >>
      open(24,file='susyin')
      open(67,file='data')
      open(98,file='log_true')
      open(97,file='true_min')
      open(96,file='pass_tests')   !  Write for all loops
      open(95,file='sigma')
      open(94,file='num.irr')
      open(93,file='log.irr')
      open(92,file='caf.out')
      open(91,file='chf.out')
C------------------------------------------
C   Gnuplot Files
C------------------------------------------
      open(90,file='ratio')
      open(89,file='sing')
      open(88,file='pertur')
      open(87,file='roundoff')
      open(86,file='avlog')
C
C     write(25,*)'#Ao   Mo      M1/2    as(M_z)full  mu>0'
C     write(26,*)'#Ao   Mo      M1/2    as(M_z)full  mu<0'
C     write(23,*)'#Ao   Mo      M1/2    LLT       mu>0'
C     write(27,*)'#Ao   Mo      M1/2    LLT       mu<0'
C     write(45,*)'#Ao   Mo      M1/2    sin^2(MSbar)       mu>0'
C     write(46,*)'#Ao   Mo      M1/2    sin^2(MSbar)       mu<0'
C
C--------------------------------
C   Header in SUSY file
C--------------------------------
```

```
      call hdiu20(NAI,NAIM)
C-----------------------------------------------------
C   Renormalization scale (explicit) is initialized
C   in block data subpr
C-----------------------------------------------------
      do   kdata = 1,idata
          read(24,*)ao,amoo,am12,efapto
          call initcnd(ao,amoo,am12,efapto)
          do  j = 1,2
             call signmu(j)
C--------------------------------------------------------
             call string(j,efapto,TTOP,hyps,nai,reject)
C--------------------------------------------------------
             call evalsoft(j,efapto,TTOP,xcom,nai,reject)
             if(reject)then
               reject=.false.
             else
               trn=2.d0*dlog(amgut/qsl)
               call xbrs(97,trn)
C----------------------------------------------
C   Finding TRUE minimum value from iterations
C   Use automatic format for comparison reasons.
C----------------------------------------------
               vev(1) = xcom(32)
               vev(2) = xcom(33)
               write(97,'(//,5X,"From iterations True min. at :")')
               write(97,*)' v1 =',xcom(32)
               write(97,*)' v2 =',xcom(33)
               write(97,*)' Veff =',ftr1(vev)
               write(97,'(///)')
C--------------------------------------------------
C   Finding TRUE minimum value with numerical tools
C--------------------------------------------------
               lpr=.false.   !  *** No printing from xamebsa
               call xamebsa(kun,temp,it,ftr2,stval,pb,yb)
               lpr=.true.    !  *** Log check
               zs=ftr2(pb)
C              temptr=1.d6
C              iiter=16
C              call xamebsa(temptr,iiter)
             endif
          enddo
          call sepmu
      enddo
      END

*==> hdiu20.f <==
**********************************************************************
*    Calculates both with thresholds and without them. The point is  *
*    that in the function F(I,II,J,..   ) :                           *
*    I=1, II=1  NO THRESHOLDS AT ALL                                  *
*    I=2, II=1  NO THRESHOLDS AT GAUGE COUPLINGS,YES ELSEWHERE        *
*    I=2, II=2  THRESHOLDS TO EVERYTHING                              *
*    I=1, II=2  THRESHOLDS AT GAUGE + YUKAWAS                         *
*    ONLY THE ABOVE COMBINATIONS ARE ALLOWED FOR THE SWITCHES         *
*                                                                     *
*    In the absence of threshold effects, in the RGEs, the "B,mu"     *
*    decouple from the rest of the RGEs but not with thresholds       *
*                                                                     *
*    Allowed combinations are:                                        *
*    NAI=1, NAIM=1 ( No thresholds )                                  *
*    NAI=2, NAIM=1 (Thresholds elsewhere except gauge couplings )     *
```



```
*    NAI=2, NAIM=2 (Thresholds to everything)                         *
*    NAI=1, NAIM=2 (Thresholds only in couplings)                     *
*    NAIH=1, NAIH=2 (No high energy thresholds, Yes high energy       *
*    thresholds, LOOK at FUNCTION NA(J) for more details              *
*    NRO=1 Yes Non Renormalizable Operators                           *
*    NRO=0 No Non Renormalizable Operators                            *
**********************************************************************
      SUBROUTINE hdiu20(NAI,NAIM)
C
      WRITE(20,'(/,"****  NAI = ",I2,"   NAIM = ",I2)') NAI,NAIM
      if(NAI.EQ.1 .AND. NAIM.EQ.1)then
         WRITE(20,'("  ***   NO THRESHOLDS APPLY TO RGEs  ***")')
      elseif(NAI.EQ.1 .AND. NAIM.EQ.2)then
         WRITE(20,'("  *** THRESHOLDS ONLY IN RGEs OF COUPLINGS",
     # " GAUGE + YUKAWAS ***  ")')
      elseif(NAI.EQ.2 .AND. NAIM.EQ.1)then
         WRITE(20,'("  *** THRESHOLDS ONLY IN RGEs OF  MASSES",
     # " + YUKAWAS ***  ")')
      elseif(NAI.EQ.2 .AND. NAIM.EQ.2)then
         WRITE(20,'("  *** THRESHOLDS IN RGEs OF EVERYTHING  ***")')
      endif
C
      return
      END

*==> initcnd.f <==
      SUBROUTINE initcnd(ao,amoo,am12,efapto)
      IMPLICIT DOUBLE PRECISION (a-h,o-z)
      PARAMETER(ID1=14)    !  Fix it as you like
      DIMENSION iu1(ID1)
      DATA (iu1(k),k=1,ID1)/22, 86, 87, 88, 89, 90, 91, 92, 93,
     #                      94, 95, 96, 97, 98/
CC
      do   ir=1,ID1
          write(iu1(ir),'(34("-"),/,"A0 = ",f10.4,/,
     # "M0 = ",f10.4,/,"M1/2 = ",f10.4,/,
     # "tanb = ",f10.4,/,34("-"),//)')ao,amoo,am12,efapto
      enddo
      return
      END

*==> signmu.f <==
      SUBROUTINE signmu(i)
      PARAMETER(ID2=15)    !  Fix it as you like
      DIMENSION mu2(ID2)
      DATA (mu2(n),n=1,ID2)/22, 86, 87, 88, 89, 90, 91, 92, 93,
     #                      94, 95, 96, 97, 98, 20/
CC
      do   k=1,ID2
      if(i.eq.1)write(mu2(k),'(//,24X,"***  mu > 0  *** ",//)')
      if(i.eq.2)write(mu2(k),'(//,24X,"***  mu < 0  *** ",//)')
      enddo
CC
      return
      END

*==> sepmu.f <==
      SUBROUTINE sepmu
      PARAMETER(ID2=15)   !  Fix it as you like
      DIMENSION mu2(ID2)
      DATA (mu2(n),n=1,ID2)/22, 86, 87, 88, 89, 90, 91, 92, 93,
     #                      94, 95, 96, 97, 98, 20/
```

```
CC
        do   jr=1,ID2
            write(mu2(jr),'(/,34(">",X),/)')
        enddo
       return
       END

*==> xbrs.f <==
       SUBROUTINE xbrs(iu,tbr)
       IMPLICIT DOUBLE PRECISION (a-h,o-z)
       COMMON /SCALE/ amgut,susy,amz,amw
       qbr=amgut*dexp(-tbr/2.d0)
       write(iu,'(/,"Qsl = ",f10.4,//)') qbr
       return
       END

*==> bld_sector.f <==
       BLOCK DATA sector
       IMPLICIT DOUBLE PRECISION (a-h,o-z)
       DIMENSION icut(44),izip(48)
       COMMON /KOFTIS/ icut
       COMMON /ZIPER/ izip
       COMMON /RENTR/ t,q
       COMMON /XMARG/ rmar1,rmar2
       COMMON/FUTURE/smvector,smhcolor,smsigma,smhfprime
C
C  Top-stop contribution
C
C      DATA (icut(i),i=1,44)
C     #/0,0,0, 0,0,1, 0,0,0, 0,0, 2*0, 4*0, 0,0,0,
C     # 2*0,2*0,2*0, 2*0,2*0,2*1, 2*0,2*0,2*0, 2*0,2*0,2*0/
C      DATA (izip(j),j=1,48)
C     #/0,0,0, 0,0,1, 0,0,0, 0,0, 4*0, 4*0, 0,0,0,
C     # 2*0,2*0,2*0, 2*0,2*0,2*1, 2*0,2*0,2*0, 8*0/
C
C  Full sector contribution
C
       DATA (icut(i),i=1,44)
      #/1,1,1, 1,1,1, 1,1,1, 1,1, 2*1, 4*1, 1,1,1,
      # 2*1,2*1,2*1, 2*1,2*1,2*1, 2*1,2*1,2*1, 2*1,2*1,2*1/
       DATA (izip(j),j=1,48)
      #/1,1,1, 1,1,1, 1,1,1, 1,1, 4*1, 4*1, 1,1,1,
      # 2*1,2*1,2*1, 2*1,2*1,2*1, 2*1,2*1,2*1, 8*1/
C
       DATA smvector,smhcolor,smsigma,smhfprime
      #    /1.d0, 3.d0, 3.d0, 5.d0/
       DATA t,q /50.d0,200.d0/
       DATA rmar1,rmar2 /1.d6, 1.d6/
       END
```