```
*==> chino.f <==
*********************************************
*  Sigma corrections due to charginos       *
*                                           *
*                                           *
*                 1      db[k]              *
*  d[i]b[k] = ------ * ------     (k=0,1)   *
*               x[i]     dx[i]    (i=1,2)   *
*                                           *
*  Details see p.n. TRUE min.               *
*                                           *
*********************************************
      SUBROUTINE chino(rmu,rm2,g2,x1,x2,ts,sig1c1,sig1c2
     #,sig2c1,sig2c2)
      IMPLICIT DOUBLE PRECISION (a-h,o-z)
      COMMON /wur/ wur,pi
      COMMON /ginrp/ i, k, ilim, klim
C
       b1=rmu**2 + rm2**2 + g2**2*(x1**2+x2**2)
       rb0=rmu*rm2 - x1*x2*g2**2
       b0=rb0**2
C
       call solve2(-b1,b0,cs1,cs2)
       sn=dsign(1.d0,cs1-cs2)
C
       d1b1=2.d0*g2**2
       d2b1=2.d0*g2**2
       d1b0=-2.d0*rb0*g2**2*x2/x1
       d2b0=-2.d0*rb0*g2**2*x1/x2
C
       pr1=pythag(rmu+rm2,g2*(x1-x2))
       pr2=pythag(rmu-rm2,g2*(x1+x2))
       rood=pr1*pr2
C
C
       gn1=b1*d1b1 - 2.d0*d1b0
       gn2=b1*d2b1 - 2.d0*d2b0
C
       d1cs1=(d1b1 + gn1/(sn*rood))/2.d0
       d2cs1=(d2b1 + gn2/(sn*rood))/2.d0
C
       d1cs2=(d1b1 - gn1/(sn*rood))/2.d0
       d2cs2=(d2b1 - gn2/(sn*rood))/2.d0
C
       sig1c1=-d1cs1*flog(cs1,ts)/(16.d0*pi*pi)
       sig2c1=-d2cs1*flog(cs1,ts)/(16.d0*pi*pi)
C
       sig1c2=-d1cs2*flog(cs2,ts)/(16.d0*pi*pi)
       sig2c2=-d2cs2*flog(cs2,ts)/(16.d0*pi*pi)
C
      if(i.eq.ilim .and. k.eq.klim)then
      write(95,*)'Chargino corrections'
      write(95,*)'sig1c1 =',sig1c1,' sig1c2 =',sig1c2
      write(95,*)'sig2c1 =',sig2c1,' sig2c2 =',sig2c2
      endif
C
      return
      END

*==> cnf.f <==
      SUBROUTINE cnf(c,dc,umm1,umm2,dumm)
      IMPLICIT DOUBLE PRECISION (a-h,o-z)
```



```
      DOUBLE PRECISION c(3),dc(2,3),dumm(2,2)
C
      b1=-c(1)-c(3)
      b0=c(1)*c(3) - c(2)*c(2)
      xd=c(1)-c(3)
      xx=2.d0*c(2)
      call solve2(b1,b0,umm1,umm2)
      sn=dsign(1.d0,umm1-umm2)
      rood=pythag(xd,xx)
C
      p1p=dc(1,1)+dc(1,3)
      p1m=dc(1,1)-dc(1,3)
C
      p2p=dc(2,1)+dc(2,3)
      p2m=dc(2,1)-dc(2,3)
C
      ar1=xd*p1m + 2.d0*xx*dc(1,2)
      ar2=xd*p2m + 2.d0*xx*dc(2,2)
C
      dumm(1,1)=( p1p + ar1/(sn*rood) )/2.d0
      dumm(2,1)=( p2p + ar2/(sn*rood) )/2.d0
      dumm(1,2)=( p1p - ar1/(sn*rood) )/2.d0
      dumm(2,2)=( p2p - ar2/(sn*rood) )/2.d0
C
      return
      END

*==> dwq.f <==
*********************************************
*                                           *
*   Sigma corrections due to DOWN quarks    *
*                                           *
*   y    :  Yukawa                          *
*   x    :  Vev                             *
*   q    :  Renorm.  scale                  *
*                                           *
*   y   |x                                  *
*   ----|----------------                   *
*   yd1 |x1                                 *
*   yd2 |x1    DOWN quarks                  *
*   yd3 |x1                                 *
*                                           *
*********************************************
      SUBROUTINE dwq(y,x,ts,s1,s2)
      IMPLICIT DOUBLE PRECISION (a-h,o-z)
      COMMON /wur/ wur,pi
      COMMON /ginrp/ i, k, ilim, klim
C
      qd2=(y*x)**2
      s1=-3.d0*y*y*flog(qd2,ts)/(8.d0*pi*pi)
      s2=0.d0
C
      if(i.eq.ilim .and. k.eq.klim)then
      write(95,*)'Down quark corrections'
      write(95,*)'s1 =',s1
      write(95,*)'s2 =',s2
      endif
C
      return
      END

*==> flog.f <==
```

```
      DOUBLE PRECISION FUNCTION flog(s,ts)
      IMPLICIT DOUBLE PRECISION (a-h,o-z)
      COMMON /SCALE/ amgut,susy,amz,amw
C
C   Logarithmic part of 1-loop corrections to Veff.
C   Some squares of field dependent masses (sqe) can be negative or
C   zero. In the latter case the contribution from u( ) is zero.
C
      if(s.eq.0.d0)then
      flog = 0.d0
      return
      endif
C
C     if(s.lt.0.d0)print*,'Negative sqe   ',s
C
      a1=dsqrt(dabs(s))
      a2=ts/2.d0 + dlog(a1) - dlog(amgut)
      flog = s*(2.d0*a2-1.d0)
C
C
      return
      END

*==> gabo.f <==
**********************************************
*                                            *
*   Sigma corrections due to gauge bosons    *
*                                            *
*          2                W                *
*         /                                  *
*    c --                                    *
*         \                                  *
*          1                Z                *
*                                            *
*                                            *
*          g2*g2            W                *
*         /                                  *
*    cf --                                   *
*         \                                  *
*          g*g + g2*g2      Z                *
*                                            *
*                                            *
**********************************************
      SUBROUTINE gabo(d,cf,x1,x2,ts,s1,s2)
      IMPLICIT DOUBLE PRECISION (a-h,o-z)
      COMMON /wur/ wur,pi
      COMMON /ginrp/ i, k, ilim, klim
C
      gb2=cf*(x1**2+x2**2)/2.d0
      s1=3.d0*d*cf*flog(gb2,ts)/(64.d0*pi*pi)
      s2=s1
C
      if(i.eq.ilim .and. k.eq.klim)then
      write(95,*)'Gauge boson corrections'
      write(95,*)'s1 = ',s1
      write(95,*)'s2 = ',s2
      endif
C
      return
      END

*==> higa.f <==
```



```
*****************************************
*                                       *
*   Sigma corrections due to a higgs    *
*                                       *
*   sm1  | sm2  | sm3                   *
*   -----|------|------                 *
*   sqm1 |sqm2  | sqm3                  *
*                                       *
*****************************************
      SUBROUTINE higa(sm1,sm2,sm3,g,g2,x1,x2,ts,
     #   s1ha1,s1ha2,s2ha1,s2ha2)
      IMPLICIT DOUBLE PRECISION (a-h,o-z)
      DOUBLE PRECISION t(3),dt(2,3),dha(2,2)
      COMMON /wur/ wur,pi
      COMMON /ginrp/ i, k, ilim, klim
C
      t(1)=sm1 + (g*g + g2*g2)*dfx(x1,x2)/4.d0 + g2*g2*x2*x2/2.d0
      t(2)=-sm3 + g2*g2*x1*x2/2.d0
      t(3)=sm2 + (g*g + g2*g2)*dfx(x2,x1)/4.d0 + g2*g2*x1*x1/2.d0
C
C
      dt(1,1)=(g*g + g2*g2)/2.d0
      dt(1,2)=g2*g2*x2/(2*x1)
      dt(1,3)=-dfx(g,g2)/2.d0
C
      dt(2,1)=-dfx(g,g2)/2.d0
      dt(2,2)=g2*g2*x1/(2*x2)
      dt(2,3)=(g*g + g2*g2)/2.d0
C
      call cnf(t,dt,ha1,ha2,dha)
C
      s1ha1=dha(1,1)*flog(ha1,ts)/(32.d0*pi*pi)
      s2ha1=dha(2,1)*flog(ha1,ts)/(32.d0*pi*pi)
      s1ha2=dha(1,2)*flog(ha2,ts)/(32.d0*pi*pi)
      s2ha2=dha(2,2)*flog(ha2,ts)/(32.d0*pi*pi)
C
      if(i.eq.ilim .and. k.eq.klim)then
      if(ha1.lt.0.d0 .or. ha2.lt.0.d0)
     # write(96,*)'some a hig^2 < 0'
      write(95,*)'Higgs A corrections '
      write(95,*)'s1ha1 = ',s1ha1,'  s1ha2 = ',s1ha2
      write(95,*)'s2ha1 = ',s2ha1,'  s2ha2 = ',s2ha2
      endif
C
      return
      END

*==> higb.f <==
*****************************************
*                                       *
*   Sigma corrections due to b higgs    *
*                                       *
*   sm1  | sm2  | sm3                   *
*   -----|------|------                 *
*   sqm1 |sqm2  | sqm3                  *
*                                       *
*****************************************
      SUBROUTINE higb(sm1,sm2,sm3,g,g2,x1,x2,ts,
     #   s1hb1,s1hb2,s2hb1,s2hb2)
      IMPLICIT DOUBLE PRECISION (a-h,o-z)
      DOUBLE PRECISION t(3),dt(2,3),dhb(2,2)
      COMMON /wur/ wur,pi
```

```
      COMMON /ginrp/ i, k, ilim, klim
C
      sco=g*g + g2*g2
      t(1)=sm1 + sco*dfx( dsqrt(3.d0)*x1 , x2)/4.d0
      t(2)=sm3 - sco*x1*x2/2.d0
      t(3)=sm2 + sco*dfx( dsqrt(3.d0)*x2 , x1)/4.d0
C
C
      dt(1,1)=3.d0*sco/2
      dt(1,2)=-sco*x2/(2.d0*x1)
      dt(1,3)=-sco/2.d0
C
      dt(2,1)=-sco/2.d0
      dt(2,2)=-sco*x1/(2.d0*x2)
      dt(2,3)=3.d0*sco/2
C
      call cnf(t,dt,hb1,hb2,dhb)
C
      s1hb1=dhb(1,1)*flog(hb1,ts)/(64.d0*pi*pi)
      s2hb1=dhb(2,1)*flog(hb1,ts)/(64.d0*pi*pi)
      s1hb2=dhb(1,2)*flog(hb2,ts)/(64.d0*pi*pi)
      s2hb2=dhb(2,2)*flog(hb2,ts)/(64.d0*pi*pi)
C
      if(i.eq.ilim .and. k.eq.klim)then
      if(hb1.lt.0.d0 .or. hb2.lt.0.d0)
     #  write(96,*)'some b hig^2 < 0'
      write(95,*)'Higgs B corrections'
      write(95,*)'s1hb1 = ',s1hb1,'  s1hb2 = ',s1hb2
      write(95,*)'s2hb1 = ',s2hb1,'  s2hb2 = ',s2hb2
      endif
C
      return
      END

*==> higc.f <==
******************************************
*                                        *
*   Sigma corrections due to c higgs     *
*                                        *
*   sm1  | sm2  | sm3                    *
*   -----|------|------                  *
*   sqm1 |sqm2  | sqm3                   *
*                                        *
******************************************
      SUBROUTINE higc(sm1,sm2,sm3,g,g2,x1,x2,ts,
     #   s1hc1,s1hc2,s2hc1,s2hc2)
      IMPLICIT DOUBLE PRECISION (a-h,o-z)
      DOUBLE PRECISION t(3),dt(2,3),dhc(2,2)
      COMMON /wur/ wur,pi
      COMMON /ginrp/ i, k, ilim, klim
C
      sco=g*g + g2*g2
      t(1)=sm1 + sco*dfx(x1,x2)/4.d0
      t(2)=-sm3
      t(3)=sm2 + sco*dfx(x2,x1)/4.d0
C
C
      dt(1,1)=sco/2.d0
      dt(1,2)=0.d0
      dt(1,3)=-sco/2.d0
C
      dt(2,1)=-sco/2.d0
```



```
      dt(2,2)=0.d0
      dt(2,3)=sco/2.d0
C
      call cnf(t,dt,hc1,hc2,dhc)
C
      s1hc1=dhc(1,1)*flog(hc1,ts)/(64.d0*pi*pi)
      s2hc1=dhc(2,1)*flog(hc1,ts)/(64.d0*pi*pi)
      s1hc2=dhc(1,2)*flog(hc2,ts)/(64.d0*pi*pi)
      s2hc2=dhc(2,2)*flog(hc2,ts)/(64.d0*pi*pi)
C
      if(i.eq.ilim .and. k.eq.klim)then
      if(hc1.lt.0.d0 .or. hc2.lt.0.d0)
     # write(96,*)'some c hig^2 < 0'
      write(95,*)'Higgs C corrections'
      write(95,*)'s1hc1 = ',s1hc1,'  s1hc2 = ',s1hc2
      write(95,*)'s2hc1 = ',s2hc1,'  s2hc2 = ',s2hc2
      write(95,*)
      write(95,*)
      endif
C
      return
      END

*==> lept.f <==
*********************************************
*                                           *
*   Sigma corrections due to leptons        *
*                                           *
*   y    :  Yukawa                          *
*   x    :  Vev                             *
*   q    :  Renorm.  scale                  *
*                                           *
*   y   |x                                  *
*   ----|----------------                   *
*   yl1 |x1                                 *
*   yl2 |x1    Leptons                      *
*   yl3 |x1                                 *
*                                           *
*********************************************
      SUBROUTINE lept(y,x,ts,s1,s2)
      IMPLICIT DOUBLE PRECISION (a-h,o-z)
      COMMON /wur/ wur,pi
      COMMON /ginrp/ i, k, ilim, klim
C
      fl2=(y*x)**2
      s1=-y*y*flog(fl2,ts)/(8.d0*pi*pi)
      s2=0.d0
C
      if(i.eq.ilim .and. k.eq.klim)then
      write(95,*)'Lepton corrections'
      write(95,*)'s1 = ',s1
      write(95,*)'s2 = ',s2
      endif
C
      return
      END

*==> neutr.f <==
**********************************************
*  Sigma corrections due to neutralinos      *
*                                            *
*                                            *
```

```
*              1      da[k]                    *
*  dia[k] = ------ * ------     (k=0,...,4)    *
*             x1      dx1                      *
*                                              *
*              1      da[k]                    *
*  diia[k] = ------ * ------    (k=0,...,4)    *
*             x2      dx2                      *
*                                              *
*  Details see p.n. TRUE min.                  *
*                                              *
************************************************
      SUBROUTINE neutr(g,g2,rmu,rm1,rm2,x1,x2,ts,rs1,rs2)
      IMPLICIT DOUBLE PRECISION (a-h,o-z)
      PARAMETER (n=4,np=4)
      DOUBLE PRECISION tn(4,4),dm(4),e(4),z(4,4)
      DOUBLE PRECISION rs1(4),rs2(4)
      COMMON /wur/ wur,pi
      COMMON /ginrp/ iaa, kiter, ilim, klim
C
      tn(1,1)=rm1
      tn(1,2)=0.d0
      tn(1,3)=g*x1/dsqrt(2.d0)
      tn(1,4)=-g*x2/dsqrt(2.d0)
      tn(2,2)=rm2
      tn(2,3)=-g2*x1/dsqrt(2.d0)
      tn(2,4)=g2*x2/dsqrt(2.d0)
      tn(3,3)=0.d0
      tn(3,4)=-rmu
      tn(4,4)=0.d0
        do  i1=2,4
           do   i2=1,i1-1
               tn(i1,i2)=tn(i2,i1)  !  Symmetric matrix
           enddo
        enddo
C
        call tred2(tn,n,np,dm,e)
        call tqli(dm,e,n,np,z)
C
      sga=rm1+rm2
      sco=g*g+g2*g2
      smix=rm1*g2*g2+rm2*g*g
      pg=x1*x1+x2*x2
      a1=-sga
      a2=-rmu*rmu + rm1*rm2 - sco*pg/2.d0
      a3=sga*rmu*rmu + smix*pg/2.d0 - rmu*sco*x1*x2
      a4=-rm1*rm2*rmu*rmu + rmu*smix*x1*x2
C
      dia1=0.d0
      dia2=-sco
      dia3=smix - rmu*sco*x2/x1
      dia4=rmu*smix*x2/x1
C
      diia1=0.d0
      diia2=-sco
      diia3=smix - rmu*sco*x1/x2
      diia4=rmu*smix*x1/x2
C
         do  k=1,4
      um1=dia1*dm(k)**3+dia2*dm(k)**2+dia3*dm(k)+dia4
      um2=diia1*dm(k)**3+diia2*dm(k)**2+diia3*dm(k)+diia4
      dnm=4.d0*dm(k)**3+3.d0*a1*dm(k)**2+2.d0*a2*dm(k)+a3
C
```



```
      d1sne=-2.d0*dm(k)*um1/dnm
      d2sne=-2.d0*dm(k)*um2/dnm
      sqr=dm(k)**2
C
      rs1(k)=-d1sne*flog(sqr,ts) / (32.d0*pi*pi)
      rs2(k)=-d2sne*flog(sqr,ts) / (32.d0*pi*pi)
C
C
          enddo
C
      if(iaa.eq.ilim .and. kiter.eq.klim)then
      write(95,*)'Neutralino corrections'
      write(95,*)'s1 =',rs1(1),rs1(2),rs1(3),rs1(4)
      write(95,*)'s2 =',rs2(1),rs2(2),rs2(3),rs2(4)
      endif
C
      return
      END

*==> slept.f <==
*********************************************
*                                           *
*   Sigma corrections due to sleptons       *
*                                           *
*   y           Yukawa                      *
*   a           Trilinear                   *
*   q           Renorm.  scale              *
*                                           *
*                                           *
*   sm2    |  smc2    |  y   |  a           *
*   ------|---------|------|------          *
*   sqml1  | sqmec1   | yl1  | al1          *
*   sqml2  | sqmec2   | yl2  | al2          *
*   sqml3  | sqmec3   | yl3  | al3          *
*                                           *
*********************************************
      SUBROUTINE slept(sm2,smc2,y,a,rmu,g,g2,x1,x2,ts,
     #   s1sl1,s1sl2,s2sl1,s2sl2)
      IMPLICIT DOUBLE PRECISION (a-h,o-z)
      DOUBLE PRECISION t(3),dt(2,3),dsl(2,2)
      COMMON /wur/ wur,pi
      COMMON /ginrp/ i, k, ilim, klim
C
      t(1)=smc2 + (y*x1)**2 + g*g*dfx(x2,x1)/2.d0
      t(2)=y*(a*x1 + rmu*x2)
      t(3)=sm2 + (y*x1)**2 + dfx(g,g2)*dfx(x1,x2)/4.d0
C
C
      dt(1,1)=dfx( dsqrt(2.d0)*y , g )
      dt(1,2)=y*a/x1
      dt(1,3)=2.d0*y*y + dfx(g,g2)/2.d0
C
      dt(2,1)=g*g
      dt(2,2)=rmu*y/x2
      dt(2,3)=-dfx(g,g2)/2.d0
C
      call cnf(t,dt,sl1,sl2,dsl)
C
      s1sl1=dsl(1,1)*flog(sl1,ts)/(32.d0*pi*pi)
      s2sl1=dsl(2,1)*flog(sl1,ts)/(32.d0*pi*pi)
      s1sl2=dsl(1,2)*flog(sl2,ts)/(32.d0*pi*pi)
      s2sl2=dsl(2,2)*flog(sl2,ts)/(32.d0*pi*pi)
```

```
C
       if(i.eq.ilim .and. k.eq.klim)then
       if(sl1.lt.0.d0 .or. sl2.lt.0.d0)
      # write(96,*)'some slepton^2 < 0'
       write(95,*)'Slepton corrections'
       write(95,*)'s1sl1 =',s1sl1,' s1sl2 =',s1sl2
       write(95,*)'s2sl1 =',s2sl1,' s2sl2 =',s2sl2
       endif
C
       return
       END

*==> snet.f <==
*******************************************
*                                         *
*  Sigma corrections due to sneutrinos    *
*                                         *
*          sfm2                           *
*         -------                         *
*          sqml1                          *
*          sqml2                          *
*          sqml3                          *
*                                         *
*******************************************
       SUBROUTINE snet(sfm2,g,g2,x1,x2,ts,s1,s2)
       IMPLICIT DOUBLE PRECISION (a-h,o-z)
       COMMON /wur/ wur,pi
       COMMON /ginrp/ i, k, ilim, klim
C
       sco=g*g + g2*g2
       sn2=sfm2 + sco*dfx(x1,x2)/4.d0
C
       s1=sco*flog(sn2,ts) / (64.d0*pi*pi)
       s2=-s1
C
       if(i.eq.ilim .and. k.eq.klim)then
       if(sn2.lt.0.d0)write(96,*)'some sneutrino^2 < 0'
       write(95,*)'Sneutrino corrections'
       write(95,*)'s1 =',s1
       write(95,*)'s2 =',s2
       endif
C
       return
       END

*==> sqd.f <==
**********************************************
*                                            *
*  Sigma corrections due to DOWN squarks     *
*                                            *
*   y             Yukawa                     *
*   a             Trilinear                  *
*   q             Renorm.  scale             *
*                                            *
*                                            *
*   sm2   |  smc2    |  y    |  a            *
*   ------|----------|-------|------         *
*   sqmq1 | sqmdc1   | yd1   | ad1           *
*   sqmq2 | sqmdc2   | yd2   | ad2           *
*   sqmq3 | sqmdc3   | yd3   | ad3           *
*                                            *
**********************************************
```



```
       SUBROUTINE sqd(sm2,smc2,y,a,rmu,g,g2,x1,x2,ts,
      #   s1sd1,s1sd2,s2sd1,s2sd2)
       IMPLICIT DOUBLE PRECISION (a-h,o-z)
       DOUBLE PRECISION t(3),dt(2,3),dsd(2,2)
       COMMON /wur/ wur,pi
       COMMON /ginrp/ i, k, ilim, klim
C
       gsum=g*g + 3.d0*g2*g2
C
       t(1)=smc2 + (y*x1)**2 + g*g*dfx(x2,x1)/6.d0
       t(2)=y*(a*x1 + rmu*x2)
       t(3)=sm2 + (y*x1)**2 + gsum*dfx(x2,x1)/12.d0
C
C
       dt(1,1)=dfx( dsqrt(2.d0)*y , g/dsqrt(3.d0) )
       dt(1,2)=y*a/x1
       dt(1,3)=dfx( dsqrt(2.d0)*y , dsqrt(gsum/6.d0) )
C
       dt(2,1)=g*g/3.d0
       dt(2,2)=rmu*y/x2
       dt(2,3)=gsum/6.d0
C
       call cnf(t,dt,sd1,sd2,dsd)
C
       s1sd1=3.d0*dsd(1,1)*flog(sd1,ts)/(32.d0*pi*pi)
       s2sd1=3.d0*dsd(2,1)*flog(sd1,ts)/(32.d0*pi*pi)
       s1sd2=3.d0*dsd(1,2)*flog(sd2,ts)/(32.d0*pi*pi)
       s2sd2=3.d0*dsd(2,2)*flog(sd2,ts)/(32.d0*pi*pi)
C
       if(i.eq.ilim .and. k.eq.klim)then
       if(sd1.lt.0.d0 .or. sd2.lt.0.d0)
      # write(96,*)'some down squark^2 < 0'
       write(95,*)'Down squark corrections'
       write(95,*)'s1sd1 =',s1sd1,' s1sd2 =',s1sd2
       write(95,*)'s2sd1 =',s2sd1,' s2sd2 =',s2sd2
       endif
C
       return
       END

*==> squ.f <==
**********************************************
*                                            *
*   Sigma corrections due to UP squarks      *
*                                            *
*   y             Yukawa                     *
*   a             Trilinear                  *
*   q             Renorm.  scale             *
*                                            *
*                                            *
*   sm2   |  smc2    |  y    |  a            *
*   ------|----------|-------|------         *
*   sqmq1 | sqmuc1   | yu1   | au1           *
*   sqmq2 | sqmuc2   | yu2   | au2           *
*   sqmq3 | sqmuc3   | yu3   | au3           *
*                                            *
**********************************************
       SUBROUTINE squ(sm2,smc2,y,a,rmu,g,g2,x1,x2,ts,
      #   s1su1,s1su2,s2su1,s2su2)
       IMPLICIT DOUBLE PRECISION (a-h,o-z)
       DOUBLE PRECISION t(3),dt(2,3),dsu(2,2)
       COMMON /wur/ wur,pi
```

```
      COMMON /ginrp/ i, k, ilim, klim
C
      t(1)=smc2 + (y*x2)**2 - g*g*dfx(x2,x1)/3.d0
      t(2)=-y*(rmu*x1 + a*x2)
      tal=dfx( g , dsqrt(3.d0)*g2 )*dfx(x2,x1)/12.d0
      t(3)=sm2 + (y*x2)**2 + tal
C
C
      dt(1,1)=2.d0*g*g/3.d0
      dt(1,2)=-y*rmu/x1
      dt(1,3)=-dfx( g , dsqrt(3.d0)*g2 )/6.d0
C
      dt(2,1)=2.d0*dfx( y , g/dsqrt(3.d0) )
      dt(2,2)=-y*a/x2
      dt(2,3)=2.d0*y*y + dfx( g , dsqrt(3.d0)*g2 )/6.d0
C
      call cnf(t,dt,su1,su2,dsu)
C
      s1su1=3.d0*dsu(1,1)*flog(su1,ts)/(32.d0*pi*pi)
      s2su1=3.d0*dsu(2,1)*flog(su1,ts)/(32.d0*pi*pi)
      s1su2=3.d0*dsu(1,2)*flog(su2,ts)/(32.d0*pi*pi)
      s2su2=3.d0*dsu(2,2)*flog(su2,ts)/(32.d0*pi*pi)
C
      if(i.eq.ilim .and. k.eq.klim)then
      if(su1.lt.0.d0 .or. su2.lt.0.d0)
     # write(96,*)'some up squark^2 < 0'
      write(95,*)'Up squark corrections'
      write(95,*)'s1su1 =',s1su1, '  s1su2 =',s1su2
      write(95,*)'s2su1 =',s2su1, '  s2su2 =',s2su2
      endif
C
      return
      END

*==> upq.f <==
*******************************************
*                                         *
*   Sigma corrections due to UP quarks    *
*                                         *
*   y    :  Yukawa                        *
*   x    :  Vev                           *
*   q    :  Renorm.  scale                *
*                                         *
*   y   |x                                *
*   ----|----------------                 *
*   yu1 |x2                               *
*   yu2 |x2    UP quarks                  *
*   yu3 |x2                               *
*                                         *
*******************************************
      SUBROUTINE upq(y,x,ts,s1,s2)
      IMPLICIT DOUBLE PRECISION (a-h,o-z)
      COMMON /wur/ wur,pi
      COMMON /ginrp/ i, k, ilim, klim
C
      qu2=(y*x)**2
      s1=0.d0
      s2=-3.d0*y*y*flog(qu2,ts)/(8.d0*pi*pi)
C
      if(i.eq.ilim .and. k.eq.klim)then
      write(95,*)'Up quark corrections'
      write(95,*)'s1 =',s1
```



```
      write(95,*)'s2 =',s2
      endif
C
      return
      END
```