thsavage

HEWLETT PACKARD

Request: labstacker-657 from sunny.isis.unc.edu

Options: flist=' func.f:36564 '
Invalid Options: '

Title: func.f

#####
sunny
Thu Jul 30 12:15:04 EDT 1998

as of 7/30/98

```
********* Option Summary *********
(see "man net_lj4x" for details)

auto (default), postscript, pcl, hpgl2, hpgl2_p, raw, relay
manual, tray1, tray2, tray3
legal, letter, A4, exec, ledger/11x17, com10
yb, nb                    bin1, bin2
dpi#                      simplex, duplex, hduplex
2up, 2+, 4up (hpux only)  portrait, landscape
color, gray
ascii,text                econo# (#=on/off)
```

```
      subroutine funcent (th,np,blog,iret)
      implicit real*8 (a-h,o-z)
      include 'incsiz.inc'
      parameter(oosq2pi = .398942280401432d0)
      parameter(rsqrt2 = .70710678118655D0)
      integer l,d,c,t,mm,kk,nl,icount,dd,cc
      character*8 xlabel(maxnp)
      dimension fpd(maxnp),spd(np,np),partial(maxnp),th(maxnp),
     &  beta(maxexplv,maxndep),gamma(maxexplv,maxndep),sd(maxndep),
     &  pw1(maxmass),hetero1(maxmass),rhod1(maxnt,maxndep),
     &  rhoc1(maxnt,maxndep),
     &  zg(maxndep,maxmass*maxndep,maxmass,maxmass,maxnt),
     &  xb(maxndep,maxmass*maxndep,maxmass,maxmass,maxnt),
     &  bpart(maxmass,maxmass),dnumer(maxnp),pw2(maxmass),
     &  hetero2(maxmass),rhod2(maxnt,maxndep),dpart1(maxmass,maxmass),
     &  prob(maxndep,maxmass*maxndep,maxmass,maxmass,maxnt),
     &  dpart2(maxmass,maxmass),
     &  dend(maxndep,maxmass*maxndep,maxmass,maxmass,maxnt),
     &  prt1(maxmass),shock1(maxmass),rhodt1(maxnt,maxndep),
     &  rhoct1(maxnt,maxndep),prt2(maxmass),shock2(maxmass),
     &  rhodt2(maxnt,maxndep),rhoct2(maxnt,maxndep),
     &  df1(maxndep,maxnt),cf1(maxndep,maxnt),tdpart1(maxmass),
     &  dft1(maxndep,maxnt),cft1(maxndep,maxnt),rhoc2(maxnt,maxndep),
     &  btemp(maxmass*maxndep,maxmass,maxmass,maxnt),contrib(maxnt),
     &  tdpart2(maxmass),df2(maxndep,maxnt),cf2(maxndep,maxnt),
     &  dft2(maxndep,maxnt),cft2(maxndep,maxnt),
     &  hetnld(maxndep,maxmass),hetnlc(maxndep,maxmass),
     &  pwnl(maxmass),ddnlm(maxndep*maxmass),dcnlm(maxndep*maxmass),
     &  tdpart3(maxmass)
        real*8 xbs(maxndep,maxnt),zgs(maxndep,maxnt)

      external ddfunc
      logical doder,dosp
      common /killme/ pw1,pw2,prt1,prt2,pwnl,hetero1,hetero2,shock1,
     &    shock2,hetnld,hetnlc,partial,fpnorm,oonob,rhod1,rhoc1,rhod2,
     &    rhoc2,rhodt1,rhoct1,rhodt2,rhoct2
      common /alabel/ xlabel
c      common /notest/noest(maxnp),nvrest(maxnp)
      blog=0.d0
      doder = .false.
      dosp = .false.
c      write(6,*) 'oonob:',oonob
c      pause
      go to 10

********** ENTRY POINT FOR FIRST PARTIALS ROUTINE **********

      entry fpent(th,np,f,fpd,iret)
      doder=.true.
      dosp = .false.
      if (.not.dosp) then
        do j=1,np
          fpd(j) = 0.d0
        enddo
        open (73,file=fpest10,form='unformatted')
        open (74,file=fpest9)
        do j = 1,np
          write(73) th(j)
          write(74,161) xlabel(j),'=',th(j)
        enddo
```

```
 161    format (t7,a8,a3,f12.5)
        close (73)
        close (74)
       endif

       if(ifdospds.eq.1) then
       do j = 1,np
       do k = 1,np
          savespd(j,k) = 0.d0
       enddo
       enddo
       endif

       go to 10

************ ENTRY POINT FOR SECOND PARTIALS ROUTINE **********

       entry spent(th,np,f,fpd,spd,iret)

       write(6,*) '  SHOULD NOT ENTER AT SPENT IN FUNC.F '
       write(12,*) '  SHOULD NOT ENTER AT SPENT IN FUNC.F '
       dosp = .true.
       doder = .true.
       do j = 1,np
       do k = 1,np
          spd(j,k) = 0.d0
       enddo
       enddo

*********** THE CALCULATIONS BELOW HERE ARE THE SAME **********
**************** FOR FUNC, FP, AND SP ROUTINES ****************

   10 continue

c beta is the vector of coefficients related to discrete
c outcomes.  gamma is the vector of coefficients related to
c continuous outcomes.  pw is the probability weight
c associated with each mass point.  hetero is the vector of mass
c points itself.  rho is the vector of factor loadings, and
c sd is the vector of standard deviations for each continuous
c variable.

************** ASSIGN PARAMETER VALUES TH(NP) *****************

       thsum1=1.d0
       thsum2=1.d0
       prtsum1 = 1.d0
       prtsum2 = 1.d0
       prnlsum=1.d0
       do d=1,discrete
       do k=1,ndvar(d)
c        beta(k,d)=ran3(-ndloc(d)-1)
       beta(k,d)=th(k+ndloc(d)-1)
c           write(6,*) 'beta(',k,d,') = ',beta(k,d)
       enddo
       if (nmass1.gt.1) then
          do nn=1,agevar
            rhod1(nn,d) = th(ndf1loc(nn,d))
c               write(6,*) 'rhod1(',nn,d,') = ',rhod1(nn,d)
          enddo
```

```
      endif
      if (nmass2.gt.1) then
         do nn=1,agevar
           rhod2(nn,d) = th(ndf2loc(nn,d))
c             write(6,*) 'rhod2: ',rhod2(nn,d)
         enddo
      endif
      if (nmasst1.gt.1) then
          do t=1,timevar
          rhodt1(t,d) = th(ndft1loc(t,d))
c               write(6,*) 'rhodt1: ',rhodt1(t,d)
          enddo
      endif
      if (nmasst2.gt.1) then
          do t=1,timevar
          rhodt2(t,d) = th(ndft2loc(t,d))
c               write(6,*) 'rhodt2: ',rhodt2(t,d)
          enddo
      endif
c        pause
      enddo
      do c=1,continuous
      do k=1,ncvar(c)
       gamma(k,c)=th(k+ncloc(c)-1)
c          write(6,*) 'gamma(',k,c,') = ',gamma(k,c)
      enddo
      sd(c) = th(nsdloc(c))
      if(sd(c).le.0.d0) then
       write(6,*) 'sd(',c,') = ',sd(c),'  WRONG SIGN or ZERO'
       iret = 1
       return
      endif
      if (nmass1.gt.1) then
          do nn=1,agevar
            rhoc1(nn,c) = th(ncf1loc(nn,c))
c             write(6,*) 'rhoc1(',nn,c,') = ',rhoc1(nn,c)
          enddo
      endif
      if (nmass2.gt.1) then
          do nn=1,agevar
            rhoc2(nn,c) = th(ncf2loc(nn,c))
c             write(6,*) 'rhoc2: ',rhoc2(nn,c)
          enddo
      endif
      if (nmasst1.gt.1) then
          do t=1,timevar
          rhoct1(t,c) = th(ncft1loc(t,c))
c               write(6,*) 'rhoct1(',t,c,') = ',rhoct1(t,c)
          enddo
      endif
      if (nmasst2.gt.1) then
          do t=1,timevar
          rhoct2(t,c) = th(ncft2loc(t,c))
c               write(6,*) 'rhoct2: ',rhoct2(t,c)
          enddo
      endif
      enddo
c      pause
      do j=1,nmass1-1
       thsum1 = thsum1 + exp(th(npw1loc(j)))
```

```
      enddo
c      write(6,*) 'thsum1 = ',thsum1
      pw1(1) = 1.d0/thsum1
c      write(6,*) 'pw1(1) = ',pw1(1)
      do j = 2,nmass1
       pw1(j) = exp(th(npw1loc(j-1)))/thsum1
c          write(6,*) 'th(',npw1loc(j-1),') = ',th(npw1loc(j-1))
c          write(6,*) 'pw1(',j,') = ',pw1(j)
      enddo
      do j=1,nmass1
       if (j.eq.1) then
         hetero1(j) = 0.d0
       else if (j.lt.nmass1) then
         hetero1(j) = exp(th(nm1loc(j-1)))/
     &                  (1.d0+exp(th(nm1loc(j-1))))
c          write(6,*) 'th(',nm1loc(j-1),') = ',th(nm1loc(j-1))
       else if (j.eq.nmass1) then
         hetero1(j) = 1.d0
       endif
c         write(6,*) 'hetero1(',j,') = ',hetero1(j)
c         pause
      enddo
      do j=1,nmass2-1
       thsum2 = thsum2 + exp(th(npw2loc(j)))
      enddo
c      write(6,*) 'thsum2 = ',thsum2
      pw2(1) = 1.d0/thsum2
c      write(6,*) 'pw2(1) = ',pw2(1)
      do j = 2,nmass2
       pw2(j) = exp(th(npw2loc(j-1)))/thsum2
c          write(6,*) 'th(',npw2loc(j-1),') = ',th(npw2loc(j-1))
c          write(6,*) 'pw2(',j,') = ',pw2(j)
      enddo
      do j=1,nmass2
       if (j.eq.1) then
         hetero2(j) = 0.d0
       else if (j.lt.nmass2) then
         hetero2(j) = exp(th(nm2loc(j-1)))/
     &                  (1.d0+exp(th(nm2loc(j-1))))
c          write(6,*) 'th(',nm2loc(j-1),') = ',th(nm2loc(j-1))
       else if (j.eq.nmass2) then
         hetero2(j) = 1.d0
       endif
c          write(6,*) 'hetero2(',j,') = ',hetero2(j)
c          pause
      enddo
      do j=1,nmasst1-1
       prtsum1 = prtsum1 + exp(th(npwt1loc(j)))
      enddo
c      write(6,*) 'prtsum1:',prtsum1
      prt1(1) = 1.d0/prtsum1
c      write(6,*) 'prt1(1) = ',prt1(1)
      do j = 2,nmasst1
       prt1(j) = exp(th(npwt1loc(j-1)))/prtsum1
c          write(6,*) 'th: ',th(npwt1loc(j-1))
c          write(6,*) 'prt1: ',prt1(j)
      enddo
      do j=1,nmasst1
       if (j.eq.1) then
         shock1(j) = 0.d0
```

```
        else if (j.lt.nmasst1) then
          shock1(j) = exp(th(nmt1loc(j-1)))/
     &                 (1.d0+exp(th(nmt1loc(j-1))))
c          write(6,*) 'th(',nmt1loc(j-1),') = ',th(nmt1loc(j-1))
        else if (j.eq.nmasst1) then
          shock1(j) = 1.d0
        endif
c         write(6,*) 'shock1(',j,') = ',shock1(j)
c         pause
       enddo
       do j=1,nmasst2-1
        prtsum2 = prtsum2 + exp(th(npwt2loc(j)))
       enddo
c       write(6,*) 'prtsum2 = ',prtsum2
       prt2(1) = 1.d0/prtsum2
c       write(6,*) 'prt2(1) = ',prt2(1)
       do j = 2,nmasst2
        prt2(j) = exp(th(npwt2loc(j-1)))/prtsum2
c         write(6,*) 'npwt2loc:',npwt2loc(j-1)
c         write(6,*) 'th: ',th(npwt2loc(j-1))
c         write(6,*) 'prt2: ',prt2(j)
       enddo
c       pause
       do j=1,nmasst2
        if (j.eq.1) then
          shock2(j) = 0.d0
        else if (j.lt.nmasst2) then
          shock2(j) = exp(th(nmt2loc(j-1)))/
     &                 (1.d0+exp(th(nmt2loc(j-1))))
c          write(6,*) 'th(',nmt2loc(j-1),') = ',th(nmt2loc(j-1))
        else if (j.eq.nmasst2) then
          shock2(j) = 1.d0
        endif
c         write(6,*) 'shock2(',j,') = ',shock2(j)
c         pause
       enddo

       do j=1,nnlmass-1
       prnlsum=prnlsum+exp(th(npwnlloc(j)))
       enddo
       pwnl(1)=1.d0/prnlsum
       do j=2,nnlmass
       pwnl(j)=exp(th(npwnlloc(j-1)))/prnlsum
       enddo

       icount=0
       do j1=1,discrete
       do j2=1,nnlmass
       if (j2.eq.1) hetnld(j1,j2)=0.d0
       if (j2.gt.1) then
       icount=icount+1
       hetnld(j1,j2)=th(nmnldloc(icount))
       endif
c       write(6,*) j1,j2,nmnldloc(icount),hetnld(j1,j2)
       enddo
       enddo

       icount=0
       do j1=1,continuous
       do j2=1,nnlmass
```

```
      if (j2.eq.1) hetnlc(j1,j2)=0.d0
      if (j2.gt.1) then
      icount=icount+1
      hetnlc(j1,j2)=th(nmnlcloc(icount))
      endif
c      write(6,*) j1,j2,nmnlcloc(icount),hetnlc(j1,j2)
      enddo
      enddo

c      pause

******** BEGIN LOOP OVER INDIVIDUALS I ***********

c      write(6,*) 'nmass1,nmass2,nmasst1,nmasst2,nnlmass:',nmass1,
c     &  nmass2,nmasst1,nmasst2,nnlmass
      do 100 i=1,nobs
      blike=0.d0
      if (doder) then
       do j=1,np
        dnumer(j)=0.d0
       enddo
      endif
c  first collect up non-hetero arguments
      do t=1,nt

       do d=1,discrete
        if (disc(d,t,i).gt.-9999) then
         xbs(d,t) = 0.d0
         do j=1,ndvar(d)
           xbs(d,t) = xbs(d,t) + beta(j,d)*x(locdisc(j,d),t,i)
         enddo
        endif
       enddo
c       if(i.eq.1) write(6,*) disc(d,t,i),xbs(d,t)

       do c=1,continuous
        if (cts(c,t,i).gt.-10000) then
         zgs(c,t)=0.d0
         do j=1,ncvar(c)
           zgs(c,t) =zgs(c,t) + gamma(j,c)*x(loccont(j,c),t,i)
         enddo
        endif
       enddo
c       if(i.eq.1) write(6,*) cts(c,t,i),zgs(c,t)

      enddo

******** BEGIN LOOP OVER MASS POINTS K ***********

      do 101 m=1,nmass2
      do 107 k=1,nmass1
       dpart1(k,m)=0.d0
       dpart2(k,m)=0.d0
       bpart(k,m) = big

************** BEGIN LOOP OVER TIME PERIODS T ***********
********** ALSO BEGIN CALCULATING FUNCTION VALUE ********

      do 2000 t=1,nt
c  skip those obs with missing data for one year of the survey
```

```
c      if (status(t,i).gt.0) go to 2000
       contrib(t) = 0.d0
********** BEGIN LOOP OVER PERIOD SPECIFIC SHOCKS ************

       do 3005 nl=1,nnlmass
       do 3003 mm = 1,nmasst2
       do 3000 kk = 1,nmasst1

        btemp(nl,kk,mm,t) = 1.d0

***** BEGIN ADDING UP CONTRIBUTION OF DISCRETE OUTCOMES ****
************** TO THE LIKELIHOOD FUNCTION ***************

       do 1001 d=1,discrete
       if (disc(d,t,i).gt.-10000) then
         xb(d,nl,kk,mm,t)=xbs(d,t)
         if (nmass1.gt.1) then
            df1(d,t) = rhod1(1,d)
            do nn = 2,agevar
              df1(d,t) = df1(d,t) + rhod1(nn,d)*age(t,i)**(nn-1)
            enddo
            xb(d,nl,kk,mm,t) = xb(d,nl,kk,mm,t) + df1(d,t)*hetero1(k)
c            if (i.eq.1264) write(6,*) 'xb:',xb(d,nl,kk,mm,t)
         endif
         if (nmass2.gt.1) then
            df2(d,t) = rhod2(1,d)
            do nn=2,agevar
            df2(d,t) = df2(d,t) + rhod2(nn,d)*age(t,i)**(nn-1)
            enddo
            xb(d,nl,kk,mm,t) = xb(d,nl,kk,mm,t) + df2(d,t)*hetero2(m)
c            if (i.eq.1264) write(6,*) 'xb:',xb(d,nl,kk,mm,t)
         endif
         if (nmasst1.gt.1) then
            dft1(d,t) = rhodt1(1,d)
            do nn = 2,timevar
            dft1(d,t) = dft1(d,t) + rhodt1(nn,d)*t**(nn-1)
            enddo
            xb(d,nl,kk,mm,t) = xb(d,nl,kk,mm,t) + dft1(d,t)*shock1(kk)
         endif
         if (nmasst2.gt.1) then
            dft2(d,t) = rhodt2(1,d)
            do nn = 2,timevar
            dft2(d,t) = dft2(d,t) + rhodt2(nn,d)*t**(nn-1)
            enddo
            xb(d,nl,kk,mm,t) = xb(d,nl,kk,mm,t) + dft2(d,t)*shock2(mm)
         endif
         if (nnlmass.gt.1) then
            xb(d,nl,kk,mm,t) = xb(d,nl,kk,mm,t) + hetnld(d,nl)
         endif
c         if (i.eq.1) write(6,*) 'xb:',xb(d,nl,kk,mm,t)

c         prob(d,nl,kk,mm,t)= xprob(xb(d,nl,kk,mm,t))
         prob(d,nl,kk,mm,t)= .5d0 + .5d0*derf(xb(d,nl,kk,mm,t)*rsqrt2)

         if(disc(d,t,i).eq.0) then
           btemp(nl,kk,mm,t) = btemp(nl,kk,mm,t)*
     &                         (1.d0-prob(d,nl,kk,mm,t))
         else if(disc(d,t,i).eq.1) then
           btemp(nl,kk,mm,t) = btemp(nl,kk,mm,t)*prob(d,nl,kk,mm,t)
         endif
```

```
c        write(6,*) 'btemp: ',btemp(nl,kk,mm,t)
       endif
 1001   enddo

***** BEGIN ADDING UP CONTRIBUTION OF CONTINUOUS OUTCOMES *****
***************** TO THE LIKELIHOOD FUNCTION ******************

      do 1002 c=1,continuous
       if (cts(c,t,i).gt.-10000) then
         zg(c,nl,kk,mm,t)=zgs(c,t)
         if (nmass1.gt.1) then
            cf1(c,t) = rhoc1(1,c)
            do nn = 2,agevar
            cf1(c,t) = cf1(c,t) + rhoc1(nn,c)*age(t,i)**(nn-1)
            enddo
            zg(c,nl,kk,mm,t) = zg(c,nl,kk,mm,t) + cf1(c,t)*hetero1(k)
         endif
c         if (i.eq.1) write(6,*) 'zg: ',zg(c,nl,kk,mm,t)
         if (nmass2.gt.1) then
            cf2(c,t) = rhoc2(1,c)
            do nn=2,agevar
            cf2(c,t) = cf2(c,t) + rhoc2(nn,c)*age(t,i)**(nn-1)
            enddo
            zg(c,nl,kk,mm,t) = zg(c,nl,kk,mm,t) + cf2(c,t)*hetero2(m)
         endif
c         if (i.eq.1) write(6,*) 'zg: ',zg(c,kk,mm,t)
         if (nmasst1.gt.1) then
            cft1(c,t) = rhoct1(1,c)
            do nn = 2,timevar
            cft1(c,t) = cft1(c,t) + rhoct1(nn,c)*t**(nn-1)
            enddo
            zg(c,nl,kk,mm,t) = zg(c,nl,kk,mm,t) + cft1(c,t)*shock1(kk)
         endif
         if (nmasst2.gt.1) then
            cft2(c,t) = rhoct2(1,c)
            do nn = 2,timevar
            cft2(c,t) = cft2(c,t) + rhoct2(nn,c)*t**(nn-1)
            enddo
            zg(c,nl,kk,mm,t) = zg(c,nl,kk,mm,t) + cft2(c,t)*shock2(mm)
         endif

      if (nnlmass.gt.1) then
         zg(c,nl,kk,mm,t) = zg(c,nl,kk,mm,t) + hetnlc(c,nl)
      endif
c         if (i.eq.1) write(6,*) 'zg: ',zg(c,nl,kk,mm,t)

         c2 = (cts(c,t,i)-zg(c,nl,kk,mm,t))**2.d0
         sd2 = sd(c)*sd(c)
         denc = (oosq2pi/sd(c))*exp(-c2/(2.d0*sd2))
         btemp(nl,kk,mm,t) = btemp(nl,kk,mm,t)*denc
       endif
 1002   enddo
        if (btemp(nl,kk,mm,t).le.0.d0) go to 3000
      contrib(t) = contrib(t) +
     &      pwnl(nl)*prt1(kk)*prt2(mm)*btemp(nl,kk,mm,t)
c      if (i.eq.1264) write(6,*) 'contrib(t):',contrib(t)
 3000   enddo
 3003   enddo
 3005   enddo
```

```
c      write(6,*) i,t,'contrib(t):',contrib(t)
       bpart(k,m) = bpart(k,m)*contrib(t)
c      write(6,*) i,t,'bpart:',bpart(k,m)
c      if (i.eq.1264) write(6,*) 'bpart:',bpart(k,m)
 2000   enddo

******* TOTAL CONTRIBUTION TO THE LIKELIHOOD FUNCTION ********

       blike = blike + pw1(k)*pw2(m)*bpart(k,m)
c      write(6,*) 'blike:',blike,'   for obs',i,'    on mass points',k,m




*********  BEGIN DERIVATIVES SECTION HERE **************

*********  BEGIN LOOPING OVER TIME PERIODS *************

       if (doder) then
      do 2001 t=1,nt
c        skip those branches with zero probability/missing data
      if (contrib(t).le.0.d0) go to 2001
      if (status(t,i).gt.0) go to 2001

********** BEGIN LOOP OVER PERIOD SPECIFIC SHOCKS ************

      do 3006 nl=1,nnlmass
      do 3004 mm = 1,nmasst2
      do 3001 kk = 1,nmasst1
         dmpart1 = 0.d0
         dmpart2 = 0.d0
      do dd=1,discrete
      ddnlm(dd)=0.d0
      enddo
      do cc=1,continuous
      dcnlm(cc)=0.d0
      enddo

         if (btemp(nl,kk,mm,t).le.0.d0) go to 3001

********  ADD UP DERIVATIVES FOR DISCRETE OUTCOMES *********

      do 1003 d=1,discrete
       if (disc(d,t,i).gt.-10000) then
         xb2o2 = (xb(d,nl,kk,mm,t)**2.d0)/2.d0
         dend(d,nl,kk,mm,t) = oosq2pi*exp(-xb2o2)

          if (disc(d,t,i).eq.1) then
          term =
     &            (pw1(k)*pw2(m)*bpart(k,m)*pwnl(nl)*
     &            prt1(kk)*prt2(mm)*btemp(nl,kk,mm,t)*dend(d,nl,kk,mm,t)
     &           /prob(d,nl,kk,mm,t))/contrib(t)
        else if (disc(d,t,i).eq.0) then
          term =  -
     &            (pw1(k)*pw2(m)*bpart(k,m)*pwnl(nl)*
     &            prt1(kk)*prt2(mm)*btemp(nl,kk,mm,t)*dend(d,nl,kk,mm,t)
     &           /(1.d0-prob(d,nl,kk,mm,t))) /contrib(t)
        endif
```

```
          do j=1,ndvar(d)
             ja = ndloc(d)+j-1
             dnumer(ja) = dnumer(ja) + term*
     &                   x(locdisc(j,d),t,i)

          enddo

******** DERIVATIVES FOR TIME-INVARIANT FACTOR LOADS *******
****************** DISCRETE OUTCOMES **********************

******** FIRST TYPE OF PERMANNENT HETEROGENEITY ************

         if (nmass1.gt.1) then
           do nn = 1,agevar
             mmm = ndf1loc(nn,d)
            dnumer(mmm) = dnumer(mmm)+term*
     &                   hetero1(k)*(age(t,i)**(nn-1))
           enddo
         endif

************* SECOND TYPE OF PERMANENT HETEROGENEITY ********

          if (nmass2.gt.1) then
           do nn = 1,agevar
             mmm = ndf2loc(nn,d)
            dnumer(mmm) = dnumer(mmm) + term*
     &                    hetero2(m)*(age(t,i)**(nn-1))
           enddo
         endif

******** DERIVATIVES FOR PERIOD-SPECIFIC FACTOR LOADS *******
******************** DISCRETE OUTCOMES **********************

***************** FIRST TRANSITORY SHOCK *********************

          if (nmasst1.gt.1) then
           do nn = 1,timevar
             mmm = ndft1loc(nn,d)
                dnumer(mmm) = dnumer(mmm) + term*
     &                        shock1(kk)*(t**(nn-1))
           enddo
c               write(6,*) i,mmm,dnumer(mmm)
         endif

**************** SECOND TRANSITORY SHOCK *********************

         if (nmasst2.gt.1) then
          do nn = 1,timevar
            mmm = ndft2loc(nn,d)
               dnumer(mmm) = dnumer(mmm) + term*
     &                        shock2(mm)*(t**(nn-1))
          enddo
         endif

******* FIRST STEP FOR TAKING DERIVATIVES WRT TIME-INVARIANT *****
************************ MASS POINTS ****************************

************ FIRST TYPE OF PERMANENT HETEROGENEITY ***************
```

```
        if (k.gt.1.and.k.lt.nmass1) then
          if (disc(d,t,i).eq.1) then
           dpartd = pwnl(nl)*prt1(kk)*prt2(mm)*btemp(nl,kk,mm,t)*
     &                dend(d,nl,kk,mm,t)
     &               *df1(d,t)/(prob(d,nl,kk,mm,t)*contrib(t))
          else if (disc(d,t,i).eq.0) then
          dpartd = -pwnl(nl)*prt1(kk)*prt2(mm)*btemp(nl,kk,mm,t)*
     &                dend(d,nl,kk,mm,t)
     &               *df1(d,t)/((1.d0-prob(d,nl,kk,mm,t))*contrib(t))
          endif
          dpart1(k,m) = dpart1(k,m) + dpartd
        endif

************ SECOND TYPE OF PERMANENT HETEROGENEITY **************

        if (m.gt.1.and.m.lt.nmass2) then
          if (disc(d,t,i).eq.1) then
          dpartd = pwnl(nl)*prt1(kk)*prt2(mm)*btemp(nl,kk,mm,t)*
     &                 dend(d,nl,kk,mm,t)
     &                *df2(d,t)/(prob(d,nl,kk,mm,t)*contrib(t))
          else if (disc(d,t,i).eq.0) then
          dpartd = -pwnl(nl)*prt1(kk)*prt2(mm)*btemp(nl,kk,mm,t)*
     &                 dend(d,nl,kk,mm,t)*df2(d,t)/
     &                 ((1.d0-prob(d,nl,kk,mm,t))*contrib(t))
          endif
          dpart2(k,m) = dpart2(k,m) + dpartd
        endif

****** FIRST STEP FOR TAKING DERIVATIVES WRT PERIOD-SPECIFIC *****
************************ MASS POINTS ****************************

****************** FIRST TYPE OF HETEROGENEITY *******************

        if ((kk.gt.1).and.(kk.lt.nmasst1)) then
            exparg = exp(th(nmt1loc(kk-1)))/
     &                (1.d0+exp(th(nmt1loc(kk-1))))
          if (disc(d,t,i).eq.1) then
           dmpart1 = dmpart1 + pwnl(nl)*prt1(kk)*prt2(mm)*
     &            dend(d,nl,kk,mm,t)
     &           *dft1(d,t)*exparg*(1.d0-exparg)*btemp(nl,kk,mm,t)
     &           /(prob(d,nl,kk,mm,t)*contrib(t))
          else if (disc(d,t,i).eq.0) then
           dmpart1 = dmpart1 - pwnl(nl)*prt1(kk)*prt2(mm)*
     &            dend(d,nl,kk,mm,t)
     &           *dft1(d,t)*exparg*(1.d0-exparg)*btemp(nl,kk,mm,t)
     &           /((1.d0-prob(d,nl,kk,mm,t))*contrib(t))
          endif
        endif

*************** SECOND TYPE OF HETEROGENEITY *******************

        if ((mm.gt.1).and.(mm.lt.nmasst2)) then
            exparg = exp(th(nmt2loc(mm-1)))/
     &                (1.d0+exp(th(nmt2loc(mm-1))))
          if (disc(d,t,i).eq.1) then
           dmpart2 = dmpart2 + pwnl(nl)*prt2(mm)*prt1(kk)*
     &            dend(d,nl,kk,mm,t)
     &           *dft2(d,t)*exparg*(1.d0-exparg)*btemp(nl,kk,mm,t)
     &           /(prob(d,nl,kk,mm,t)*contrib(t))
```

```
          else if (disc(d,t,i).eq.0) then
           dmpart2 = dmpart2 - pwnl(nl)*prt1(kk)*prt2(mm)*
     &             dend(d,nl,kk,mm,t)
     &            *dft2(d,t)*exparg*(1.d0-exparg)*btemp(nl,kk,mm,t)
     &            /((1.d0-prob(d,nl,kk,mm,t))*contrib(t))
          endif
         endif

        endif

****************** NONLINEAR HETEROGENEITY ************************

       if (nl.gt.1) then
       if (disc(d,t,i).eq.1) then
       ddnlm(d)=ddnlm(d) + pwnl(nl)*prt2(mm)*prt1(kk)*
     &             dend(d,nl,kk,mm,t)*btemp(nl,kk,mm,t)
     &            /(prob(d,nl,kk,mm,t)*contrib(t))
       elseif (disc(d,t,i).eq.0) then
       ddnlm(d)=ddnlm(d) - pwnl(nl)*prt2(mm)*prt1(kk)*
     &             dend(d,nl,kk,mm,t)*btemp(nl,kk,mm,t)
     &            /((1.d0-prob(d,nl,kk,mm,t))*contrib(t))
       endif
       endif

 1003     enddo

******** END DERIVATIVES WRT DISCRETE OUTCOMES ***********

******** BEGIN DERIVATIVES WRT CONTINUOUS OUTCOMES ********

      do 1004 c=1,continuous
       if (cts(c,t,i).gt.-10000.d0) then
         termc =
     &             pw1(k)*pw2(m)*bpart(k,m)*
     &             btemp(nl,kk,mm,t)*pwnl(nl)*prt1(kk)*prt2(mm)*
     &  (cts(c,t,i) - zg(c,nl,kk,mm,t))/(contrib(t)*sd(c)**2.d0)
        do j=1,ncvar(c)
         ja = ncloc(c)+j-1
         dnumer(ja)=dnumer(ja) + termc*
     &               x(loccont(j,c),t,i)

        enddo

*********** DERIVATIVES WRT STANDARD DEVIATIONS *************

          res2 = ((cts(c,t,i)-zg(c,nl,kk,mm,t))/sd(c))**2.d0
          dnumer(nsdloc(c)) = dnumer(nsdloc(c)) + pw1(k)*
     &          pw2(m)*bpart(k,m)*pwnl(nl)*prt1(kk)*prt2(mm)*
     &           btemp(nl,kk,mm,t)
     &         *(res2-1.d0)/(sd(c)*contrib(t))

******** DERIVATIVES WRT TIME-INVARIANT FACTOR LOAD ***********
****************** CONTINUOUS OUTCOMES ************************

************* FIRST PERMANENT HETEROGENEITY COMPONENT *********

        if (nmass1.gt.1) then
          do nn = 1,agevar
            mmm = ncf1loc(nn,c)
            dnumer(mmm) = dnumer(mmm) + termc*
```

```
     &            hetero1(k)*(age(t,i)**(nn-1))
           enddo
         endif

************* SECOND PERMANENT HETEROGENITY COMPONENT ***********

         if (nmass2.gt.1) then
            do nn = 1,agevar
           mmm = ncf2loc(nn,c)
            dnumer(mmm) = dnumer(mmm) + termc*
     &            hetero2(m)*(age(t,i)**(nn-1))
            enddo
         endif

******** DERIVATIVES WRT PERIOD-SPECIFIC FACTOR LOAD ********
****************** CONTINUOUS OUTCOMES **********************

*************** FIRST TRANSITORY FACTOR *******************

         if (nmasst1.gt.1) then
          do nn = 1,timevar
            mmm = ncft1loc(nn,c)
            dnumer(mmm)=dnumer(mmm) + termc*
     &                 shock1(kk)*(t**(nn-1))
          enddo
         endif

*************** SECOND TRANSITORY FACTOR *******************

         if (nmasst2.gt.1) then
          do nn = 1,timevar
            mmm = ncft2loc(nn,c)
            dnumer(mmm)=dnumer(mmm) + termc*
     &                 shock2(mm)*(t**(nn-1))

          enddo
         endif


******** SECOND STEP FOR TAKING DERIVATIVES WRT MASS POINTS *******
********** HAVE TO ADD ON CONTRIB. OF CONTINUOUS OUTCOMES *********

************ FIRST PERMANENT HETEROGENEITY COMPONENT **************

        if (k.gt.1.and.k.lt.nmass1) then
           dpartc = pwnl(nl)*prt1(kk)*prt2(mm)*btemp(nl,kk,mm,t)*
     &            cf1(c,t)
     &          *(cts(c,t,i)-zg(c,nl,kk,mm,t))/(contrib(t)*sd(c)**2.d0)
           dpart1(k,m) = dpart1(k,m) + dpartc
        endif

************ SECOND PERMANENT HETEROGENEITY COMPONENT ************

        if (m.gt.1.and.m.lt.nmass2) then
          dpartc = pwnl(nl)*prt1(kk)*prt2(mm)*btemp(nl,kk,mm,t)*cf2(c,t)
     &         *(cts(c,t,i)-zg(c,nl,kk,mm,t))/(contrib(t)*sd(c)**2.d0)
          dpart2(k,m) = dpart2(k,m) + dpartc
        endif

***** SECOND STEP FOR TAKING DERIVATIVES WRT PERIOD-SPECIFIC *****
```

```
************************ MASS POINTS ****************************

****************** FIRST TRANSITORY FACTOR **********************

         if ((kk.gt.1).and.(kk.lt.nmasst1)) then
          exparg = exp(th(nmt1loc(kk-1)))/(1.d0 +
     &                  exp(th(nmt1loc(kk-1))))
          dmpart1 = dmpart1 + pwnl(nl)*prt1(kk)*prt2(mm)*cft1(c,t)
     &              *exparg*(1.d0-exparg)*btemp(nl,kk,mm,t)
     &              *(cts(c,t,i)-zg(c,nl,kk,mm,t))/(contrib(t)*sd(c)**2)
         endif

****************** SECOND TRANSITORY FACTOR **********************

         if ((mm.gt.1).and.(mm.lt.nmasst2)) then
          exparg = exp(th(nmt2loc(mm-1)))/(1.d0 +
     &                  exp(th(nmt2loc(mm-1))))
          dmpart2 = dmpart2 + pwnl(nl)*prt1(kk)*prt2(mm)*cft2(c,t)
     &              *exparg*(1.d0-exparg)*btemp(nl,kk,mm,t)
     &              *(cts(c,t,i)-zg(c,nl,kk,mm,t))/(contrib(t)*sd(c)**2)
         endif

****************** NONLINEAR HETEROGENEITY ************************

       if (nl.gt.1) then
          dcnlm(c)=dcnlm(c) + pwnl(nl)*prt1(kk)*prt2(mm)*
     &      btemp(nl,kk,mm,t)*(cts(c,t,i)-zg(c,nl,kk,mm,t))
     &      /(contrib(t)*sd(c)**2)
c       if(i.eq.1) write(6,*) c,dcnlm(c)
       endif

       endif
 1004     enddo

******** END OF LOOP OVER CONTINUOUS OUTCOMES **********

************** DERIVS WRT PERIOD-SPECIFIC PWTs **************

***************** FIRST TRANSITORY FACTOR ********************
       do jj = 1,nmasst1-1
         if (kk.eq.jj+1) then
            tdpart1(jj) = btemp(nl,kk,mm,t)*pwnl(nl)*prt2(mm)*
     &                   prt1(kk)*(1.d0-prt1(kk))
         else if (kk.le.nmasst1) then
            tdpart1(jj) = -btemp(nl,kk,mm,t)*pwnl(nl)*prt2(mm)*
     &                   prt1(jj+1)*prt1(kk)
         endif
         dnumer(npwt1loc(jj)) = dnumer(npwt1loc(jj)) +
     &             tdpart1(jj)*bpart(k,m)*pw1(k)*pw2(m)/contrib(t)
       enddo

***************** SECOND TRANSITORY FACTOR ********************

       do jj = 1,nmasst2-1
         if (mm.eq.jj+1) then
            tdpart2(jj) = btemp(nl,kk,mm,t)*pwnl(nl)*prt1(kk)*
     &                   prt2(mm)*(1.d0-prt2(mm))
         else if (mm.le.nmasst2) then
            tdpart2(jj) = -btemp(nl,kk,mm,t)*pwnl(nl)*prt1(kk)*
     &                   prt2(jj+1)*prt2(mm)
```

```
        endif
c           write(6,*) 'jj,kk,mm,t:',jj,kk,mm,t
c           write(6,*) 'btemp:',btemp(nl,kk,mm,t)
c           write(6,*) 'prt1(kk):',prt1(kk)
c           write(6,*) 'prt2(mm):',prt2(mm)
c           write(6,*) 'prt2(jj+1):',prt2(jj+1)
c           write(6,*) 'tdpart2:',tdpart2(jj)
        dnumer(npwt2loc(jj)) = dnumer(npwt2loc(jj)) +
     &           tdpart2(jj)*bpart(k,m)*pw1(k)*pw2(m)/contrib(t)
c              write(6,*) 'npwt2loc(jj):',npwt2loc(jj)
c              write(6,*) 'bpart(k,m):',bpart(k,m)
c              write(6,*) 'pw1(k) & pw2(m):',pw1(k),pw2(m)
c              write(6,*) 'contrib(t):',contrib(t)
c              write(6,*) 'dnumer:',dnumer(npwt2loc(jj))
       enddo

       do jj=1,nnlmass-1
         if (nl.eq.jj+1) then
            tdpart3(jj) = btemp(nl,kk,mm,t)*prt1(kk)*
     &                  prt2(mm)*pwnl(nl)*(1.d0-pwnl(nl))
         else if (nl.le.nnlmass) then
            tdpart3(jj) = -btemp(nl,kk,mm,t)*prt1(kk)*
     &                  prt2(mm)*pwnl(jj+1)*pwnl(nl)
         endif
         dnumer(npwnlloc(jj))=dnumer(npwnlloc(jj))+
     &      tdpart3(jj)*bpart(k,m)*pw1(k)*pw2(m)/contrib(t)
       enddo

************ END OF LOOP OVER PERIOD-SPECIFIC PWTs ***********

********* ADD UP DERIVS WRT PERIOD-SPECIFIC MASS PTS *********

***************** FIRST TRANSITORY FACTOR ********************

       if ((kk.gt.1).and.(kk.lt.nmasst1)) then
          dnumer(nmt1loc(kk-1)) = dnumer(nmt1loc(kk-1)) +
     &                pw1(k)*pw2(m)*bpart(k,m)*dmpart1
       endif

***************** SECOND TRANSITORY FACTOR ********************

       if ((mm.gt.1).and.(mm.lt.nmasst2)) then
          dnumer(nmt2loc(mm-1)) = dnumer(nmt2loc(mm-1)) +
     &                pw1(k)*pw2(m)*bpart(k,m)*dmpart2
       endif

***************** NONLINEAR TRANSITORY FACTORS ****************

       if (nl.gt.1) then

       icount=nl-1
       do dd=1,discrete
       dnumer(nmnldloc(icount))=dnumer(nmnldloc(icount)) +
     &       pw1(k)*pw2(m)*bpart(k,m)*ddnlm(dd)
       icount=icount+nnlmass-1
       enddo

       icount=nl-1
       do cc=1,continuous
       dnumer(nmnlcloc(icount))=dnumer(nmnlcloc(icount)) +
```

```
     &       pw1(k)*pw2(m)*bpart(k,m)*dcnlm(cc)
c      if(i.eq.1) write(6,*) cc,icount,dnumer(nmnlcloc(icount))
       icount=icount+nnlmass-1
       enddo

       endif

 3001    enddo
 3004    enddo
 3006    enddo

 2001    enddo

************** END LOOP OVER TIME PERIODS *********************

*********** TAKE DERIVATIVES WRT THE PROB WEIGHTS *************
***************** TIME-INVARIANT HETEROG. *********************

********* FIRST PERMANENT HETEROGENEITY COMPONENT *************

      do j=1,nmass1-1
c             'exparg' is the value of the derivative of
c              a logit probability 'pw1'.
           if (k.eq.1) then
             exparg = -pw1(j+1)/thsum1
          else if (k.eq.(j+1)) then
              exparg = pw1(k)*(1.d0-pw1(k))
          else if (k.le.nmass1) then
              exparg = -pw1(k)*pw1(j+1)
          endif
            dnumer(npw1loc(j)) = dnumer(npw1loc(j)) +
     &                           pw2(m)*bpart(k,m)*exparg
       enddo

********* SECOND PERMANENT HETEROGENEITY COMPONENT **********

      do j=1,nmass2-1
c             'exparg' is the value of the derivative of
c              a logit probability 'pw2'.
           if (m.eq.1) then
             exparg = -pw2(j+1)/thsum2
          else if (m.eq.(j+1)) then
              exparg = pw2(m)*(1.d0-pw2(m))
          else if (m.le.nmass2) then
              exparg = -pw2(m)*pw2(j+1)
          endif
            dnumer(npw2loc(j)) = dnumer(npw2loc(j)) +
     &                           pw1(k)*bpart(k,m)*exparg
       enddo

******* ENDIF FROM 'IF DODER = TRUE' ABOVE ********

       endif

******* END LOOP OVER MASS POINTS *****************

  107 enddo
  101 enddo

       if (blike.le.0.d0) then
```

```
          if (.not.doder) then
           write(6,*) '  i:',i,'   blike@101:',blike,'  too small'
           iret = 1
           return
          endif

            write(6,*) '  Evaluating Derivatives and'
            write(6,*) ' Function value does not exist. Obs :',i
            write(12,*) '  Evaluating Derivatives and'
            write(12,*) ' Function value does not exist. Obs :',i
            do t=1,nt
           do d=1,discrete
             write(6,*) '  disc(d,t,i):',disc(d,t,i)
             write(6,*) 'explanatory vars:',(x(locdisc(l,d),t,i),l=1,np)
             write(12,*) '  disc(d,t,i):',disc(d,t,i)
             write(12,*) 'explanatory vars:',(x(locdisc(l,d),t,i),l=1,np)
           enddo
           do c=1,continuous
             write(6,*) '  cts(c,t,i):',cts(c,t,i)
c             write(6,*) 'explanatory vars:',(z(l,c,t,i),l=1,np)
             write(6,*) 'explanatory vars:',(x(loccont(l,c),t,i),l=1,np)
             write(12,*) '  cts(c,t,i):',cts(c,t,i)
             write(12,*) 'explanatory vars:',(x(loccont(l,c),t,i),l=1,np)
c             write(12,*) 'explanatory vars:',(z(l,c,t,i),l=1,np)
           enddo
            enddo
            write(6,*) '  *** STOPPING ***'
            write(12,*) '  *** STOPPING ***'
            stop
       endif

****** FINISH CALCULATING DERIVATIVES, GET HESSIAN ***********

       if(doder) then

*******  THIRD STEP FOR TAKING DERIVATIVES WRT MASS POINTS ********

         do m=1,nmass2
         do k=1,nmass1-2
           dnumer(nm1loc(k))=dnumer(nm1loc(k))+hetero1(k+1)*
     &                     (1.d0-hetero1(k+1))*pw1(k+1)*pw2(m)
     &                     *dpart1(k+1,m)*bpart(k+1,m)
       enddo
       enddo

       do k=1,nmass1
       do m=1,nmass2-2
           dnumer(nm2loc(m))=dnumer(nm2loc(m))+hetero2(m+1)*
     &                     (1.d0-hetero2(m+1))*pw2(m+1)*pw1(k)
     &                     *dpart2(k,m+1)*bpart(k,m+1)
       enddo
       enddo

         do j=1,np
          adder1 = wt(i)*dnumer(j)/blike
c          write(6,*) ' adder1: ',adder1,'    blike: ',blike
c          write(6,*) 'for obs:',i,'        on parm: ',j
         if (.not.dosp) fpd(j) = fpd(j) + adder1
c               write(6,*) 'fpd(',j,'): ',fpd(j)
         if (dosp) then
```

```
            do k=j,np
              adder2=wt(i)*dnumer(k)/blike
              spd(j,k) = spd(j,k) - adder1*adder2
c                    write(6,*) 'spd(j,k) : ',spd(j,k)
            enddo
          endif
          if(ifdospds.eq.1) then
            do k=j,np
              adder2=wt(i)*dnumer(k)/blike
              savespd(j,k) = savespd(j,k) - adder1*adder2
c                    write(6,*) 'spd(j,k) : ',spd(j,k)
            enddo
          endif

        enddo
      endif

******** END OF DERIVATIVES SECTION ****************

******** ADD UP THE LOG LIKELIHOOD FUNCTION ************

      if ((.not.doder).and.(.not.dosp)) then
c           write(6,*) 'obs,BLIKE :',i,BLIKE
        if (blike.gt.0.d0) blog = blog + log(BLIKE)*wt(i)
c           write(6,*) 'obs,BLOG : ',i,BLOG
c           PAUSE
      endif

************ END LOOP OVER INDIVIDUALS ***************

  100 continue

************ NORMALIZE FUNC,FP AND SP ****************

      if (.not.doder) then
        blog = blog*oonob
c           write(6,*) 'blog:',blog
c           write(6,*) 'oonob',oonob
c           write(6,*) 'normalized blog: ',blog
      endif
      if ((doder).and.(.not.dosp)) then
      do j = 1, np
c         if(noest(j).eq.1 .or. nvrest(j).eq.1) then
c           fpd(j) = 0.d0
c         else
          fpd(j) = fpd(j)*oonob
c          write(6,*) 'j:',j,'              fpd:',fpd(j)
c         endif
      enddo
      endif
      if (dosp) then
      do jj = 1,np
         do kk = jj,np
            spd(jj,kk) = spd(jj,kk)*oonob
         enddo
      enddo
      endif
      if (ifdospds.eq.1) then
      do jj = 1,np
         do kk = jj,np
```

```
            savespd(jj,kk) = savespd(jj,kk)*oonob
          enddo
        enddo
        endif

******* MAKE HESSIAN USING AVERAGE OF TWO VALUES*********
************* ALSO MAKE HESSIAN SYMMETRIC ***************

        if (dosp) then
          do j = 1,np
c           if(noest(j).eq.1 .or. nvrest(j).eq.1) then
c              do k=1,np
c               spd(j,k) = 0.d0
c               spd(k,j) = 0.d0
c              enddo
c              spd(j,j) = -1.d0
c           else
              do k = j+1,np
                spd(k,j) = spd(j,k)
c               spd(j,k) = 0.5d0*(spd(j,k) + spd(k,j))
c               spd(k,j) = spd(j,k)
              enddo
c           endif
          enddo
        endif

        if (ifdospds.eq.1) then
          do j = 1,np
              do k = j+1,np
                savespd(k,j) = savespd(j,k)
c               savespd(j,k) = 0.5d0*(savespd(j,k) + savespd(k,j))
c               savespd(k,j) = savespd(j,k)
              enddo
          enddo
        endif

******** CALCULATE THE NORM OF THE FIRST PARTIAL VECTOR ******

        if(doder) then
        fpnorm = 0.d0
          do j=1,np
              fpnorm = fpnorm + fpd(j)*fpd(j)
              partial(j) = fpd(j)
          enddo
          fpnorm = sqrt(fpnorm)/np
          write(6,*) 'fpnorm: ',fpnorm
          write(12,*) 'fpnorm: ',fpnorm
        endif

        RETURN
        END
```