```
      subroutine walk_scan
      Implicit real*8(a-h,o-z)
      double precision:: alpha,beta,beta0,alpha0,r0,pas
      integer:: cont, m, mi
      common/dades1/Z,beta,alpha,r0,damping
      common/dades2/m, mi,sec,encerts
      common/results/ene1,ene2
      double precision E,E2,Ea,Eb,Fa,Fb,EEa,EEb,EFa,EFb,EaEb,Ea2,Eb2
      double precision E0,E20,Ea0,Eb0,Fa0,Fb0,EEa0,EEb0,EFa0,EFb0,EaEb0,Ea20,Eb20
      double precision det, aux, auxa, auxb
      double precision, dimension(2):: grad
      double precision, dimension(2,2) :: hess,hessm1

      x1=r0*(2*ran2(idum)-1)
      y1=r0*(2*ran2(idum)-1)
      z1=r0*(2*ran2(idum)-1)
      x2=r0*(2*ran2(idum)-1)
      y2=r0*(2*ran2(idum)-1)
      z2=r0*(2*ran2(idum)-1)

      r1=sqrt(x1**2+y1**2+z1**2)
      r2=sqrt(x2**2+y2**2+z2**2)
      dist=sqrt((x1-x2)**2+(y1-y2)**2+(z1-z2)**2)

      grad(1)=alpha
      grad(2)=beta

      do i=1,2
       do j=1,2
          hess(i,j)=0.d0
          hessm1(i,j)=0.d0
       enddo
      enddo

      p0= prob(r1,r2,dist)

!       write(*,*) "p0 ",p0


      ! termalitzation

    cont=0
    do i=1,mi

      x1b=r0*(2*ran2(idum)-1)
      y1b=r0*(2*ran2(idum)-1)
      z1b=r0*(2*ran2(idum)-1)
      x2b=r0*(2*ran2(idum)-1)
      y2b=r0*(2*ran2(idum)-1)
      z2b=r0*(2*ran2(idum)-1)

      r1b=sqrt(x1b**2+y1b**2+z1b**2)
      r2b=sqrt(x2b**2+y2b**2+z2b**2)
      distb=sqrt((x1b-x2b)**2+(y1b-y2b)**2+(z1b-z2b)**2)

      p1=prob(r1b,r2b,distb)

!       write(*,*) "p1 ",p1

      w=ran2(idum);

      IF(p1/p0>sec*w) THEN
          x1=x1b
          y1=y1b
          z1=z1b
          x2=x2b
          y2=y2b
          z2=z2b
          p0=p1
          cont=cont+1
      ENDIF
```

```
    enddo

!   write(*,*) "cont ",cont
!   write(*,*) "initial  step",r0
        pas=r0*cont/(encerts*mi);
!   write(*,*) " fitted step ",pas


    E0=0.d0
    E20=0.d0
    Ea0=0.d0
    Eb0=0.d0
    Fa0=0.d0
    Fb0=0.d0
    EFa0=0.d0
    EFb0=0.d0
    Fab0=0.d0
    Faa0=0.d0
    Fbb0=0.d0
    EFab0=0.d0
    EFaa0=0.d0
    EFbb0=0.d0
    FaFb0=0.d0
    FaFa0=0.d0
    FbFb0=0.d0
    EFaFb0=0.d0
    EFaFa0=0.d0
    EFbFb0=0.d0
    FaEb0=0.d0
    FbEa0=0.d0

    E=0.d0
    E2=0.d0
    Ea=0.d0
    Eb=0.d0
    Fa=0.d0
    Fb=0.d0
    EFa=0.d0
    EFb=0.d0
    Fab=0.d0
    Faa=0.d0
    Fbb=0.d0
    EFab=0.d0
    EFaa=0.d0
    EFbb=0.d0
    FaFb=0.d0
    FaFa=0.d0
    FbFb=0.d0
    EFaFb=0.d0
    EFaFa=0.d0
    EFbFb=0.d0
    FaEb=0.d0
    FbEa=0.d0

!

    cont=0
    do i=1,m

        x1b=pas*(2*ran2(idum)-1)
        y1b=pas*(2*ran2(idum)-1)
        z1b=pas*(2*ran2(idum)-1)
        x2b=pas*(2*ran2(idum)-1)
        y2b=pas*(2*ran2(idum)-1)
        z2b=pas*(2*ran2(idum)-1)

        r1b=sqrt(x1b**2+y1b**2+z1b**2)
        r2b=sqrt(x2b**2+y2b**2+z2b**2)
        distb=sqrt((x1b-x2b)**2+(y1b-y2b)**2+(z1b-z2b)**2)

        p1=prob(r1b,r2b,distb)
```

```
      w=ran2(idum);

      IF(p1/p0>sec*w) THEN
          cont=cont+1
          E0=Ene(distb,r1b,r2b,x1b,x2b,y1b,y2b,z1b,z2b)
          E20=E0**2
          Ea0=Enea(distb,r1b,r2b,x1b,x2b,y1b,y2b,z1b,z2b)
          Eb0=Eneb(distb,r1b,r2b,x1b,x2b,y1b,y2b,z1b,z2b)
          Fa0=Funa(distb)
          Fb0=Funb(distb)
          EFa0=E0*Fa0
          EFb0=E0*Fb0
          Fab0=Funab(distb)
          Faa0=Funaa(distb)
          Fbb0=Funbb(distb)
          EFab0=E0*Fab0
          EFaa0=E0*Faa0
          EFbb0=E0*Fbb0
          FaFb0=Fa0*Fb0
          FaFa0=Fa0*Fa0
          FbFb0=Fb0*Fb0
          EFaFb0=E0*Fa0*Fb0
          EFaFa0=E0*Fa0*Fa0
          EFbFb0=E0*Fb0*Fb0
          FaEb0=Fa0*Eb0
          FbEa0=Fb0*Ea0
          FaEa0=Fa0*Ea0
          FbEb0=Fb0*Eb0

          E=E+E0
          E2=E2+E20
          Ea=Ea+Ea0
          Eb=Eb+Eb0
          Fa=Fa+Fa0
          Fb=Fb+Fb0
          EFa=EFa+EFa0
          EFb=EFb+EFb0
          Fab=Fab+Fab0
          Faa=Faa+Faa0
          Fbb=Fbb+Fbb0
          EFab=EFab+EFab0
          EFaa=EFaa+EFaa0
          EFbb=EFbb+EFbb0
          FaFb=FaFb+FaFb0
          FaFa=FaFa+FaFa0
          FbFb=FbFb+FbFb0
          EFaFb=EFaFb+EFaFb0
          EFaFa=EFaFa+EFaFa0
          EFbFb=EFbFb+EFbFb0
          FaEb=FaEb+FaEb0
          FbEa=FbEa+FbEa0
          FaEa=FaEa+FaEa0
          FbEb=FbEb+FbEb0

          p0=p1

      ENDIF

  enddo

  alpha0=alpha
  beta0=beta

  E=E/cont
  E2=E2/cont
  Ea=Ea/cont
  Eb=Eb/cont
  Fa=Fa/cont
  Fb=Fb/cont
  EFa=EFa/cont
  EFb=EFb/cont
  Fab=Fab/cont
```

```
    Faa=Faa/cont
    Fbb=Fbb/cont
    EFab=EFab/cont
    EFaa=EFaa/cont
    EFbb=EFbb/cont
    FaFb=FaFb/cont
    FaFa=FaFa/cont
    FbFb=FbFb/cont
    EFaFb=EFaFb/cont
    EFaFa=EFaFa/cont
    EFbFb=EFbFb/cont
    FaEb=FaEb/cont
    FbEa=FbEa/cont
    FaEa=FaEa/cont
    FbEb=FbEb/cont


    ene1=E
    ene2=E2
    grad(1)=2.d0*(EFa-E*Fa)
    grad(2)=2.d0*(EFb-E*Fb)

    auxa=FaEa-Fa*Ea
    auxb=FbEb-Fb*Eb
    hess(1,1)=2.d0*(EFaa-E*Faa+2.d0*(EFaFa-E*FaFa)-Fa*grad(1)*2+auxa)
    hess(2,2)=2.d0*(EFbb-E*Fbb+2.d0*(EFbFb-E*FbFb)-Fb*grad(2)*2+auxb)
    aux=FaEb-Fa*Eb+FbEa-Fb*Ea
    hess(1,2)=2.d0*(EFab-E*Fab+2.d0*(EFaFb-E*FaFb)-Fa*grad(2)-Fb*grad(1))+aux
    hess(2,1)=2.d0*(EFab-E*Fab+2.d0*(EFaFb-E*FaFb)-Fb*grad(1)-Fa*grad(2))+aux

    det=hess(1,1)*hess(2,2)-hess(1,2)*hess(2,1)

!   write(*,*) "Hessia = ",hess(1,1),hess(1,2),hess(2,1),hess(2,2)
    write(*,*) "det = ",det

    hessm1(1,1)=hess(2,2)/det
    hessm1(2,2)=hess(1,1)/det
    hessm1(1,2)=-hess(2,1)/det
    hessm1(2,1)=-hess(1,2)/det

!   write(*,*) "det = ",det
!   write(*,*) "Hessià = ",hess(1,1),hess(1,2),hess(2,1),hess(2,2)
!   write(*,*) " gradient ", grad(1:2)
    write(40,*) "energia ", E," variancia  ", E2-E**2
!   write(40,*) "gradient ", grad(1:2)
!   write(40,*) "Hessia = ",hess(1,1),hess(1,2),hess(2,1),hess(2,2)
!   write(40,*) E," ",E2," ",Ea," ",Eb," ",Fa," ",Fb," ",EEa," ",EEb," ",EFa," ",EFb,"
",EaEb," ",Ea2," ",Eb2

    alpha=alpha0-damping*(hessm1(1,1)*grad(1)+hessm1(1,2)*grad(2))
    beta=beta0-damping*(hessm1(2,1)*grad(1)+hessm1(2,2)*grad(2))

!   write(40,*) "alpha0 = ",alpha0,"  beta0 = ",beta0
    write(41,*) "alpha = ",alpha,"  beta = ",beta
!   write(*,*) "E = ",ene1,"  var = ",ene2-ene1**2
!   write(*,*) "punts encertats ",cont,"   punts fallits ",m-cont;


return
end
```