```fortran
!234567890
!       Written by In-Ho Lee, KRISS, April 29, 2016.
      module elemdist
      implicit none
      private
      save
      integer ndeg,natot,nspecies,nstrc,ndim
      real*8 rmax,rc1,rc2
      logical lpbc
      integer, allocatable :: nelements(:)
      real*8, allocatable :: hist1(:),histsave1(:,:),histtest1(:),qext(:)
      real*8, allocatable :: hist2(:),histsave2(:,:),histtest2(:)
      public :: elemdist_init,elemdist_final,get_hist,elemdist_cmp
      contains
!234567890
!       Written by In-Ho Lee, KRISS, April 29, 2016.
      subroutine elemdist_init(rmax0,nspecies0,nelements0,rc10,rc20,nstrc0,lpbc
0)
      implicit none
      integer nspecies0,nelements0(nspecies0),nstrc0
      real*8 rmax0,rc10,rc20
      logical lpbc0
      integer i

      lpbc=lpbc0 ; rmax=rmax0
      if(rmax <= 0.d0) rmax=10.d0
      rc1=rc10 ; rc2=rc20
      nstrc=nstrc0
      nspecies=nspecies0
      allocate(nelements(nspecies))
      do i=1,nspecies
      nelements(i)=nelements0(i)
      enddo
      natot=sum(nelements) ; ndeg=3*natot+6
      ndim=natot*8
      allocate(hist1(0:ndim),histsave1(0:ndim,nstrc),histtest1(0:ndim))
      allocate(hist2(0:ndim),histsave2(0:ndim,nstrc),histtest2(0:ndim))
      end subroutine elemdist_init
!234567890
!       Written by In-Ho Lee, KRISS, April 29, 2016.
      subroutine elemdist_final(kmd,avg,sig)
      implicit none
      integer kmd
      real*8 avg,sig
      integer i,j
      character*280 fname
      real*8, allocatable :: wmat(:,:)

      if(kmd >= 1)then
      allocate(wmat(nstrc,nstrc)) ; wmat=0.d0
      do i=1,nstrc
      do j=1,nstrc
      if(j > i)then
      call elemdist_cmp(i,j,wmat(i,j)) ; wmat(j,i)=wmat(i,j)
              endif
      enddo
      enddo
      call gen_avgsig(avg,sig,nstrc,wmat)
      if(kmd > 1)then
      fname='mat1.dat'          ; call plotdiff(fname,nstrc,wmat)
      fname='mat1_hist.dat'     ; call stats(fname,nstrc,wmat)
              endif
      deallocate(wmat)
              endif
      if(allocated(nelements)) deallocate(nelements)
      if(allocated(hist1)) deallocate(hist1)
      if(allocated(hist2)) deallocate(hist2)
      if(allocated(histsave1)) deallocate(histsave1)
      if(allocated(histsave2)) deallocate(histsave2)
```

```fortran
      if(allocated(histtest1)) deallocate(histtest1)
      if(allocated(histtest2)) deallocate(histtest2)
      end subroutine elemdist_final
!234567890
!       Written by In-Ho Lee, KRISS, April 29, 2016.
      subroutine get_hist(iisv,qqq)
      implicit none
      integer iisv
      real*8 qqq(ndeg)
      integer kk1,kk2,ish,n1,n2,n3,natotext,m,i,j,k
      real*8 r,x,y,z,d1,d2,d3,t6(6),a1(3),a2(3),a3(3),cmatrix(3,3),aa1(3),aa2(3
),aa3(3)

      hist1(:)=0.0d0 ; hist2(:)=0.0d0
      if(lpbc)then
      ish=ndeg-6
      do i=1,6
      t6(i)=qqq(ish+i)
      enddo
      call latmat(t6,cmatrix,1)
      a1(:)=cmatrix(1,:) ; a2(:)=cmatrix(2,:) ; a3(:)=cmatrix(3,:)
!     call get_extension(a1,a2,a3,rmax,ncext)
!     ncext=ncext*2
!     n1=ncext(1) ; n2=ncext(2) ; n3=ncext(3)
      n1=2 ; n2=2 ; n3=2
      j=(n1*n2*n3)*natot
      allocate(qext(3*j+6))
      aa1=a1*dble(n1) ; aa2=a2*dble(n2) ; aa3=a3*dble(n3)
      natotext=0
      do m=1,natot
      do i=0,n1-1
      do j=0,n2-1
      do k=0,n3-1
      natotext=natotext+1
      qext(3*(natotext-1)+1)=qqq(3*(m-1)+1)+dble(i)
      qext(3*(natotext-1)+2)=qqq(3*(m-1)+2)+dble(j)
      qext(3*(natotext-1)+3)=qqq(3*(m-1)+3)+dble(k)
      enddo
      enddo
      enddo
      enddo
      do i=1,natotext
      qext(3*(i-1)+1)=qext(3*(i-1)+1)/dble(n1)
      qext(3*(i-1)+2)=qext(3*(i-1)+2)/dble(n2)
      qext(3*(i-1)+3)=qext(3*(i-1)+3)/dble(n3)
      enddo
      do i=1,natotext
      kk1=0 ; kk2=0
      do j=1,natotext
      if(j == i) cycle
      d1=qext(3*(i-1)+1)-qext(3*(j-1)+1)
      d2=qext(3*(i-1)+2)-qext(3*(j-1)+2)
      d3=qext(3*(i-1)+3)-qext(3*(j-1)+3)
      d1=d1-anint(d1)
      d2=d2-anint(d2)
      d3=d3-anint(d3)
      x=d1*aa1(1)+d2*aa2(1)+d3*aa3(1)
      y=d1*aa1(2)+d2*aa2(2)+d3*aa3(2)
      z=d1*aa1(3)+d2*aa2(3)+d3*aa3(3)
      r=sqrt(x*x+y*y+z*z)
      if( r <= rc1) then
      kk1=kk1+1
              endif
      if( r > rc1 .and. r <= rc2) then
      kk2=kk2+1
                      endif
      enddo
      if(kk1 > -1 .and. kk1 <= ndim) hist1(kk1)=hist1(kk1)+1.0d0
      if(kk2 > -1 .and. kk2 <= ndim) hist2(kk2)=hist2(kk2)+1.0d0
```

```fortran
        enddo
        deallocate(qext)
              endif
        if(.not. lpbc)then
        do i=1,natot
        kk1=0 ; kk2=0
        do j=1,natot
        if(j == i) cycle
        x=qqq(3*(i-1)+1)-qqq(3*(j-1)+1)
        y=qqq(3*(i-1)+2)-qqq(3*(j-1)+2)
        z=qqq(3*(i-1)+3)-qqq(3*(j-1)+3)
        r=sqrt(x*x+y*y+z*z)
        if( r <= rc1) then
        kk1=kk1+1
              endif
        if( r > rc1 .and. r <= rc2) then
        kk2=kk2+1
                endif
        enddo
        if(kk1 > -1 .and. kk1 <= ndim) hist1(kk1)=hist1(kk1)+1.0d0
        if(kk2 > -1 .and. kk2 <= ndim) hist2(kk2)=hist2(kk2)+1.0d0
        enddo
                endif
        if(iisv /= 0)then
        histsave1(:,iisv)=hist1(:)
        histsave2(:,iisv)=hist2(:)
              endif
        end subroutine get_hist
!234567890
!       Written by In-Ho Lee, KRISS, April 29, 2016.
        subroutine elemdist_cmp(ii,jj,dista)
        implicit none
        integer ii,jj
        real*8 dista
        integer k

        if(ii == 0 .and. jj /= 0)then
        histtest1(:)=histsave1(:,jj)-hist1(:) ; histtest2(:)=histsave2(:,jj)-hist
2(:)
                endif
        if(ii /= 0 .and. jj == 0)then
        histtest1(:)=hist1(:)-histsave1(:,ii) ; histtest2(:)=hist2(:)-histsave2(:
,ii)
                endif
        if(ii /= 0 .and. jj /= 0)then
        histtest1(:)=histsave1(:,jj)-histsave1(:,ii) ; histtest2(:)=histsave2(:,j
j)-histsave2(:,ii)
                endif

        dista=0.d0
        do k=0,natot
        dista=dista+dble(k)*(2.d0*abs(histtest1(k))+abs(histtest2(k)))
        enddo
        end subroutine elemdist_cmp

        end module elemdist
!234567890
!       Written by In-Ho Lee, KRISS, September 11, 2013.
        module bldist
        implicit none
        private
        save
        integer natot,ndeg,natotext1,natotext2,i222,ndim,nstrc,jdiffcode
        real*8 rmax
        logical lpbc
        real*8, allocatable :: qext(:),wrk44(:)
        real*8, allocatable :: blsave(:,:),bltest(:),blsrtd(:)
        integer, allocatable :: iwrk44(:)
        public :: bldist_init,bldist_final,get_blsrtd,bldist_cmp
```

```fortran
        contains
!234567890
!       Written by In-Ho Lee, KRISS, September 11, 2013.
        subroutine bldist_init(rmax0,jdiff0,i2220,nelements0,nspecies0,nstrc0,lpb
c0)
        implicit none
        integer jdiff0,nstrc0,i2220,nspecies0,nelements0(nspecies0)
        real*8 rmax0
        logical lpbc0
        integer j

        jdiffcode=jdiff0
        i222=i2220 ; rmax=rmax0 ; lpbc=lpbc0 ; nstrc=nstrc0
        if(rmax <= 0.d0) rmax=10.d0
        natot=sum(nelements0) ; ndeg=6+3*natot
        j=27*natot
        ndim=(j*(j-1))/2
        allocate(blsave(ndim,nstrc))
        allocate(bltest(ndim)) ; allocate(blsrtd(ndim))
        allocate(wrk44(ndim)) ; allocate(iwrk44(ndim))
        end subroutine bldist_init
!234567890
!       Written by In-Ho Lee, KRISS, September 11, 2013.
        subroutine bldist_final(kmd,avg,sig)
        implicit none
        integer kmd
        real*8 avg,sig
        integer i,j
        character*280 fname
        real*8, allocatable :: wmat(:,:)

        if(kmd >= 1)then
        allocate(wmat(nstrc,nstrc)) ; wmat=0.d0
        do i=1,nstrc
        do j=1,nstrc
        if(j > i)then
        call bldist_cmp(i,j,wmat(i,j)) ; wmat(j,i)=wmat(i,j)
                endif
        enddo
        enddo
        call gen_avgsig(avg,sig,nstrc,wmat)
        if(kmd > 1)then
        if(i222 == 1)then
        fname='mat2.dat'        ; call plotdiff(fname,nstrc,wmat)
        fname='mat2_hist.dat'   ; call stats(fname,nstrc,wmat)
                endif
        if(i222 == 2)then
        fname='mat3.dat'        ; call plotdiff(fname,nstrc,wmat)
        fname='mat3_hist.dat'   ; call stats(fname,nstrc,wmat)
                endif
        deallocate(wmat)
                endif
        if(allocated(wrk44)) deallocate(wrk44)
        if(allocated(iwrk44)) deallocate(iwrk44)
        if(allocated(blsave)) deallocate(blsave)
        if(allocated(bltest)) deallocate(bltest)
        if(allocated(blsrtd)) deallocate(blsrtd)
        end subroutine bldist_final
!234567890
!       Written by In-Ho Lee, KRISS, September 11, 2013.
        subroutine bldist_cmp(ii,jj,dista)
        implicit none
        integer ii,jj
        real*8 dista
        integer k

        if(ii /= 0 .and. jj /= 0)then
        bltest(:)=blsave(:,jj)-blsave(:,ii)
```

```fortran
                              endif
        if(ii == 0 .and. jj /= 0)then
        bltest(:)=blsave(:,jj)-blsrtd(:)
                              endif
        if(ii /= 0 .and. jj == 0)then
        bltest(:)=blsrtd(:)-blsave(:,ii)
                              endif

        dista=0.d0
        do k=1,ndim
        dista=dista+abs(bltest(k))
        enddo
        dista=sqrt(dista)
        end subroutine bldist_cmp
!234567890
!     Written by In-Ho Lee, KRISS, September 11, 2013.
        subroutine get_blsrtd(iisv,qqq)
        implicit none
        integer iisv
        real*8 qqq(ndeg)
        real*8 r,x,y,z,d1,d2,d3
        real*8 aa1(3),aa2(3),aa3(3),t6(6),cmatrix(3,3),a1(3),a2(3),a3(3)
        integer ish,ncext(3),n1,n2,n3,natotext,ij,i,j,k,m

        if(lpbc)then
        do j=1,natot
        qqq(3*(j-1)+1)=qqq(3*(j-1)+1)-anint(qqq(3*(j-1)+1))
        qqq(3*(j-1)+2)=qqq(3*(j-1)+2)-anint(qqq(3*(j-1)+2))
        qqq(3*(j-1)+3)=qqq(3*(j-1)+3)-anint(qqq(3*(j-1)+3))
        if(qqq(3*(j-1)+1) < 0.d0) qqq(3*(j-1)+1)=qqq(3*(j-1)+1)+1.d0
        if(qqq(3*(j-1)+2) < 0.d0) qqq(3*(j-1)+2)=qqq(3*(j-1)+2)+1.d0
        if(qqq(3*(j-1)+3) < 0.d0) qqq(3*(j-1)+3)=qqq(3*(j-1)+3)+1.d0
        enddo
        ish=ndeg-6
        do i=1,6
        t6(i)=qqq(ish+i)
        enddo
        call latmat(t6,cmatrix,1)
        a1(:)=cmatrix(1,:) ; a2(:)=cmatrix(2,:) ; a3(:)=cmatrix(3,:)
        call get_extension(a1,a2,a3,rmax,ncext)
        ncext=ncext*2
        n1=ncext(1) ; n2=ncext(2) ; n3=ncext(3)
        if(i222 >  6)then
        n1=i222 ; n2=i222 ; n3=i222
                        endif
        if(i222 == 5)then
        n1=5 ; n2=5 ; n3=5
                        endif
        if(i222 == 4)then
        n1=4 ; n2=4 ; n3=4
                        endif
        if(i222 == 3)then
        n1=3 ; n2=3 ; n3=3
                        endif
        if(i222 == 2)then
        n1=2 ; n2=2 ; n3=2
                        endif
        if(i222 == 1)then
        n1=1 ; n2=1 ; n3=1
                        endif
        j=(n1*n2*n3)*natot
        allocate(qext(3*j+6))
        aa1=a1*dble(n1) ; aa2=a2*dble(n2) ; aa3=a3*dble(n3)
        natotext=0
        do m=1,natot
        do i=0,n1-1
        do j=0,n2-1
        do k=0,n3-1
        natotext=natotext+1
```

```fortran
        qext(3*(natotext-1)+1)=qqq(3*(m-1)+1)+dble(i)
        qext(3*(natotext-1)+2)=qqq(3*(m-1)+2)+dble(j)
        qext(3*(natotext-1)+3)=qqq(3*(m-1)+3)+dble(k)
        enddo
        enddo
        enddo
        enddo
        do i=1,natotext
        qext(3*(i-1)+1)=qext(3*(i-1)+1)/dble(n1)
        qext(3*(i-1)+2)=qext(3*(i-1)+2)/dble(n2)
        qext(3*(i-1)+3)=qext(3*(i-1)+3)/dble(n3)
        enddo
        ij=0
        do i=1,natotext-1
        do j=i+1,natotext
        d1=qext(3*(i-1)+1)-qext(3*(j-1)+1)
        d2=qext(3*(i-1)+2)-qext(3*(j-1)+2)
        d3=qext(3*(i-1)+3)-qext(3*(j-1)+3)
        d1=d1-anint(d1)
        d2=d2-anint(d2)
        d3=d3-anint(d3)
        x=d1*aa1(1)+d2*aa2(1)+d3*aa3(1)
        y=d1*aa1(2)+d2*aa2(2)+d3*aa3(2)
        z=d1*aa1(3)+d2*aa2(3)+d3*aa3(3)
        r=sqrt(x*x+y*y+z*z)
        if(r < rmax)then
        ij=ij+1
        if(ij <= ndim) blsrtd(ij)=r
                        endif
        enddo
        enddo
              endif
        if(.not. lpbc)then
        natotext=natot
        ij=0
        do i=1,natot-1
        do j=i+1,natot
        x=qqq(3*(i-1)+1)-qqq(3*(j-1)+1)
        y=qqq(3*(i-1)+2)-qqq(3*(j-1)+2)
        z=qqq(3*(i-1)+3)-qqq(3*(j-1)+3)
        r=sqrt(x*x+y*y+z*z)
        if(r < rmax)then
        ij=ij+1
        if(ij <= ndim) blsrtd(ij)=r
                        endif
        enddo
        enddo
                  endif
        ij=min(ij,ndim)
        do i=1,ij
        wrk44(i)=blsrtd(i)
        enddo
        call sortnr(ij,wrk44,iwrk44)
        do i=1,ij
        blsrtd(i)=wrk44(iwrk44(i))
        enddo
        if(jdiffcode == 1) r=0.d0
        if(jdiffcode == -1) r=blsrtd(ij)
        do i=ij+1,ndim
        blsrtd(i)=r
        enddo
        if(allocated(qext)) deallocate(qext)
        if(iisv /= 0)then
        blsave(:,iisv)=blsrtd(:)
                        endif
        end subroutine get_blsrtd

        end module bldist
!234567890
```

```fortran
!       Written by In-Ho Lee, KRISS, September 11, 2013.
        module qlabmod
        implicit none
        private
        save
        integer natot,nspecies,ndeg,nstrc
        real*8 rmax
        integer, allocatable :: itype(:),itypeext(:),ndum(:,:)
        real*8, allocatable :: qext(:),qlab(:,:,:),qlabtest(:,:,:),qlabsave(:,:,:
,:)
        real*8, allocatable :: sigmamatrix(:,:)
        complex*16, allocatable :: ctdangl(:,:,:,:)
        logical lpbc
        public :: qlab_init,qlab_final,get_qlab,qlab_cmp
        contains
!234567890
!       Written by In-Ho Lee, KRISS, September 11, 2013.
        subroutine qlab_init(rmax0,nspecies0,nelements0,sigmamatrix0,nstrc0,lpbc0
)
        implicit none
        integer nspecies0,nelements0(nspecies0),nstrc0
        real*8 sigmamatrix0(nspecies0,nspecies0),rmax0
        logical lpbc0
        integer i,j,k

        nstrc=nstrc0 ; rmax=rmax0 ; lpbc=lpbc0
        if(rmax <= 0.d0) rmax=10.d0
        nspecies=nspecies0
        allocate(ctdangl(nspecies,nspecies,0:10,-10:10))
        allocate(ndum(nspecies,nspecies))
        natot=sum(nelements0) ; ndeg=6+3*natot
        allocate(itype(natot))
        k=0
        do i=1,nspecies0
        do j=1,nelements0(i)
        k=k+1
        itype(k)=i
        enddo
        enddo
        if(k /= natot)then
        write(6,*) k,natot,'k,natot'
                    stop
                  endif
        allocate(sigmamatrix(nspecies,nspecies))
        sigmamatrix=sigmamatrix0
        allocate(qlab(nspecies,nspecies,0:10))
        allocate(qlabtest(nspecies,nspecies,0:10))
        allocate(qlabsave(nspecies,nspecies,0:10,nstrc))
        end subroutine qlab_init
!234567890
!       Written by In-Ho Lee, KRISS, September 11, 2013.
        subroutine qlab_final(kmd,avg,sig)
        implicit none
        integer kmd
        real*8 avg,sig
        integer i,j
        character*280 fname
        real*8, allocatable :: wmat(:,:)

        if(kmd >= 1)then
        allocate(wmat(nstrc,nstrc)) ; wmat=0.d0
        do i=1,nstrc
        do j=1,nstrc
        if(j > i)then
        call qlab_cmp(i,j,wmat(i,j))  ; wmat(j,i)=wmat(i,j)
                    endif
        enddo
        enddo
        call gen_avgsig(avg,sig,nstrc,wmat)
```

```fortran
        if(kmd > 1)then
        fname='mat4.dat'       ; call plotdiff(fname,nstrc,wmat)
        fname='mat4_hist.dat' ; call stats(fname,nstrc,wmat)
                endif
        deallocate(wmat)
                  endif
        if(allocated(sigmamatrix)) deallocate(sigmamatrix)
        if(allocated(itype)) deallocate(itype)
        if(allocated(ctdangl)) deallocate(ctdangl)
        if(allocated(ndum)) deallocate(ndum)
        if(allocated(qlab)) deallocate(qlab)
        if(allocated(qlabsave)) deallocate(qlabsave)
        if(allocated(qlabtest)) deallocate(qlabtest)
        end subroutine qlab_final
!234567890
!       Written by In-Ho Lee, KRISS, September 11, 2013.
        subroutine qlab_cmp(ii,jj,dista)
        implicit none
        integer ii,jj
        real*8 dista
        real*8 tmp,tmq
        integer i,j,l

        if(ii /= 0 .and. jj /= 0)then
        qlabtest(:,:,:)=qlabsave(:,:,:,jj)-qlabsave(:,:,:,ii)
                    endif
        if(ii == 0 .and. jj /= 0)then
        qlabtest(:,:,:)=qlabsave(:,:,:,jj)-qlab(:,:,:)
                    endif
        if(ii /= 0 .and. jj == 0)then
        qlabtest(:,:,:)=qlab(:,:,:)-qlabsave(:,:,:,ii)
                    endif

        tmq=0.d0
        dista=0.d0
        do i=1,nspecies
        do j=1,nspecies
        tmp=0.d0
        do l=0,10,2
        tmp=tmp+(qlabtest(i,j,l))**2
        enddo
        tmq=tmq+1.d0
        dista=dista+tmp
        enddo
        enddo
        dista=dista/tmq
        dista=sqrt(dista)
        end subroutine qlab_cmp
!234567890
!       Written by In-Ho Lee, KRISS, September 11, 2013.
        subroutine get_qlab(iisv,qqq)
        implicit none
        integer iisv
        real*8 qqq(ndeg)
        real*8 pi,x,y,z,r,d1,d2,d3,theta,phi,arg
        real*8 aa1(3),aa2(3),aa3(3),t6(6),cmatrix(3,3),a1(3),a2(3),a3(3)
        integer n1,n2,n3,i,j,k,l,m,nb,ish,iti,itj,ncext(3)
        complex*16 ylm,ctmp(0:10,-10:10),cl(0:10)

        ctdangl(:,:,:,:)=cmplx(0.d0,0.d0) ; ndum(:,:)=0
        if(      lpbc)then
        do j=1,natot
        qqq(3*(j-1)+1)=qqq(3*(j-1)+1)-anint(qqq(3*(j-1)+1))
        qqq(3*(j-1)+2)=qqq(3*(j-1)+2)-anint(qqq(3*(j-1)+2))
        qqq(3*(j-1)+3)=qqq(3*(j-1)+3)-anint(qqq(3*(j-1)+3))
        if(qqq(3*(j-1)+1) < 0.d0) qqq(3*(j-1)+1)=qqq(3*(j-1)+1)+1.d0
        if(qqq(3*(j-1)+2) < 0.d0) qqq(3*(j-1)+2)=qqq(3*(j-1)+2)+1.d0
        if(qqq(3*(j-1)+3) < 0.d0) qqq(3*(j-1)+3)=qqq(3*(j-1)+3)+1.d0
        enddo
```

```
      ish=ndeg-6
      do i=1,6
      t6(i)=qqq(ish+i)
      enddo
      call latmat(t6,cmatrix,1)
      a1(:)=cmatrix(1,:) ; a2(:)=cmatrix(2,:) ; a3(:)=cmatrix(3,:)
      call get_extension(a1,a2,a3,rmax,ncext)
      ncext=ncext*2
      n1=ncext(1) ; n2=ncext(2) ; n3=ncext(3)
      nb=(n1*n2*n3)*natot
      allocate(itypeext(nb))
      allocate(qext(3*nb+6))
      aa1=a1*dble(n1) ; aa2=a2*dble(n2) ; aa3=a3*dble(n3)
      nb=0
      do m=1,natot
      do i=0,n1-1
      do j=0,n2-1
      do k=0,n3-1
      nb=nb+1
      itypeext(nb)=itype(m)
      qext(3*(nb-1)+1)=qqq(3*(m-1)+1)+dble(i)
      qext(3*(nb-1)+2)=qqq(3*(m-1)+2)+dble(j)
      qext(3*(nb-1)+3)=qqq(3*(m-1)+3)+dble(k)
      enddo
      enddo
      enddo
      enddo
      do i=1,nb
      qext(3*(i-1)+1)=qext(3*(i-1)+1)/dble(n1)
      qext(3*(i-1)+2)=qext(3*(i-1)+2)/dble(n2)
      qext(3*(i-1)+3)=qext(3*(i-1)+3)/dble(n3)
      enddo
      do i=1,nb
      iti=itypeext(i)
      do j=1,nb
      if(j == i) cycle
      itj=itypeext(j)
      d1=qext(3*(i-1)+1)-qext(3*(j-1)+1)
      d2=qext(3*(i-1)+2)-qext(3*(j-1)+2)
      d3=qext(3*(i-1)+3)-qext(3*(j-1)+3)
      d1=d1-anint(d1)
      d2=d2-anint(d2)
      d3=d3-anint(d3)
      x=d1*aa1(1)+d2*aa2(1)+d3*aa3(1)
      y=d1*aa1(2)+d2*aa2(2)+d3*aa3(2)
      z=d1*aa1(3)+d2*aa2(3)+d3*aa3(3)
      r=sqrt(x*x+y*y+z*z)
      if(r > 6.d0*sigmamatrix(iti,itj)) cycle
      call xyz2rtp(x,y,z,r,theta,phi)
      do l=0,10,2
      do m=-l,l
      call sphhar(l,m,theta,phi,ylm)
      arg=-(r-2.d0*sigmamatrix(iti,itj))/2.d0
      if(arg < -50.d0) arg=-50.d0 ; if(arg >  50.d0) arg= 50.d0
      ctdangl(iti,itj,l,m)=ctdangl(iti,itj,l,m)+ylm*exp(arg)
      enddo
      enddo
      ndum(iti,itj)=ndum(iti,itj)+1
      enddo
      enddo
                  endif
      if(.not. lpbc)then
      do i=1,natot
      iti=itype(i)
      do j=1,natot
      if(j == i) cycle
      itj=itype(j)
      x=qqq(3*(i-1)+1)-qqq(3*(j-1)+1)
      y=qqq(3*(i-1)+2)-qqq(3*(j-1)+2)
```

```
      z=qqq(3*(i-1)+3)-qqq(3*(j-1)+3)
      r=sqrt(x*x+y*y+z*z)
      if(r > 6.d0*sigmamatrix(iti,itj)) cycle
      call xyz2rtp(x,y,z,r,theta,phi)
      do l=0,10,2
      do m=-l,l
      call sphhar(l,m,theta,phi,ylm)
      arg=-(r-2.d0*sigmamatrix(iti,itj))/2.d0
      if(arg < -50.d0) arg=-50.d0 ; if(arg >  50.d0) arg= 50.d0
      ctdangl(iti,itj,l,m)=ctdangl(iti,itj,l,m)+ylm*exp(arg)
      enddo
      enddo
      ndum(iti,itj)=ndum(iti,itj)+1
      enddo
      enddo
                  endif
      do i=1,nspecies
      do j=1,nspecies
      if(ndum(i,j) > 0) ctdangl(i,j,:,:)=ctdangl(i,j,:,:)/dble(ndum(i,j))
      enddo
      enddo
!
      pi=4.d0*atan(1.d0)
      qlab(:,:,:)=0.d0
      do i=1,nspecies
      do j=1,nspecies
      ctmp(:,:)=ctdangl(i,j,:,:)
      do l=0,10,2
      cl=cmplx(0.d0,0.d0)
      do m=-l,l
      cl(l)=cl(l)+ctmp(l,m)*conjg(ctmp(l,m))
      enddo
      cl(l)=sqrt((4.d0*pi)*cl(l)/(2.d0*dble(l)+1.d0))
      qlab(i,j,l)=real(cl(l))
      enddo
      enddo
      enddo
      if(allocated(itypeext)) deallocate(itypeext)
      if(allocated(qext)) deallocate(qext)
      if(iisv /= 0)then
      qlabsave(:,:,:,iisv)=qlab(:,:,:)
                  endif
      end subroutine get_qlab

      end module qlabmod
!234567890
!     Written by In-Ho Lee, KRISS, April 15, 2016.
      module prdf
      implicit none
      private
      save
      integer nr,ndim,nstrc,ndeg,natot,kosine
      real*8 r0,r1,dr,rmax
      real*8, allocatable :: prdf0(:,:),prdfsave(:,:,:),prdftest(:,:)
      integer, allocatable :: irow(:,:)
      integer nspecies
      integer, allocatable :: nelements(:)
      character*2, allocatable :: symbl(:)
      logical lpbc
      public :: prdf_init,prdf_final,get_prdf,prdf_cmp,stepft
      contains
!234567890
!     Written by In-Ho Lee, KRISS, April 15, 2016.
      subroutine prdf_init(rmax0,nspecies0,nelements0,symbl0,nstrc0,lpbc0)
      implicit none
      integer nstrc0,nspecies0,nelements0(nspecies0)
      real*8 rmax0
      character*2 symbl0(nspecies0)
      logical lpbc0
```

```fortran
      integer i,j,jprint

      kosine=0
      if(nstrc0 < 0)then
      nstrc=iabs(nstrc0)
      kosine=1
                endif
      lpbc=lpbc0 ; nstrc=nstrc0
      nspecies=nspecies0
      allocate(nelements(nspecies)) ; allocate(symbl(nspecies))
      allocate(irow(nspecies,nspecies))
      nelements=nelements0
      natot=sum(nelements) ; ndeg=3*natot+6
      ndim=0
      do i=1,nspecies
      do j=1,nspecies
      ndim=ndim+1
      irow(i,j)=ndim
      enddo
      enddo
      do i=1,nspecies
      symbl(i)=trim(adjustl(symbl0(i)))
      nelements(i)=nelements0(i)
      enddo
      if(rmax0 > 10.d0)then
      write(6,*) rmax0,' has been changed to',10.d0
      rmax0=10.d0
                endif
      rmax=rmax0
      if(rmax <= 0.d0) rmax=10.d0
!     nr=1001 ; r0=0.0d0 ; r1=rmax*2.0d0 ; dr=(r1-r0)/dble(nr-1)
      nr=401  ; r0=0.0d0 ; r1=20.0d0    ; dr=(r1-r0)/dble(nr-1)
      jprint=1
      jprint=0
      if(jprint == 1)then
      write(6,'(2f18.8,1x,a7)') rmax,dr,'rmax,dr'
      write(6,'(3f18.8)') r0,r1,dr
      write(6,'(i5)') nr
                endif
      if(.not. allocated(prdf0)) allocate(prdf0(nr,ndim))
      if(.not. allocated(prdfsave)) allocate(prdfsave(nr,ndim,nstrc))
      if(.not. allocated(prdftest)) allocate(prdftest(nr,ndim))
      end subroutine prdf_init
!234567890
!     Written by In-Ho Lee, KRISS, April 15, 2016.
      subroutine prdf_final(kmd,avg,sig)
      implicit none
      integer kmd
      real*8 avg,sig
      integer i,j
      character*280 fname
      real*8, allocatable :: wmat(:,:)

      if(kmd >= 1)then
      allocate(wmat(nstrc,nstrc)) ; wmat=0.d0
      do i=1,nstrc
      do j=1,nstrc
      if(j > i)then
      call prdf_cmp(i,j,wmat(i,j))  ; wmat(j,i)=wmat(i,j)
                endif
      enddo
      enddo
      call gen_avgsig(avg,sig,nstrc,wmat)
      if(kmd > 1)then
      if(kosine == 1)then
      fname='mat6.dat'          ; call plotdiff(fname,nstrc,wmat)
      fname='mat6_hist.dat'     ; call stats(fname,nstrc,wmat)
                else
      fname='mat5.dat'          ; call plotdiff(fname,nstrc,wmat)
```

```fortran
      fname='mat5_hist.dat'     ; call stats(fname,nstrc,wmat)
                endif
                endif
      deallocate(wmat)
                endif
      if(allocated(nelements)) deallocate(nelements)
      if(allocated(symbl)) deallocate(symbl)
      if(allocated(irow)) deallocate(irow)
      if(allocated(prdf0)) deallocate(prdf0)
      if(allocated(prdfsave)) deallocate(prdfsave)
      if(allocated(prdftest)) deallocate(prdftest)
      end subroutine prdf_final
!234567890
!     Written by In-Ho Lee, KRISS, April 15, 2016.
      subroutine prdf_cmp(ii,jj,v)
      implicit none
      integer ii,jj
      real*8 v
      integer ir
      real*8 rr

      if(kosine == 1)then
      call prdf_cmp0(ii,jj,v)
                return
                endif

      if(ii /= 0 .and. jj == 0)then
      prdftest(:,:)=prdf0(:,:)-prdfsave(:,:,ii)
                endif
      if(ii == 0 .and. jj /= 0)then
      prdftest(:,:)=prdfsave(:,:,jj)-prdf0(:,:)
                endif
      if(ii /= 0 .and. jj /= 0)then
      prdftest(:,:)=prdfsave(:,:,jj)-prdfsave(:,:,ii)
                endif
      do ir=1,nr
      rr=r0+dr*float(ir-1)
      if(rr > rmax) prdftest(ir,:)=0.d0
      enddo
      call frobeniusnorm(nr,ndim,prdftest,v)
!     write(6,*) v
      end subroutine prdf_cmp
!234567890
!     Written by In-Ho Lee, KRISS, April 15, 2016.
      subroutine prdf_cmp0(ii,jj,v)
      implicit none
      integer ii,jj
      real*8 v
      integer ir,i1,i2,i3
      real*8 rr,tmp,tmq,tmr,tms

      tmp=0.0d0 ; tmq=0.0d0 ; tmr=0.0d0
      do ir=1,nr
      rr=r0+dr*float(ir-1)
      if(rr < rmax)then
      do i1=1,nspecies
      do i2=1,nspecies
      if(i2 > i1) cycle
      tms=dble(nelements(i1))*dble(nelements(i2))
      i3=irow(i1,i2)
      if(ii /= 0 .and. jj /= 0)then
      tmp=tmp+(prdfsave(ir,i3,jj)-1.d0)*(prdfsave(ir,i3,ii)-1.d0)*tms
      tmq=tmq+(prdfsave(ir,i3,ii)-1.d0)*(prdfsave(ir,i3,ii)-1.d0)*tms
      tmr=tmr+(prdfsave(ir,i3,jj)-1.d0)*(prdfsave(ir,i3,jj)-1.d0)*tms
                endif
      if(ii == 0 .and. jj /= 0)then
      tmp=tmp+(prdfsave(ir,i3,jj)-1.d0)*(prdf0(ir,i3)-1.d0)*tms
      tmq=tmq+(prdf0(ir,i3)-1.d0)*(prdf0(ir,i3)-1.d0)*tms
      tmr=tmr+(prdfsave(ir,i3,jj)-1.d0)*(prdfsave(ir,i3,jj)-1.d0)*tms
```

```fortran
                             endif
        if(ii /= 0 .and. jj == 0)then
        tmp=tmp+(prdf0(ir,i3)-1.d0)*(prdfsave(ir,i3,ii)-1.d0)*tms
        tmq=tmq+(prdfsave(ir,i3,ii)-1.d0)*(prdfsave(ir,i3,ii)-1.d0)*tms
        tmr=tmr+(prdf0(ir,i3)-1.d0)*(prdf0(ir,i3)-1.d0)*tms
                             endif
        enddo
        enddo
                     endif
        enddo
        tmq=sqrt(tmq) ; tmr=sqrt(tmr) ; v=0.5d0*(1.d0-tmp/(tmq*tmr))
        end subroutine prdf_cmp0
!234567890
!       Written by In-Ho Lee, KRISS, April 13, 2016.
        real*8 function stepft(x)
        implicit none
        real*8 x

        stepft=0.d0
        if(x >= 0.d0) stepft=1.d0
        end function stepft
!234567890
!       Written by In-Ho Lee, KRISS, April 15, 2016.
        subroutine frobeniusnorm(m,n,a,v)
        implicit none
        integer m,n
        real*8 a(m,n),v
        integer i,j

        v=0.d0
        do i=1,m
        do j=1,n
        v=v+(abs(a(i,j)))**2
        enddo
        enddo
        v=sqrt(v)
        end subroutine frobeniusnorm
!234567890
!       Written by In-Ho Lee, KRISS, April 13, 2016.
        subroutine get_prdf(iisv,qqq)
        implicit none
        integer iisv
        real*8 qqq(ndeg)
        integer, allocatable :: itype(:),nelementsext(:)
        real*8, allocatable :: dir(:,:),dirext(:,:)
        integer i1,i2,i3,j1,j2,k1,k2,ir,ip,iq,ks,ksext,ish,iprint,kprint
        integer i,j,k,ie,je,ke,k0,kk0,ncext(3)
        real*8 rr,pi,tmr,vtest,vtest0,vec(3),wec(3)
        real*8 a1(3),a2(3),a3(3),aa1(3),aa2(3),aa3(3),t6(6),cmatrix(3,3)

        pi=4.0d0*atan(1.0d0)
        if(lpbc)then
        do j=1,natot
        qqq(3*(j-1)+1)=qqq(3*(j-1)+1)-anint(qqq(3*(j-1)+1))
        qqq(3*(j-1)+2)=qqq(3*(j-1)+2)-anint(qqq(3*(j-1)+2))
        qqq(3*(j-1)+3)=qqq(3*(j-1)+3)-anint(qqq(3*(j-1)+3))
        if(qqq(3*(j-1)+1) < 0.d0) qqq(3*(j-1)+1)=qqq(3*(j-1)+1)+1.d0
        if(qqq(3*(j-1)+2) < 0.d0) qqq(3*(j-1)+2)=qqq(3*(j-1)+2)+1.d0
        if(qqq(3*(j-1)+3) < 0.d0) qqq(3*(j-1)+3)=qqq(3*(j-1)+3)+1.d0
        enddo
        ish=ndeg-6
        do i=1,6
        t6(i)=qqq(ish+i)
        enddo
        call latmat(t6,cmatrix,1)
        a1(:)=cmatrix(1,:) ; a2(:)=cmatrix(2,:) ; a3(:)=cmatrix(3,:)
        call get_extension(a1,a2,a3,rmax,ncext)
        ncext=ncext*2
        vtest=(cmatrix(1,2)*cmatrix(2,3)-cmatrix(1,3)*cmatrix(2,2))*cmatrix(3,1)
```

```fortran
      &
                +(cmatrix(1,3)*cmatrix(2,1)-cmatrix(1,1)*cmatrix(2,3))*cmatrix(3,2)
      &
                +(cmatrix(1,1)*cmatrix(2,2)-cmatrix(1,2)*cmatrix(2,1))*cmatrix(3,3)
        vtest0=abs(vtest)
        ks=sum(nelements)
        kprint=1
        kprint=0
        if(kprint == 1)then
        write(6,'(6f16.8,1x,f12.5)') t6(1),t6(2),t6(3),t6(4)*180.d0/pi,t6(5)*180.d0/pi
,t6(6)*180.d0/pi,vtest0
        write(6,*) ncext
                     endif
        kprint=1
        kprint=0
!
        allocate(dir(natot,3))
        do i=1,natot
        dir(i,1)=qqq(3*(i-1)+1) ; dir(i,2)=qqq(3*(i-1)+2) ; dir(i,3)=qqq(3*(i-1)+
3)
        enddo
        aa1=a1*dble(ncext(1)) ; aa2=a2*dble(ncext(2)) ; aa3=a3*dble(ncext(3))
        k=sum(nelements)*(ncext(1)*ncext(2)*ncext(3))
        allocate(nelementsext(nspecies)) ; allocate(itype(k))
        nelementsext(:)=nelements(:)*(ncext(1)*ncext(2)*ncext(3))
        ksext=sum(nelementsext)
!       write(6,*)k,ksext
        allocate(dirext(ksext,3))
        cmatrix(1,:)=aa1(:) ; cmatrix(2,:)=aa2(:) ; cmatrix(3,:)=aa3(:)
        vtest=(cmatrix(1,2)*cmatrix(2,3)-cmatrix(1,3)*cmatrix(2,2))*cmatrix(3,1)
      &
                +(cmatrix(1,3)*cmatrix(2,1)-cmatrix(1,1)*cmatrix(2,3))*cmatrix(3,2)
      &
                +(cmatrix(1,1)*cmatrix(2,2)-cmatrix(1,2)*cmatrix(2,1))*cmatrix(3,3)
        vtest=abs(vtest)
        k=0 ; kk0=0
        do i=1,nspecies
        k0=0
        do j=1,nelements(i)
        k0=k0+1 ; kk0=kk0+1
        do ie=0,ncext(1)-1
        do je=0,ncext(2)-1
        do ke=0,ncext(3)-1
        k=k+1
        dirext(k,1)=(dble(ie)+dir(kk0,1))/dble(ncext(1))
        dirext(k,2)=(dble(je)+dir(kk0,2))/dble(ncext(2))
        dirext(k,3)=(dble(ke)+dir(kk0,3))/dble(ncext(3))
        itype(k)=i
        enddo
        enddo
        enddo
        enddo
        if(k0 /= nelements(i))then
        write(6,*) 'error, k0',i
                     stop
                     endif
        enddo
        if(k /= ksext)then
        write(6,*) 'error ksext',k,ksext
                     stop
                     endif
        deallocate(dir)
        i3=0
        do i1=1,nspecies
        do i2=1,nspecies
        i3=i3+1
        if(ndim < i3)then
        write(6,*) 'check ndim',ndim
                     stop
```

```fortran
                  endif
      irow(i1,i2)=i3
      prdf0(:,i3)=0.d0
      enddo
      enddo
      k1=0
      do i1=1,nspecies
      do j1=1,nelementsext(i1)
      k1=k1+1
      if(k1 > ksext) stop
      k2=0
      do i2=1,nspecies
      i3=irow(i1,i2)
      do j2=1,nelementsext(i2)
      k2=k2+1
      if(k2 > ksext) stop
      if(k1 == k2) cycle
      if(itype(k1) == i1 .and. itype(k2) == i2)then
      vec(:)=dirext(k1,:)-dirext(k2,:)
      vec(1)=vec(1)-anint(vec(1))
      vec(2)=vec(2)-anint(vec(2))
      vec(3)=vec(3)-anint(vec(3))
      wec(:)=vec(1)*aa1(:)+vec(2)*aa2(:)+vec(3)*aa3(:)
      tmr=sqrt(dot_product(wec,wec))
      ir=(1+int(tmr/dr))-2 ; ip=max(ir,2) ; ir=(1+int(tmr/dr))+2 ; iq=min(ir,nr
-1)
      do ir=ip,iq
      rr=r0+dr*float(ir-1)
      prdf0(ir,i3)=prdf0(ir,i3)+stepft(tmr-rr)*stepft(rr+dr-tmr)*dble(ksext)/(d
ble(ksext)/vtest) &
                         /(dble(nelementsext(i1)))/(dble(nelementsext(i2)))
      enddo
                                  endif
      enddo
      enddo
      enddo
      enddo
      deallocate(dirext) ; deallocate(nelementsext) ; deallocate(itype)
            endif
      if(.not. lpbc)then
      ish=ndeg-6
      do i=1,6
      t6(i)=qqq(ish+i)
      enddo
      call latmat(t6,cmatrix,1)
      ks=sum(nelements)
      vtest=(cmatrix(1,2)*cmatrix(2,3)-cmatrix(1,3)*cmatrix(2,2))*cmatrix(3,1)
&
         +(cmatrix(1,3)*cmatrix(2,1)-cmatrix(1,1)*cmatrix(2,3))*cmatrix(3,2)
&
         +(cmatrix(1,1)*cmatrix(2,2)-cmatrix(1,2)*cmatrix(2,1))*cmatrix(3,3)
      vtest0=abs(vtest)
      allocate(itype(ks))
      k=0
      do i=1,nspecies
      do j=1,nelements(i)
      k=k+1
      itype(k)=i
      enddo
      enddo
      i3=0
      do i1=1,nspecies
      do i2=1,nspecies
      i3=i3+1
      if(ndim < i3)then
      write(6,*) 'check ndim',ndim
                  stop
                  endif
      irow(i1,i2)=i3
```

```fortran
      prdf0(:,i3)=0.d0
      enddo
      enddo
      k1=0
      do i1=1,nspecies
      do j1=1,nelements(i1)
      k1=k1+1
      if(k1 > ks) stop
      k2=0
      do i2=1,nspecies
      i3=irow(i1,i2)
      do j2=1,nelements(i2)
      k2=k2+1
      if(k2 > ks) stop
      if(k1 == k2) cycle
      if(itype(k1) == i1 .and. itype(k2) == i2)then
      wec(1)=qqq(3*(k1-1)+1)-qqq(3*(k2-1)+1)
      wec(2)=qqq(3*(k1-1)+2)-qqq(3*(k2-1)+2)
      wec(3)=qqq(3*(k1-1)+3)-qqq(3*(k2-1)+3)
      tmr=sqrt(dot_product(wec,wec))
      ir=(1+int(tmr/dr))-2 ; ip=max(ir,2) ; ir=(1+int(tmr/dr))+2 ; iq=min(ir,nr
-1)
      do ir=ip,iq
      rr=r0+dr*float(ir-1)
      prdf0(ir,i3)=prdf0(ir,i3)+stepft(tmr-rr)*stepft(rr+dr-tmr)*dble(ks)/(dble
(ks)/vtest0) &
                         /(dble(nelements(i1)))/(dble(nelements(i2)))
      enddo
                                  endif
      enddo
      enddo
      enddo
      enddo
      deallocate(itype)
            endif
      iprint=1
      iprint=0
      do i1=1,nspecies
      do i2=1,nspecies
      i3=irow(i1,i2)
      if(iprint == 1)then
      write(6,'(a1,1x,2i4)') '#', i1,i2
            endif
      do ir=2,nr
      rr=r0+dr*float(ir-1)
      prdf0(ir,i3)=prdf0(ir,i3)/(dr*4.0d0*pi*rr**2)
      if(iprint == 1)then
      if(rr <= rmax) write(6,'(f12.6,f22.12)') rr,prdf0(ir,i3)
            endif
      enddo
      if(iprint == 1)then
      write(6,*) '&'
            endif
      enddo
      enddo
      if(iisv /= 0)then
      prdfsave(:,:,iisv)=prdf0(:,:)
      endif
      end subroutine get_prdf

      end module prdf
!234567890
!     Written by In-Ho Lee, KRISS, April 13, 2016.
      subroutine plotdiff(fname,nstrc,zmat)
      implicit none
      character*80 fname
      integer nstrc
      real*8 zmat(nstrc,nstrc)
      integer i,j
```

```fortran
      open(11,file=trim(fname),form='formatted')
      do i=1,nstrc
      do j=1,nstrc
      write(11,'(2i5,1x,f20.8)') i,j,zmat(i,j)
      enddo
      write(11,*)
      enddo
      close(11)
      end
!234567890
!     Written by In-Ho Lee, KRISS, April 13, 2016.
      subroutine stats(fname,nstrc,rmat)
      USE prdf, ONLY : stepft
      implicit none
      character*80 fname
      integer nstrc
      real*8 rmat(nstrc,nstrc)
      integer i,j,ip,iq,ir,nr
      real*8 rr,r0,r1,dr,tmr,avg,sig,xnorm
      real*8 rmax
      real*8, allocatable :: histo(:)

      rmax=maxval(rmat)
      nr=501 ; r1=rmax*1.1d0 ; r0=0.0d0 ; dr=(r1-r0)/dble(nr-1)
      allocate(histo(nr)) ; histo=0.d0
      avg=0.d0
      xnorm=0.0d0
      do i=1,nstrc
      do j=1,nstrc
      if(j <= i) cycle
      tmr=rmat(i,j)
      avg=avg+tmr
      xnorm=xnorm+1.0d0
      ir=(1+int(tmr/dr))-2 ; ip=max(ir,2) ; ir=(1+int(tmr/dr))+2 ; iq=min(ir,nr
-1)
      do ir=ip,iq
      rr=r0+dr*float(ir-1)
      histo(ir)=histo(ir)+stepft(tmr-rr)*stepft(rr+dr-tmr)
      enddo
      enddo
      enddo
      avg=avg/xnorm
      sig=0.d0
      do i=1,nstrc
      do j=1,nstrc
      if(j <= i) cycle
      tmr=rmat(i,j)
      sig=sig+(tmr-avg)**2
      enddo
      enddo
      sig=sig/xnorm
      sig=sqrt(sig)
      open(11,file=trim(fname),form='formatted')
      write(11,'(a1,2x,2f22.10)') '#', avg,sig
      do ir=1,nr
      rr=r0+dr*float(ir-1)
      if(rr <= rmax) write(11,'(f12.6,f22.12)') rr,histo(ir)
      enddo
      close(11)
      deallocate(histo)
      end
!234567890
!     Written by In-Ho Lee, KRISS, April 13, 2016.
      subroutine gen_avgsig(avg,sig,nstrc,wmat)
      implicit none
      integer nstrc
      real*8 avg,sig
      real*8 wmat(nstrc,nstrc)
```

```fortran
      real*8 xnorm,tmr
      integer i,j

      avg=0.d0 ; xnorm=0.d0
      do i=1,nstrc
      do j=1,nstrc
      if(j <= i) cycle
      tmr=wmat(i,j)
      avg=avg+tmr
      xnorm=xnorm+1.d0
      enddo
      enddo
      avg=avg/xnorm ; sig=0.d0
      do i=1,nstrc
      do j=1,nstrc
      if(j <= i) cycle
      tmr=wmat(i,j)
      sig=sig+(tmr-avg)**2
      enddo
      enddo
      sig=sig/xnorm ; sig=sqrt(sig)
      end
!234567890
!     Written by In-Ho Lee, KRISS, April 13, 2016.
      subroutine get_extension(a1,a2,a3,rmax00,ncext)
      implicit none
      real*8 a1(3),a2(3),a3(3),rmax00
      integer ncext(3)
      real*8 v(3),h(3)

      call cross3(a1,a2,v) ; v=v/sqrt(sum(v*v)) ; h(3)=abs(sum(v*a3))
      call cross3(a3,a1,v) ; v=v/sqrt(sum(v*v)) ; h(2)=abs(sum(v*a2))
      call cross3(a2,a3,v) ; v=v/sqrt(sum(v*v)) ; h(1)=abs(sum(v*a1))
      v=rmax00/h+0.5d0
      ncext=nint(v)
      if(ncext(1) < 1) ncext(1)=1
      if(ncext(2) < 1) ncext(2)=1
      if(ncext(3) < 1) ncext(3)=1
      if(ncext(1) > 10) ncext(1)=10
      if(ncext(2) > 10) ncext(2)=10
      if(ncext(3) > 10) ncext(3)=10
      end
!234567890
!     Written by In-Ho Lee, KRISS, April 13, 2016.
      subroutine get_spbd(lpbc0,ndeg,qqq,test)
      implicit none
      logical lpbc0
      integer ndeg
      real*8 qqq(ndeg),test
      real*8 rmaxq0,r,x,y,z,d1,d2,d3,tmp,tmq,tmr
      real*8 aa1(3),aa2(3),aa3(3),t6(6),cmatrix(3,3),a1(3),a2(3),a3(3)
      integer ish,ncext(3),n1,n2,n3,natot0,natot0ext,ij,i,j,k,m,ndimq0,i222q0
      real*8, allocatable :: qext(:)
      real*8, allocatable :: blsrtdq(:)
      real*8, allocatable :: wrk45(:)
      integer, allocatable :: iwrk45(:)

      rmaxq0=2.d0
      i222q0=2
      ish=ndeg-6 ; natot0=ish/3
      j=27*natot0 ; ndimq0=(j*(j-1))/2
      if(.not. allocated(blsrtdq)) allocate(blsrtdq(ndimq0))
      if(lpbc0)then
      do j=1,natot0
      qqq(3*(j-1)+1)=qqq(3*(j-1)+1)-anint(qqq(3*(j-1)+1))
      qqq(3*(j-1)+2)=qqq(3*(j-1)+2)-anint(qqq(3*(j-1)+2))
      qqq(3*(j-1)+3)=qqq(3*(j-1)+3)-anint(qqq(3*(j-1)+3))
      if(qqq(3*(j-1)+1) < 0.d0) qqq(3*(j-1)+1)=qqq(3*(j-1)+1)+1.d0
      if(qqq(3*(j-1)+2) < 0.d0) qqq(3*(j-1)+2)=qqq(3*(j-1)+2)+1.d0
```

```fortran
      if(qqq(3*(j-1)+3) < 0.d0) qqq(3*(j-1)+3)=qqq(3*(j-1)+3)+1.d0
      enddo
      ish=ndeg-6
      do i=1,6
      t6(i)=qqq(ish+i)
      enddo
      call latmat(t6,cmatrix,1)
      a1(:)=cmatrix(1,:) ; a2(:)=cmatrix(2,:) ; a3(:)=cmatrix(3,:)
      call get_extension(a1,a2,a3,rmaxq0,ncext)
      ncext=ncext*2
      n1=ncext(1) ; n2=ncext(2) ; n3=ncext(3)
      if(i222q0 >  6)then
      n1=i222q0 ; n2=i222q0 ; n3=i222q0
                  endif
      if(i222q0 == 5)then
      n1=5 ; n2=5 ; n3=5
                  endif
      if(i222q0 == 4)then
      n1=4 ; n2=4 ; n3=4
                  endif
      if(i222q0 == 3)then
      n1=3 ; n2=3 ; n3=3
                  endif
      if(i222q0 == 2)then
      n1=2 ; n2=2 ; n3=2
                  endif
      if(i222q0 == 1)then
      n1=1 ; n2=1 ; n3=1
                  endif
      j=(n1*n2*n3)*natot0
      if(.not. allocated(qext)) allocate(qext(3*j+6))
      aa1=a1*dble(n1) ; aa2=a2*dble(n2) ; aa3=a3*dble(n3)
      natot0ext=0
      do m=1,natot0
      do i=0,n1-1
      do j=0,n2-1
      do k=0,n3-1
      natot0ext=natot0ext+1
      qext(3*(natot0ext-1)+1)=qqq(3*(m-1)+1)+dble(i)
      qext(3*(natot0ext-1)+2)=qqq(3*(m-1)+2)+dble(j)
      qext(3*(natot0ext-1)+3)=qqq(3*(m-1)+3)+dble(k)
      enddo
      enddo
      enddo
      enddo
      do i=1,natot0ext
      qext(3*(i-1)+1)=qext(3*(i-1)+1)/dble(n1)
      qext(3*(i-1)+2)=qext(3*(i-1)+2)/dble(n2)
      qext(3*(i-1)+3)=qext(3*(i-1)+3)/dble(n3)
      enddo
      ij=0
      do i=1,natot0ext-1
      do j=i+1,natot0ext
      d1=qext(3*(i-1)+1)-qext(3*(j-1)+1)
      d2=qext(3*(i-1)+2)-qext(3*(j-1)+2)
      d3=qext(3*(i-1)+3)-qext(3*(j-1)+3)
      d1=d1-anint(d1)
      d2=d2-anint(d2)
      d3=d3-anint(d3)
      x=d1*aa1(1)+d2*aa2(1)+d3*aa3(1)
      y=d1*aa1(2)+d2*aa2(2)+d3*aa3(2)
      z=d1*aa1(3)+d2*aa2(3)+d3*aa3(3)
      r=sqrt(x*x+y*y+z*z)
      if(r < rmaxq0)then
      ij=ij+1
      if(ij <= ndimq0) blsrtdq(ij)=r
                      endif
      enddo
      enddo
```

```fortran
         endif
      if(.not. lpbc0)then
      natot0ext=natot0
      ij=0
      do i=1,natot0-1
      do j=i+1,natot0
      x=qqq(3*(i-1)+1)-qqq(3*(j-1)+1)
      y=qqq(3*(i-1)+2)-qqq(3*(j-1)+2)
      z=qqq(3*(i-1)+3)-qqq(3*(j-1)+3)
      r=sqrt(x*x+y*y+z*z)
      if(r < rmaxq0)then
      ij=ij+1
      if(ij <= ndimq0) blsrtdq(ij)=r
                   endif
      enddo
      enddo
                   endif
      ij=min(ij,ndimq0)
      if(.not. allocated(iwrk45)) allocate(iwrk45(ij))
      if(.not. allocated(wrk45)) allocate(wrk45(ij))
      do i=1,ij
      wrk45(i)=blsrtdq(i)
      enddo
      call sortnr(ij,wrk45,iwrk45)
      do i=1,ij
      blsrtdq(i)=wrk45(iwrk45(i))
      enddo
      do i=ij+1,ndimq0
      blsrtdq(i)=0.d0
      enddo
      tmp=0.d0 ; tmq=0.d0
      do i=1,ij
      if(abs(blsrtdq(i)-1.42d0) < 0.059d0) tmp=tmp+1.d0
      if(abs(blsrtdq(i)-1.54d0) < 0.059d0) tmq=tmq+1.d0
      enddo
      tmr=tmp ; if(tmq > tmr) tmr=tmq ; tmr=tmr-dble(ij)/2.d0
!     test=abs(tmr)/dble(ij)
      test=test/dble(natot0)+abs(tmr)/dble(ij)
      if(allocated(qext)) deallocate(qext)
      if(allocated(blsrtdq)) deallocate(blsrtdq)
      if(allocated(iwrk45)) deallocate(iwrk45)
      if(allocated(wrk45)) deallocate(wrk45)
      end
!234567890
```