Adding independent pair correlations to code

Cody L. Petrie

June 11, 2015

1 Questions

1. Would you please check the red notes I have made and make sure that I am understanding those parts correctly?
2. This is the simplest way I could think of adding in the independent pair correlations. Is this too dumb of a way or is this what you were thinking?

2 Independent pair correlations

Currently the correlated trial wave function looks like

$$\Psi_T = \langle R,S| \left[\prod_{i<j} f_c(r_{ij})\right] \left[1+\sum_{i<j}\sum_p f_p(r_{ij})\mathcal{O}^p_{ij}\right] |\Phi\rangle \,. \tag{1}$$

The sum term is only the linear term of the correlation. I plan to add the next term. It is called the independent pair correlation term because the it includes the pairs that are summed over in the linear term but each of those is multiplied by a sum over all of the other terms that don't involve particles i or j. With the independent pair correlations the correlated trial wave function looks like this.

$$\Psi_T = \langle R,S| \left[\prod_{i<j} f_c(r_{ij})\right] \left[1+\sum_{i<j}\sum_p f_p(r_{ij})\mathcal{O}^p_{ij} + \sum_p\sum_{i<j}\sum_{\substack{k<l\\k,l\neq i,j}} f_p(r_{ij})\mathcal{O}^p_{ij}f_p(r_{ij})\mathcal{O}^p_{kl}\right] |\Phi\rangle \,. \tag{2}$$

2.1 Old and new code for correlation

From what I can tell these correlations are primarily handled in the subroutine "corpsi" in the file "correlatorcart.f90". I will use the Cartesian component version to simplify understanding. The black text represents code that is already in the program, red text is my notes to help me understand what the code is doing, and blue text represents pieces that I have added to include the independent pair terms, except I will leave the new subroutine black. The black is exactly what the code is in the version that I have right now.

```
ij=0
do i=1,npart-1
   do iop=1,15
      call sxzupdate(sxzi(:,:,:,iop),d15(iop),sxz0,i,sx15(:,iop,:,i),sp(:,i))
```

This piece does all 15 updates for the ith particle. This is precomputed in an outer loop since it will be used repeatedly for the 39 inverse updates for the jth particles.

```
   enddo
   do j=i+1,npart
      ij=ij+1
      if (doft(ij)) then
         do it=1,2
            fij=ft(ij)
            sx15j(:,:,:)=conjg(opmult(conjg(sxzi(:,j,:,3+it))))
            call sxzupdate(sxzj,d2,sxzi(:,:,:,3+it),j,sx15j(:,3+it,:) &
               ,sp(:,j))
```

This uses sxzi(:,:,:,4-5) as the old sxz so it’s doing the terms txi*txj tyi*tyj.

```
            detrat=d15(3+it)*d2
            fij=detrat*fij
            call g1bval(d1b,sxzj,fij)
            call g2bval(d2b,sxzj,fij)
            call g3bval(d3b,sxzj,fij)
            call corindpair(sp,sxzj,i,j,d1b,d2b,d3b)
         enddo
      endif
      if (doft(ij).or.doftpp(ij).or.doftnn(ij)) then
         it=3
         fij=ft(ij)
         if (doftpp(ij)) fij=fij+0.25_r8*ftpp(ij)
         if (doftnn(ij)) fij=fij+0.25_r8*ftnn(ij)
         sx15j(:,:,:)=conjg(opmult(conjg(sxzi(:,j,:,3+it))))
         call sxzupdate(sxzj,d2,sxzi(:,:,:,3+it),j,sx15j(:,3+it,:),sp(:,j))
```

This uses sxzi(:,:,:,6) as the old sxz so it’s doing tzi*tzj. Why is this done separately from tx and ty? I assume it as to do with doftpp and doftnn but I don’t know what those are.

```
         detrat=d15(3+it)*d2
         fij=detrat*fij
         call g1bval(d1b,sxzj,fij)
         call g2bval(d2b,sxzj,fij)
         call g3bval(d3b,sxzj,fij)
        call corindpair(sp,sxzj,i,j,d1b,d2b,d3b)
      endif
      if (dofs(ij)) then
         do is=1,3
            sx15j(:,:,:)=conjg(opmult(conjg(sxzi(:,j,:,is))))
            do js=1,3
               call sxzupdate(sxzj,d2,sxzi(:,:,:,is),j,sx15j(:,js,:),sp(:,j))
```

Here since the old sxz is sxz(:,:,:,is) where is and js both go from 1 to 3 this is doing the sxi*(sxj+syj+szj), syi*(sxj+syj+szj), and szi*(sxj+syj+szj) terms. There are 9 total here.

```
               detrat=d15(is)*d2
               fij=detrat*fs(is,js,ij)
               call g1bval(d1b,sxzj,fij)
               call g2bval(d2b,sxzj,fij)
               call g3bval(d3b,sxzj,fij)
               call corindpair(sp,sxzj,i,j,d1b,d2b,d3b)
            enddo
         enddo
      endif
      if (dofst(ij)) then
         do it=1,3
            do is=1,3
               sx15j(:,:,:)=conjg(opmult(conjg(sxzi(:,j,:,3*is+it+3))))
               do js=1,3
                  call sxzupdate(sxzj,d2,sxzi(:,:,:,3*is+it+3),j &
                     ,sx15j(:,3*js+it+3,:),sp(:,j))
```

Since the old sxz is sxzi(:,:,:,3*is+it+3) (giving 7,7,7,10,10,10,13,13,13,8,8,8, 11,11,11,14,14,14,9,9,9,12,12,12,15,15,15) and since we use sx15(3*js+it+3) (giving 7,10,13,8,11,14,9,12,15) we will get the 27 pairs
(sxi*txi+syi*txi+szi*txi)*(sxj*txj+syj*txj+szj*txj)),
(sxi*tyi+syi*tyi+szi*tyi)*(sxj*tyj+syj*tyj+szj*tyj)),
and (sxi*tzi+syi*tzi+szi*tzi)*(sxj*tzj+syj*tzj+szj*tzj)).

```
                  detrat=d15(3*is+it+3)*d2
                  fij=detrat*fst(is,js,ij)
                  call g1bval(d1b,sxzj,fij)
                  call g2bval(d2b,sxzj,fij)
                  call g3bval(d3b,sxzj,fij)
                  call corindpair(sp,sxzj,i,j,d1b,d2b,d3b)
               enddo
            enddo
         enddo
      endif
   enddo
enddo
```

Where the new subroutine is

```
subroutine corindpair(sp,sxz0,i,j,d1b,d2b,d3b)
complex(kind=r8), intent(in) :: sp(:,:)
complex(kind=r8), intent(inout) :: d1b(:,:),d2b(:,:,:),d3b(:,:,:,:)
complex(kind=r8), intent(in) :: sxz0(:,:,:)
integer(kind=i4), intent(in) :: i,j
complex(kind=r8) :: sxzk(4,npart,npart,15)
complex(kind=r8) :: fkl
complex(kind=r8) :: detrat
complex(kind=r8) :: sxzl(4,npart,npart),d2,d15(15)
complex(kind=r8) :: sx15(4,15,npart,npart),sx15l(4,15,npart)
integer(kind=i4) :: k,l,kl,kop,ks,kt,ls
kl=0
do k=1,npart-1
   if (k.eq.i .or. k.eq.j) cycle
   do kop=1,15
      call sxzupdate(sxzk(:,:,:,kop),d15(kop),sxz0,k,sx15(:,kop,:,k),sp(:,k))
   enddo
   do l=k+1,npart
      if (l.eq.i .or. l.eq.j) cycle
      kl=kl+1
      if (doft(kl)) then
         do kt=1,2
            fkl=ft(kl)
            sx15l(:,:,:)=conjg(opmult(conjg(sxzk(:,l,:,3+kt))))
            call sxzupdate(sxzl,d2,sxzk(:,:,:,3+kt),l,sx15l(:,3+kt,:) &
               ,sp(:,l))
            detrat=d15(3+kt)*d2
            fkl=detrat*fkl
            call g1bval(d1b,sxzl,fkl)
            call g2bval(d2b,sxzl,fkl)
            call g3bval(d3b,sxzl,fkl)
         enddo
      endif
      if (doft(kl).or.doftpp(kl).or.doftnn(kl)) then
         kt=3
         fkl=ft(kl)
         if (doftpp(kl)) fkl=fkl+0.25_r8*ftpp(kl)
         if (doftnn(kl)) fkl=fkl+0.25_r8*ftnn(kl)
         sx15l(:,:,:)=conjg(opmult(conjg(sxzk(:,l,:,3+kt))))
         call sxzupdate(sxzl,d2,sxzk(:,:,:,3+kt),l,sx15l(:,3+kt,:),sp(:,l))
         detrat=d15(3+kt)*d2
         fkl=detrat*fkl
         call g1bval(d1b,sxzl,fkl)
         call g2bval(d2b,sxzl,fkl)
         call g3bval(d3b,sxzl,fkl)
      endif
      if (dofs(kl)) then
         do ks=1,3
```

```
            sx15l(:,:,:)=conjg(opmult(conjg(sxzk(:,l,:,ks))))
            do ls=1,3
               call sxzupdate(sxzl,d2,sxzk(:,:,:,ks),l,sx15l(:,ls,:),sp(:,l))
               detrat=d15(ks)*d2
               fkl=detrat*fs(ks,ls,kl)
               call g1bval(d1b,sxzl,fkl)
               call g2bval(d2b,sxzl,fkl)
               call g3bval(d3b,sxzl,fkl)
            enddo
         enddo
      endif
      if (dofst(kl)) then
         do kt=1,3
            do ks=1,3
               sx15l(:,:,:)=conjg(opmult(conjg(sxzk(:,l,:,3*ks+kt+3))))
               do ls=1,3
                  call sxzupdate(sxzl,d2,sxzk(:,:,:,3*ks+kt+3),l &
                     ,sx15l(:,3*ls+kt+3,:),sp(:,l))
                  detrat=d15(3*ks+kt+3)*d2
                  fkl=detrat*fst(ks,ls,kl)
                  call g1bval(d1b,sxzl,fkl)
                  call g2bval(d2b,sxzl,fkl)
                  call g3bval(d3b,sxzl,fkl)
               enddo
            enddo
         enddo
      endif
   enddo
enddo
end subroutine corindpair
```

All I have done here is modified the code used above to put the $\sum_{i<j}\sum_{p} f_p(r_{ij})\mathcal{O}^p_{ij}$ correlation such that is did the same thing but for the k'th and l'th pairs. I also made it cycle if $k = i$ or $k = j$ or $l = i$ or $l = j$. Then just before moving on to the next correlation I do each independent pair correlation but starting with the already changed matrix sxzj where the i'th and j'th columns are changed.