Adding independent pair correlations to code

Cody L. Petrie

June 10, 2015

1 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.

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

1.1 Pseudo 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, while blue text represents pieces that need to be added to include the independent pair terms.

```
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
outerloop 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)
         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 i'ts doing tzi*tzj. Why is this done seperately 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)
      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)
            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))
                  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)
               enddo
            enddo
         enddo
      endif
   enddo
enddo
```