

Dissection of subroutine corpsi

Cody L. Petrie

June 8, 2015

The subroutine `corpsi` I believe stands for “correlate psi” where psi is the trial wavefunction. I will post the code and try to explain what each section above the comments means.

```
subroutine corpsi(sp,d1b,d2b,d3b)
```

Here the `d1b`, `d2b`, and `d3b` are used to get expectation values like as so

$$\langle \mathcal{O} \rangle = \sum_i \sum_s \alpha_{is} \mathbf{d1b}(\mathbf{s}, \mathbf{i}). \quad (1)$$

The other two are calculated in a similar manner. See my write up on the trial wave function to see details.

```
complex(kind=r8), intent(in) :: sp(:, :)
complex(kind=r8), intent(inout) :: d1b(:, :), d2b(:, :, :), d3b(:, :, :, :)
complex(kind=r8) :: fij, f1, fijk
complex(kind=r8) :: detrat, sxzi(4, npart, npart, 15)
complex(kind=r8) :: sxzj(4, npart, npart), d1, d2, d15(15)
complex(kind=r8) :: sx15(4, 15, npart, npart), sx15j(4, 15, npart)
complex(kind=r8) :: sx15j1(4, 15, npart), sx15j2(4, 15, npart)
complex(kind=r8) :: sx15k1(4, 15, npart), sx15k2(4, 15, npart)
complex(kind=r8) :: sxzi1(4, npart, npart), di1
complex(kind=r8) :: sxzj1(4, npart, npart), sxzj2(4, npart, npart)
complex(kind=r8) :: sxzk(4, npart, npart), dj1, dj2, dk
integer(kind=i4) :: i, j, ij, iop, is, it, js, k, ks, ijk
d1b=czero
d2b=czero
d3b=czero
call g1bval(d1b, sxz0, cone+fctau)
call g2bval(d2b, sxz0, cone+fctau)
call g3bval(d3b, sxz0, cone+fctau)
```

Most of this is just setup but `g1bval`, `g2bval`, and `g3bval` are subroutines used to calculate `d1b` etc. This part includes the determinant of the unchanged Slater matrix.

```

do i=1,npart
  sx15(:, :, :, i)=conjg(opmult(conjg(sxz0(:, i, :))))
enddo

```

The subroutine `opmult` operates on the spin components of a spinor with all 15 spin operators and returns the new spinor.

$$|s_i\rangle \rightarrow \mathcal{O}_i |s_i\rangle \quad (2)$$

Here the \mathcal{O}_i 's in order are 1-3 s_x, s_y, s_z , 4-6 t_x, t_y, t_z , 7-9 $s_x^*(t_x, t_y, t_z)$, 10-12 $s_y^*(t_x, t_y, t_z)$, 13-15 $s_z^*(t_x, t_y, t_z)$.

```

do i=1,npart
  if (abs(ftau1(i)).le.0.0_r8) cycle
  call sxzupdate(sxzj,d1,sxz0,i,sx15(:, 6, :, i),sp(:, i))
  f1=d1*0.25_r8*ftau1(i)
  call g1bval(d1b,sxzj,f1)
  call g2bval(d2b,sxzj,f1)
  call g3bval(d3b,sxzj,f1)
enddo

```

Here somekind of correlation is being added, it looks like a correlation related to the tau operators. `sxzupdate` is updating the inverse (`sxz`) and `g1bval` and the other two are adding on their respective determinants.

```

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

```

```

if (dofst(ij).or.dofstpp(ij).or.dofstnn(ij)) then
  it=3
  fij=ft(ij)
  if (dofstpp(ij)) fij=fij+0.25_r8*ftpp(ij)
  if (dofstnn(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))
  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))
      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
if (.not.dof3) return !skip 3-body correlation

```

```

do i=1,npart-2
  do is=1,3
    do it=1,3
      call sxzupdate(sxzi1,di1,sxz0,i,sx15(:,3*is+it+3,:),i),sp(:,i))
      do j=i+1,npart-1
        sx15j(:, :, :)=conjg(opmult(conjg(sxzi1(:,j, :))))
        do js=1,3
          call sxzupdate(sxzj1,dj1,sxzi1,j &
            ,sx15j(:,3*js+3+levi(1,it),:),sp(:,j))
          call sxzupdate(sxzj2,dj2,sxzi1,j &
            ,sx15j(:,3*js+3+levi(2,it),:),sp(:,j))
          do k=j+1,npart
!maple ijk := simplify(sum((n-1)*(n-1-1)/2,l=1..i-1)+sum(n-1,l=i+1..j-1)+k-j);
            ijk=(i*(i-1)*(i-3*npart+4))/6 &
              +((npart-2)*(npart-1)*(i-1))/2-2 &
              +((2*npart-4-j+1)*(j-2))/2+k
            sx15k1(:, :, :)=conjg(opmult(conjg(sxzj1(:,k, :))))
            sx15k2(:, :, :)=conjg(opmult(conjg(sxzj2(:,k, :))))
            do ks=1,3
              call sxzupdate(sxzk,dk,sxzj1,k &
                ,sx15k1(:,3*ks+3+levi(2,it),:),sp(:,k))
              fijk=f3(is,js,ks,ijk)*di1*dj1*dk
              call g1bval(d1b,sxzk,fijk)
              call g2bval(d2b,sxzk,fijk)
              call g3bval(d3b,sxzk,fijk)
              call sxzupdate(sxzk,dk,sxzj2,k, &
                sx15k2(:,3*ks+3+levi(1,it),:),sp(:,k))
              fijk=-f3(is,js,ks,ijk)*di1*dj1*dk
              call g1bval(d1b,sxzk,fijk)
              call g2bval(d2b,sxzk,fijk)
              call g3bval(d3b,sxzk,fijk)
            enddo
          enddo
        enddo
      enddo
    enddo
  enddo
enddo
end subroutine corpsi

```