From: Tom Rescigno <kingkohn@pacbell.net>
Subject: m928
Date: May 24, 2013 8:06:02 AM PDT
To: Robert Lucchese <lucchese@mail.chem.tamu.edu>
Reply-To: tnrescigno@lbl.gov

Bob,
I'm attaching a new version of subroutine press.f which is in the m928 link. You can now specify which of the valence orbitals to use in each channel when constructing Q-space vectors. I attached a sample H2O input. I ran an 8-channel problem, with 6 valence orbitals and picked 2 orbitals to use in each channel.
You put qkeep=(2,2,...) on the kohn line, which specifies how many orbitals to use in each channel. Then include a $qkeep section in the input which specifies the orbitals to use in each channel.
If qkeep in not on the kohn line, it works as before.

--
Thomas Rescigno
tnrescigno@lbl.gov
510-486-8652
http://devo.lbl.gov/~tnr

```
$title
h2o scf
$end

$route
hf scf=(pulay=convergence=6,cycles=250)
ksym=(symmetry=1,nsmall=8)
2s+1=1
drt=(nsym=4,ngroups=7,nrefs=16)
kohn=(nsmall=8,smallcq=.1,core=2,smallsq=.2,
qkeep=(2,2,2,2,2,2,2,2))
geom=(coord,inau)
print=(basis,m928)
ci=(nroots=8)
$end
$nonstd
1//1,2;
8//6;
9//28,90;
$end

$qkeep
1,4
2,6
2,6
3,5
1,4
3,5
1,4
1,4
$end

$drt
2typ1;1 1typ2;4 1typ2;1 1typ2;3 1typ3;1 1typ3;4 1typ3;3
1typ4;3 1typ5;4 1typ6;1 78typ7;1
na=5 nb=1 ns=3
$end

$groups
numel=(4,6,0,1,0,0,0)
numel=(4,5,1,1,0,0,0)
numel=(4,4,2,1,0,0,0)
numel=(4,3,3,1,0,0,0)
numel=(4,6,0,0,1,0,0)
numel=(4,5,1,0,1,0,0)
numel=(4,4,2,0,1,0,0)
numel=(4,3,3,0,1,0,0)
```

```
numel=(4,6,0,0,0,1,0)
numel=(4,5,1,0,0,1,0)
numel=(4,4,2,0,0,1,0)
numel=(4,3,3,0,0,1,0)
numel=(4,6,1,0,0,0,0)
numel=(4,5,2,0,0,0,0)
numel=(4,4,3,0,0,0,0)
numel=(4,3,4,0,0,0,0)
$end


$geom
o (basis=sc)   0.        0.       0.
h1 (basis=gil) 0.        1.43115  1.108113
h2 (basis=gil) 0.       -1.43115  1.108113
zqO(basis=bigo)   0.      0.       0.
zqH1(basis=bigh)  0.      1.43115  1.108113
zqH2(basis=bigh) 0.      -1.43115  1.108113
$end
$gil h
/basis from Gil et al PRA 49 2642 (1994)
/
type=s
74.69   0.025374
11.23   0.189684
2.546   0.852933
type=s
0.7130   1.0
type=s
0.2249   1.0
type=p
0.55    1.0
type=p
0.3    1.0
$end

$sc o
 type=s
      7816.5400     0.002031
      1175.8200     0.015436
       273.1880     0.073771
        81.1696     0.247606
        27.1836     0.611832
 type=s
         9.5322     1.0
 type=s
         3.4136     0.241205
 type=s
         0.9398     1.0
 type=s
         0.2846     1.0
 type=s
         0.095     1.0
  type=p
        35.1832     0.019580
         7.9040     0.124189
  type=p
         2.3051     1.0
  type=p
         0.7171     1.0
  type=p
         0.2137     1.0
  type=p
         0.0737     1.0
  type=d
         0.85     1.0
  type=d
         0.015     1.0
$end
```

```
$bigo zq
/ extra functions centered on O
type=s
0.0316  1.0
type=p
0.0254 1.0
type=d
2.0     1.0
type=d
0.32     1.0
type=d
0.128     1.0
$end

$bigh zq
/ extra functions centered on h
type=s
0.08  1.0
type=s
0.0333  1.0
type=p
0.2   1.0
type=p
0.05  1.0
$end
      subroutine press(bl,tl,symroot,symorb,nq,valence,core,
     # nroots,ops,nqvecs,orbtbf,symmap,rnum,rpt)
      implicit integer(a-z)
      real*8 bl(nq,*),tl(nq,*),xx,sdot,small,fpkey
      real*8 smallsq
      character*(*) ops
      character card*80,filnam*3
      dimension symroot(nroots),symorb(valence),nkeep(20)
      dimension orbtbf(*),symmap(valence),keep(20,20)
      dimension rnum(nroots),rpt(valence,nroots)
      logical logkey
      logical positn
      common /io/ inp,iout
c
c
      filnam='inp'
      if(logkey(ops,'kohn=qkeep',.false.,' '))then
        call intarr(ops,'kohn=qkeep',nkeep,nroots,' ')
        if(.not.positn('$qkeep',card,inp)) then
          call lnkerr(' no $vectors section found on '//filnam)
        endif
        write(iout,100)(nkeep(i),i=1,nroots)
        do i=1,nroots
          if(nkeep(i).ne.0)then
            read(inp,*)(keep(j,i),j=1,nkeep(i))
            write(iout,101)i,(keep(j,i),j=1,nkeep(i))
          else
            write(iout,*)' For root',i,' no orbitals used'
          end if
        enddo
      else
        do i=1,nroots
          nkeep(i)=valence
          do j=1,valence
            keep(j,i)=j
          enddo
          write(iout,101)i,(keep(j,i),j=1,nkeep(i))
        enddo
      end if
100   format('   number of nsmall orbitals to use in projection for each
     # target state:'/(10i5))
101   format(' For root',i3,' using orbitals:',(10i5))
      small=.0001
      write(iout,*)' '
      small=fpkey(ops,'kohn=smallcq',small,' ')
```

```
      write(iout,*)' Press SmallCQ ',small
c
c       ix=0
      ip=0
      nprev=0
      do 32 i=1,nroots
       is=0
       rnum(i)=0
c        do 33 j=1,valence
       do 33 j=1,nkeep(i)
          ix=nprev+keep(j,i)
c        ix=ix+1
       xx=sdot(nq,bl(1,ix),1,bl(1,ix),1)
       xx=sqrt(xx)
       if(logkey(ops,'print=m928',.false.,' ')) then
          write(iout,*)'Vector',ix,'' Norm='',xx
       endif
       if(xx.gt.small) then
        ip=ip+1
        is=is+1
        rpt(is,i)=ip
        call scopy(nq,bl(1,ix),1,tl(1,ip),1)
       end if
33     continue
       nprev=nprev+valence
       rnum(i)=is
       write(iout,11) i,is
32    continue
c
c
      if(logkey(ops,'kohn=nostatic',.false.,' ')) then
       write(iout,*)' Combining Transformation Vectors '
       if(logkey(ops,'print=m928',.false.,' ')) then
        write(iout,*)' Transformation Vector Before Contraction '
        call matout(tl,nq,ip,nq,ip,iout)
       end if
       do 50 i=1,nroots
       rn=rnum(i)
        if(rn.ne.0) then
        call rzero(bl(1,i),nq)
        do 51 j=1,rn
        jp=rpt(j,i)
         do 52 k=1,nq
         bl(k,i)=bl(k,i)+tl(k,jp)
52       continue
51      continue
       end if
50     continue
       ip=nroots
      else
       call scopy(nq*ip,tl,1,bl,1)
       if(logkey(ops,'print=m928',.false.,' ')) then
        write(iout,*)' Transformation Vector Before Normalization '
        call matout(bl,nq,ip,nq,ip,iout)
       end if
      end if
c
      if(logkey(ops,'kohn=nonorm',.false.,' ')) then
c
      write(iout,*)' UnNormalized Transformation Used '
c
      else
c
c     Normalized Transformation Vector
c
      do 60 i=1,ip
      xx=sdot(nq,bl(1,i),1,bl(1,i),1)
      xx=1./sqrt(xx)
      call sscal(nq,xx,bl(1,i),1)
60    continue
```

```
c
c     Overlap Matrix
c
      call ebtc(tl,bl,bl,ip,nq,ip)
c
      write(iout,61)
61    format(/,'  Raw Overlap Matrix ')
      call matout(tl,ip,ip,ip,ip,iout)
c
      write(iout,10) ip
10    format(/,' Total  No. of Penetration Terms Before Schmidt ',
     $ i4)
c
      smallsq=.0001
      smallsq=fpkey(ops,'kohn=smallsq',smallsq,' ')
      write(iout,*)'smallsq in press=',smallsq
      call schmdt(bl,tl,nq,ip,smallsq,ops)
c
      call ebtc(tl,bl,bl,ip,nq,ip)
c
      write(iout,62)
62    format(/,'  Final Overlap Matrix ')
      call matout(tl,ip,ip,ip,ip,iout)
c
      end if
c
11    format(' Root ',i3,'  No. of Orbital of Penetration Terms ',i4)
      write(iout,12) ip
12    format(/,' Total  No. of Orbital of Penetration Terms Retained ',
     $ i4)
c
      nqvecs=ip
c
      return
      end
```