Notes on the implementation of symbolic CSFs in the RCI code
Per Jönsson
Malmö, 2017-04-03

Problem formulation

So far we have not exploited that angular integration is independent of the principal quantum numbers. It is possible to use symbolic CSFs and do the angular integration for these symbolic CSFs to get angular coefficient. The angular information is then combined with radial integrals to form matrix elements between ordinary CSFs. Benefits:

1. CSF lists more compact. Things to be stored is the structure of the symbolic CSFs and information about the largest principal quantum numbers associated with the symbolic CSFs.
2. Angular data needed for constructing the direct- and exchange potentials as well as the Lagrange multipliers in MCDHF will decrease dramatically. This is good since the extensive reading from disc slows down the construction very much.
3. Construction of the interaction matrix in RCI can be done much more efficiently.

The CSF space

To make things general we consider, for each symmetry block, a set $\{\mathrm{CSF}_{\mathrm{GEN}}\}$ of CSFs obtained by some general substitution method to a canonically ordered orbital set $\{\mathrm{ORB}_{\mathrm{GEN}}\}$ augmented by a set $\{\mathrm{CSF}_{\mathrm{SD\text{-}MR}}\}$ of CSFs obtained by SD-MR substitutions to a symmetry ordered orbital set $\{\mathrm{ORB}_{\mathrm{SD\text{-}MR}}\}$. As a concrete example we start to define a total orbital set in the file `clist.ref`

```
1s
2s
2p
3s
3p
3d
4s
5s
6s
7s
4p
5p
```

```
6p
7p
4d
5d
6d
7d
```

The canonically ordered orbitals `1s,2s,2p,3s,3p,3d` are elements in $\{\mathrm{ORB}_{\mathrm{GEN}}\}$ and the symmetry ordered orbitals `4s,5s,6s,7s,4p,5p,6p,7p,4d,5d,6d,7d` are elements in $\{\mathrm{ORB}_{\mathrm{SD\text{-}MR}}\}$. We then run `rcsfgenerate` with the input

```
u              ! Orbital order
0              ! Selected core
1s(2,*)2s(1,*)2p(1,*)
*
3s,3p,3d
0           0  ! Lower and higher 2*J
4              ! Number of excitations
y
1s(2,*)2s(1,*)2p(1,*)
*
7s,7p,7d
0           0  ! Lower and higher 2*J
2              ! Number of excitations
n
```

The first part will generate $\{\mathrm{CSF}_{\mathrm{GEN}}\}$ by SDTQ substitutions from $1s^2 2s2p$ to $\{\mathrm{ORB}_{\mathrm{GEN}}\}$. This set of CSFs will be augmented by $\{\mathrm{CSF}_{\mathrm{SD\text{-}MR}}\}$ obtained by SD substitutions from $1s^2 2s2p$ to $\{\mathrm{ORB}_{\mathrm{SD\text{-}MR}}\}$. The total CSF space is

$$\{\mathrm{CSF}_{\mathrm{GEN}}\} \cup \{\mathrm{CSF}_{\mathrm{SD\text{-}MR}}\},$$

where the two sets are disjoint. The CSFs generated by `rcsfgenerate` should be rearranged by `rcsfzerofirst` so that the CSFs in $\{\mathrm{CSF}_{\mathrm{GEN}}\}$ comes first in the list.

Structure of $\{\mathrm{CSF}_{\mathrm{GEN}}\} \cup \{\mathrm{CSF}_{\mathrm{SD\text{-}MR}}\}$

The CSFs in $\{\mathrm{CSF}_{\mathrm{GEN}}\}$ are built from orbitals in $\{\mathrm{ORB}_{\mathrm{GEN}}\}$ and can not, in the general case, be grouped by angular properties.

The CSFs in $\{\mathrm{CSF}_{\mathrm{SD\text{-}MR}}\}$ have either one or two orbitals from $\{\mathrm{ORB}_{\mathrm{SD\text{-}MR}}\}$. The remaining orbitals are from $\{\mathrm{ORB}_{\mathrm{GEN}}\}$. The CSFs in $\{\mathrm{CSF}_{\mathrm{SD\text{-}MR}}\}$ can, due to the fact that $\{\mathrm{ORB}_{\mathrm{SD\text{-}MR}}\}$ is ordered by angular symmetry, be ordered in groups sharing angular quantum numbers. The groups of CSFs sharing angular properties are of four general types. Together with the CSFs in $\{\mathrm{CSF}_{\mathrm{GEN}}\}$ we have five different types.

Type 1: CSFs with all orbitals $\{\mathrm{ORB}_{\mathrm{GEN}}\}$

One example of type 1, all orbitals $\{\mathrm{ORB}_{\mathrm{GEN}}\}$, is

```
  1s  ( 2)  2s  ( 1)  2p-( 1)
                 1/2       1/2
                              0-
```

For type 1 we do not consider groups of CSFs having the same angular structure. Each CSFs of type 1 are considered as ordinary CSFs.

Type 2: CSFs with one orbital in $\{\mathrm{ORB}_{\mathrm{SD\text{-}MR}}\}$

One example of a grouping is

```
  1s  ( 2)  2s  ( 1)  4p-( 1)
                 1/2       1/2
                              0-
  1s  ( 2)  2s  ( 1)  5p-( 1)
                 1/2       1/2
                              0-
  1s  ( 2)  2s  ( 1)  6p-( 1)
                 1/2       1/2
                              0-
  1s  ( 2)  2s  ( 1)  7p-( 1)
                 1/2       1/2
                              0-
```

This group of four CSFs is spanned by the symbolic CSF

```
  1s  ( 2)  2s  ( 1)  7p-( 1)
                 1/2       1/2
                              0-
```

where it is implied that `7p-` will take on all the orbitals of the same symmetry in $\{\mathrm{ORB}_{\mathrm{SD\text{-}MR}}\}$, i.e. `4p-, 5p-, 6p-, 7p-`

Type 3: CSFs with two orbitals in $\{\mathrm{ORB}_{\mathrm{SD\text{-}MR}}\}$, different symmetry

One example of a grouping is

```
  1s  ( 2)  4s  ( 1)  4p-( 1)
                 1/2       1/2
                              0-
  1s  ( 2)  4s  ( 1)  5p-( 1)
                 1/2       1/2
```

```
                                   0-
  1s ( 2)  4s ( 1)  6p-( 1)
               1/2      1/2
                                   0-
  1s ( 2)  4s ( 1)  7p-( 1)
               1/2      1/2
                                   0-
  1s ( 2)  5s ( 1)  4p-( 1)
               1/2      1/2
                                   0-
  1s ( 2)  5s ( 1)  5p-( 1)
               1/2      1/2
                                   0-
  1s ( 2)  5s ( 1)  6p-( 1)
               1/2      1/2
                                   0-
  1s ( 2)  5s ( 1)  7p-( 1)
               1/2      1/2
                                   0-
  1s ( 2)  6s ( 1)  4p-( 1)
               1/2      1/2
                                   0-
  1s ( 2)  6s ( 1)  5p-( 1)
               1/2      1/2
                                   0-
  1s ( 2)  6s ( 1)  6p-( 1)
               1/2      1/2
                                   0-
  1s ( 2)  6s ( 1)  7p-( 1)
               1/2      1/2
                                   0-
  1s ( 2)  7s ( 1)  4p-( 1)
               1/2      1/2
                                   0-
  1s ( 2)  7s ( 1)  5p-( 1)
               1/2      1/2
                                   0-
  1s ( 2)  7s ( 1)  6p-( 1)
               1/2      1/2
                                   0-
  1s ( 2)  7s ( 1)  7p-( 1)
               1/2      1/2
                                   0-
```

This group of CSFs is spanned by the symbolic CSF

```
  1s ( 2)  7s ( 1)  7p-( 1)
               1/2      1/2
                           0-
```

where it is implied that `7s` and `7p-` will take on all the orbitals of the same symmetry in $\{\mathrm{ORB}_{\text{SD-MR}}\}$, i.e. `4s, 5s, 6s, 7s` and `4p-, 5p-, 6p-, 7p-`.

Type 4: CSFs with two orbitals in $\{\mathrm{ORB}_{\text{SD-MR}}\}$, same symmetry

One example of a grouping is

```
  2s ( 1)  2p-( 1)  4s ( 1)  5s ( 1)
      1/2      1/2      1/2      1/2
                    1       1/2      0-
  2s ( 1)  2p-( 1)  4s ( 1)  6s ( 1)
      1/2      1/2      1/2      1/2
                    1       1/2      0-
  2s ( 1)  2p-( 1)  4s ( 1)  7s ( 1)
      1/2      1/2      1/2      1/2
                    1       1/2      0-
  2s ( 1)  2p-( 1)  5s ( 1)  6s ( 1)
      1/2      1/2      1/2      1/2
                    1       1/2      0-
  2s ( 1)  2p-( 1)  5s ( 1)  7s ( 1)
      1/2      1/2      1/2      1/2
                    1       1/2      0-
  2s ( 1)  2p-( 1)  6s ( 1)  7s ( 1)
      1/2      1/2      1/2      1/2
                    1       1/2      0-
```

This group of CSFs is spanned by the symbolic CSF

```
  2s ( 1)  2p-( 1)  6s ( 1)  7s ( 1)
      1/2      1/2      1/2      1/2
                    1       1/2      0-
```

where it is implied that `6s` and `7s` will take on all allowed combinations of the orbitals of the same symmetry in $\{\mathrm{ORB}_{\text{SD-MR}}\}$.

Type 5: CSFs with doubly occupied orbitals in $\{\mathrm{ORB}_{\text{SD-MR}}\}$

One example of a grouping

```
  2s ( 1)  2p-( 1)  4s ( 2)
      1/2      1/2
                    0      0-
  2s ( 1)  2p-( 1)  5s ( 2)
      1/2      1/2
                    0      0-
  2s ( 1)  2p-( 1)  6s ( 2)
      1/2      1/2
                    0      0-
  2s ( 1)  2p-( 1)  7s ( 2)
      1/2      1/2
                    0      0-
```

This group of CSFs is spanned by the symbolic CSF

```
  2s ( 1)  2p-( 1)  7s ( 2)
      1/2      1/2
                    0      0-
```

where it is implied that `7s` will take on all the orbitals of the same symmetry in $\{\mathrm{ORB}_{\text{SD-MR}}\}$, i.e. `4s, 5s, 6s, 7s`

Generation of $\{\mathrm{CSF}_{\text{GEN}}\} \cup \{\mathrm{CSF}_{\text{SD-MR}}\}$

The generation of a CSF list $\{\mathrm{CSF}_{\text{GEN}}\} \cup \{\mathrm{CSF}_{\text{SD-MR}}\}$, where each group in $\{\mathrm{CSF}_{\text{SD-MR}}\}$ is represented by a symbolic CSF, can be done in a straight forward manner with a modified version of `rcsfgenerate`. A very simple example of such a list, that we later will use to exemplify the code modifications, is

```
Core subshells:

Peel subshells:
  1s   2s   2p-  2p   3s   4s   5s   6s   7s   3p-  4p-  5p-  6p-  7p-
CSF(s):
  1s ( 2)  2s ( 1)  2p-( 1)
               1/2      1/2
                           0-
  1s ( 1)  2s ( 2)  2p-( 1)
      1/2               1/2
                           0-
  1s ( 1)  2p-( 1)  2p ( 2)
      1/2      1/2        0
                    0      0-
  2s ( 1)  2p-( 1)  2p ( 2)
```

```
     1/2       1/2         0
                     0      0-
  1s ( 2)  2s ( 1)  7p-( 1)
               1/2      1/2
                            0-
  1s ( 2)  7s ( 1)  7p-( 1)
               1/2      1/2
                            0-
  2s ( 1)  2p-( 1)  6s ( 1)  7s ( 1)
     1/2       1/2      1/2       1/2
                     1      1/2      0-
  2s ( 1)  2p-( 1)  7s ( 2)
     1/2       1/2
                     0      0-
```

where the first four CSFs are of type 1 and the four last are symbolic CSFs of types 2 - 5 spanning groups of CSFs. Note the order of the orbital set. The first four orbitals building the CSFs of type 1 are in canonical order. The remaining orbitals are ordered by symmetry.

Input to the new RCI program

As input to the new RCI program we have a list of CSFs as described above. Given the information from the orbital set the new subroutine `findtype` reads the list and determines the type of each CSF, the number of CSFs spanned by the different symbolic CSFs, the start and end position in the orbital list of each (symbolic) orbital in the total orbital list. The information is stored in a two dimensional array `ntype(6,NCSF)`.

`ntype(1,J)` type of CSFs J in the list

`ntype(2,J)` number of CSFs spanned by CSFs J in the list

`ntype(3,J)` starting position of first symbolic orbital of CSFs J in the total orbital list

`ntype(4,J)` end position of first symbolic orbital of CSFs J in the total orbital list

`ntype(5,J)` starting position of second symbolic orbital of CSFs J in the total orbital list

`ntype(6,J)` end position of second symbolic orbital of CSFs J in the total orbital list

As an example we look at CSFs 5 to 8. The information is

```
ntype(:,5) = 2   5  10  14   0   0
ntype(:,6) = 3  25   5   9  10  14
nyype(:,7) = 4  10   5   8   5   9
nyype(:,8) = 5   5   5   9   0   0
```

where we for CSF 7 with type 4 have agreed on the convention that the principal quantum number of the second symbolic orbital is always larger that the principal quantum number of the first symbolic orbital. `findtype.f` also computes and outputs the number of CSFs of type 1, `NCFGEN`, the total number of CSFs, `NCFTOT`, spanned by the CSF list and finally `NBLOCK` the largest number of CSFs spanned by a single symbolic CSF.

Structure of the Hamiltonian matrix

In our example NCFGEN $= 4$, NCFTOT $= 49$ and NBLOCK $= 25$. The 49×49 matrix has the following block structure

$$\begin{pmatrix} 1\times 1 & 1\times 1 & 1\times 1 & 1\times 1 & 1\times 5 & 1\times 25 & 1\times 10 & 1\times 5 \\ & 1\times 1 & 1\times 1 & 1\times 1 & 1\times 5 & 1\times 25 & 1\times 10 & 1\times 5 \\ & & 1\times 1 & 1\times 1 & 1\times 5 & 1\times 25 & 1\times 10 & 1\times 5 \\ & & & 1\times 1 & 1\times 5 & 1\times 25 & 1\times 10 & 1\times 5 \\ & & & & 5\times 5 & 5\times 25 & 5\times 10 & 5\times 5 \\ & & & & & 25\times 25 & 25\times 10 & 25\times 5 \\ & & & & & & 10\times 10 & 10\times 5 \\ & & & & & & & 5\times 5 \end{pmatrix},$$

where for the diagonal blocks only the upper triangular parts should be computed. All matrix elements of a block are computed at the same time (to be described). The matrix elements for a block are densely stored in an array EMTBLOCK(NBLOCK,NBLOCK).

Computing the Hamiltonian matrix and save if to file

Old setham

In the old subroutine setham_gg.f the matrix elements are computed are computed one by one. The non-zero matrix elements along with the row positions for each column are saved to arrays EMT and IROW. The arrays are written to the file rci.res column by column. The code snippet is

```
      NELC = 0
      DO IC = 1,NCF      !Loop over columns
        DO IR = 1,IC     !Loop over rows

          CALL ONESCALAR AND RKCO_GG AND COMPUTE THE <IC|H|IR> MATRIX ELEMENT
          IF <IC|H|IR> NONZERO
            NELC = NELC + 1
            SAVE MATRIX ELEMENT AND ROW IN EMT(NELC) AND IROW(NELC)
          END IF

        END DO

        WRITE(IMCDF) NELC,ELSTO,(EMT(IR),IR = 1,NELC),! non-zero elements and
     :                         (IROW(IR),IR = 1,NELC)! row positions for
                                                     ! current column
      END DO
```

New setham

In the new subroutine `setham_gg.f` the matrix elements are computed by block.

```
ICTOT = 0          ! Total IC counter
NELC = 0
DO IC = 1,NCF      !Loop over columns (of symbolic CSFs)
  IRTOT = 0        !Total IR counter
  DO IR = 1,IC     !Loop over rows (of symbolic CSFs)

    CALL SUBROUTINE THAT COMPUTES ALL THE MATRIX ELEMENT IN
    THE IC, IR BLOCK.
    SAVE IN EMTBLOCK
    UPDATE THE ICTOT AND IRTOT COUNTERS
    IF MATRIX ELEMENTS IN EMTBLOCK ARE NON-ZERO
      CALL SUBROUTINE TRANSFER TO UPDATE THE NELC COUNTER
      (ONE DIMENSIONAL ARRAY) AND SAVE MATRIX ELEMENTS AND
      AND ROWS IN EMT AND IROW (TWO DIMENSIONAL ARRAYS)
    END IF

  END DO

  DO I = 1,NCFS OF SYMBOLIC CSF IC
    WRITE(IMCDF) NELC(I),ELSTO,(EMT(IR,I),IR = 1,NELC(I)),! non-zero elements
:                            (IROW(IR,I),IR = 1,NELC(I)) ! and row positions
  END DO                                                 ! current columns
END DO
```

Subroutine transfer

The new subroutine `transfer.f` successively takes the arrays `EMT(NCFTOT,NBLOCK)` that stores all matrix elements of the blocks corresponding to `IC` and transfers the non-zero matrix elements and the row positions (total row) to the arrays `EMT(NCFTOT,NBLOCK)` and `IROW(NCFTOT,NBLOCK)`. The number of elements transferred for each column corresponding to `IC` are saved in an one-dimensional array `NELC(NBLOCK)`. After the loop over `IR` is the content of `EMT` and `IROW` are then written column by column.

As an example we look at the blocks marked in red corresponding to the interactions of the CSFs spanned by the seventh CSF (symbolic) of the input list.

$$\begin{pmatrix} 1\times 1 & 1\times 1 & 1\times 1 & 1\times 1 & 1\times 5 & 1\times 25 & 1\times 10 & 1\times 5 \\ & 1\times 1 & 1\times 1 & 1\times 1 & 1\times 5 & 1\times 25 & 1\times 10 & 1\times 5 \\ & & 1\times 1 & 1\times 1 & 1\times 5 & 1\times 25 & 1\times 10 & 1\times 5 \\ & & & 1\times 1 & 1\times 5 & 1\times 25 & 1\times 10 & 1\times 5 \\ & & & & 5\times 5 & 5\times 25 & 5\times 10 & 5\times 5 \\ & & & & & 25\times 25 & 25\times 10 & 25\times 5 \\ & & & & & & 10\times 10 & 10\times 5 \\ & & & & & & & 5\times 5 \end{pmatrix},$$

The content of the first block (1×10) is computed and stored in a 1×10 submatrix of `EMTBLOCK(25,25)`. The non-zero-elements and the rows are transferred to `EMT(49,25)` and `IROW(49,25)`. The content of block two (1×10) is computed and stored in a 1×10 submatrix `EMTBLOCK(25,25)`. The non-zero-elements and the rows are appended to `EMT(49,25)` and `IROW(49,25)`. Finally the content of the upper triangular part of block seven (10×10) is computed and stored in a 10×10 upper triangular submatrix `EMTBLOCK(25,25)`. The non-zero-elements and the rows are appended to `EMT(49,25)` and `IROW(49,25)`. The number of non-zero elements for columns 1 to 10 in `EMT(49,25)` are stored in the 10 first positions of `NELC(25)`.

Computation of matrix elements of a block

The computation of all matrix elements of a block is done by a call to a subroutine `matrixblocab` where `a` and `b` indicate the type. There are the following combinations:

`matrixblock11`	matrix element between two CSFs of type 1.
`matrixblock12`	matrix elements between CSFs of type 1 and 2
`matrixblock13`	matrix elements between CSFs of type 1 and 3
`matrixblock14`	matrix elements between CSFs of type 1 and 4
`matrixblock15`	matrix elements between CSFs of type 1 and 5
`matrixblock2`	matrix element between the same CSFs of type 2
`matrixblock22`	matrix elements between different CSFs of type 2
`matrixblock23`	matrix elements between CSFs of type 2 and 3
`matrixblock24`	matrix elements between CSFs of type 2 and 4
`matrixblock25`	matrix elements between CSFs of type 2 and 5
`matrixblock3`	matrix element between the same CSFs of type 3
`matrixblock33`	matrix elements between different CSFs of type 3
`matrixblock34`	matrix elements between CSFs of type 3 and 4
`matrixblock35`	matrix elements between CSFs of type 3 and 5
`matrixblock4`	matrix element between the same CSFs of type 4
`matrixblock44`	matrix elements between different CSFs of type 4
`matrixblock45`	matrix elements between CSFs of type 4 and 4
`matrixblock5`	matrix elements between the same CSFs of type 5
`matrixblock55`	matrix elements between different CSFs of type 5.

matrixblock11

Description of each type ....

Using symbolic CSFs, matrix elements need to be calculated only in a few cases to get the angular coefficients. Then all the remaining matrix elements can be directly obtained by changing the radial integrals. For illustration we take the concrete example of an input file to the modified `RCI` program.

```
  1s ( 2)  2s ( 1)  2p-( 1)
                1/2      1/2
                            0-
  1s ( 2)  2s ( 1)  7p-( 1)
```

```
              1/2       1/2
                           0-
  1s ( 2)  7s ( 1)  7p-( 1)
              1/2       1/2
                           0-
```

The first CSFs is an ordinary CSF in $\{\mathrm{CSF}_{\mathrm{GEN}}\}$. The second and third are symbolic CSFs spanning several CSFs as discussed above. This example is simple, but general enough to convey the idea.

CSF in $\{\mathrm{CSF}_{\mathrm{GEN}}\}$ and symbolic CSF

We compute the matrix element between

```
  1s ( 2)  2s ( 1)  2p-( 1)
              1/2       1/2
                           0-
```

and

```
 1s ( 2)  2s ( 1)  7p-( 1)
              1/2       1/2
                           0-
```

This gives

```
    1.000000000 I(2p-,7p-)
    1.000000000 R0(2s 2p-,2s 7p-)
   -0.333333333 R1(2s 2s ,7p-2p-)
    2.000000000 R0(1s 2p-,1s 7p-)
   -0.333333333 R1(1s 1s ,7p-2p-)
```

The matrix elements between

```
  1s ( 2)  2s ( 1)  2p-( 1)
              1/2       1/2
                           0-
```

and all the 4 CSFs spanned by the symbolic CSF now follow immediately by keeping the angular coefficients and multiplying with appropriate radial integrals.

We compute the matrix element between

```
  1s ( 2)  2s ( 1)  2p-( 1)
              1/2       1/2
                           0-
```

and

```
 1s ( 2)  7s ( 1)  7p-( 1)
              1/2       1/2
                           0-
```

This gives

```
  1.000000000 R0(2s 2p-,7s 7p-)
 -0.333333333 R1(2s 2p-,7p-7s )
```

The matrix elements between

```
 1s ( 2)  2s ( 1)  2p-( 1)
              1/2      1/2
                          0-
```

and all the 16 CSFs spanned by the symbolic CSF again follow by keeping the angular coefficients and multiplying with appropriate radial integrals.

Matrix elements within a symbolic CSF

Symbolic CSF with one orbital in $\{\mathrm{ORB}_{\text{SD-MR}}\}$

The diagonal matrix element between

```
1s ( 2)  2s ( 1)  7p-( 1)
              1/2      1/2
                          0-
```

and

```
1s ( 2)  2s ( 1)  7p-( 1)
              1/2      1/2
                          0-
```

is

```
  2.000000000 I(1s ,1s )
  1.000000000 I(2s ,2s )
  1.000000000 I(7p-,7p-)
  1.000000000 R0(2s 7p-,2s 7p-)
 -0.333333333 R1(2s 2s ,7p-7p-)
  1.000000000 R0(1s 1s ,1s 1s )
  2.000000000 R0(1s 2s ,1s 2s )
 -1.000000000 R0(1s 1s ,2s 2s )
  2.000000000 R0(1s 7p-,1s 7p-)
 -0.333333333 R1(1s 1s ,7p-7p-)
```

whereas the off-diagonal element between

```
1s ( 2)  2s ( 1)  6p-( 1)
              1/2      1/2
                          0-
```

and

```
 1s ( 2)  2s ( 1)  7p-( 1)
              1/2      1/2
                          0-
```

is

```
   1.000000000 I(6p-,7p-)
   1.000000000 R0(2s 6p-,2s 7p-)
  -0.333333333 R1(2s 2s ,6p-7p-)
   2.000000000 R0(1s 6p-,1s 7p-)
  -0.333333333 R1(1s 1s ,6p-7p-)
```

These are the only two cases, and all the matrix elements between CSFs spanned by the symbolic CSF follow by keeping the angular coefficients and multiplying with appropriate radial integrals. In the first case we can save

```
   2.000000000 I(1s ,1s )
   1.000000000 I(2s ,2s )
   1.000000000 R0(1s 1s ,1s 1s )
   2.000000000 R0(1s 2s ,1s 2s )
  -1.000000000 R0(1s 1s ,2s 2s )
```

and there is no need to recompute this part for the other matrix elements, but we just modify the terms containing the orbitals in $\{\mathrm{ORB}_{\text{SD-MR}}\}$.

Symbolic CSF with two orbitals in $\{\mathrm{ORB}_{\text{SD-MR}}\}$

The matrix element between

```
 1s ( 2)  7s ( 1)  7p-( 1)
              1/2      1/2
                          0-
```

and

```
 1s ( 2)  7s ( 1)  7p-( 1)
              1/2      1/2
                          0-
```

is

```
   2.000000000 I(1s ,1s )
   1.000000000 I(7s ,7s )
   1.000000000 I(7p-,7p-)
   1.000000000 R0(7s 7p-,7s 7p-)
  -0.333333333 R1(7s 7s ,7p-7p-)
   1.000000000 R0(1s 1s ,1s 1s )
```

```
    2.000000000 R0(1s 7s ,1s 7s )
   -1.000000000 R0(1s 1s ,7s 7s )
    2.000000000 R0(1s 7p-,1s 7p-)
   -0.333333333 R1(1s 1s ,7p-7p-)
```

The matrix element between

```
 1s ( 2)  6s ( 1)  7p-( 1)
              1/2      1/2
                          0-
```

and

```
 1s ( 2)  7s ( 1)  7p-( 1)
              1/2      1/2
                          0-
```

is

```
    1.000000000 I(6s ,7s )
    1.000000000 R0(6s 7p-,7s 7p-)
   -0.333333333 R1(6s 7s ,7p-7p-)
    2.000000000 R0(1s 6s ,1s 7s )
   -1.000000000 R0(1s 1s ,6s 7s )
```

The matrix element between

```
 1s ( 2)  7s ( 1)  6p-( 1)
              1/2      1/2
                          0-
```

and

```
 1s ( 2)  7s ( 1)  7p-( 1)
              1/2      1/2
                          0-
```

is

```
    1.000000000 I(6p-,7p-)
   -0.333333333 R1(6p-7s ,7s 7p-)
    1.000000000 R0(6p-7s ,7p-7s )
    2.000000000 R0(1s 6p-,1s 7p-)
   -0.333333333 R1(1s 1s ,6p-7p-)
```

Finally the matrix element between

```
 1s ( 2)  6s ( 1)  6p-( 1)
              1/2      1/2
                          0-
```

and

```
 1s ( 2)  7s ( 1)  7p-( 1)
              1/2      1/2
                          0-
```

is

```
    1.000000000 R0(6s 6p-,7s 7p-)
   -0.333333333 R1(6s 6p-,7p-7s )
```

These are the only four cases, and all the matrix elements between CSFs spanned by the symbolic CSF follow by keeping the angular coefficients and multiplying with appropriate radial integrals.

Matrix elements between two symbolic CSF

The matrix element between

```
 1s ( 2)  2s ( 1)  7p-( 1)
              1/2      1/2
                          0-
```

and

```
 1s ( 2)  7s ( 1)  7p-( 1)
              1/2      1/2
                          0-
```

is

```
    1.000000000 I(2s ,7s )
    1.000000000 R0(2s 7p-,7s 7p-)
   -0.333333333 R1(2s 7s ,7p-7p-)
    2.000000000 R0(1s 2s ,1s 7s )
   -1.000000000 R0(1s 1s ,2s 7s )
```

The matrix element between

```
 1s ( 2)  2s ( 1)  6p-( 1)
              1/2      1/2
                          0-
```

and

```
 1s ( 2)  7s ( 1)  7p-( 1)
              1/2      1/2
                          0-
```

is

```
    1.000000000 R0(2s 6p-,7s 7p-)
   -0.333333333 R1(2s 6p-,7p-7s )
```

These are the only two cases, and all the matrix elements between CSFs spanned by the two symbolic CSFs follow by keeping the angular coefficients and multiplying with appropriate radial integrals.

Comments

The examples above, although simple, are quite general. Whatever matrix element we consider there will never be more than 5 cases and thus it will not be very difficult to code. In addition the selections will be fast.

Code implementation

To make RCI run with $\{\mathrm{CSF}_{\mathrm{GEN}}\} \cup \{\mathrm{CSF}_{\mathrm{SD\text{-}MR}}\}$ where $\{\mathrm{CSF}_{\mathrm{SD\text{-}MR}}\}$ is spanned by symbolic CSFs, only the subroutine `setham_gg.f` needs to be modified. This is a great advantage!

Variables that need to be accessed

We need to have access to the occupation numbers and angular data for each CSF, some of which are symbolic, in the list. These data are packed and stored in `integer*4` pointer arrays

```
INTEGER*4 IQA,JQSA,JCUPA
POINTER (PNTRIQ,IQA(NNNWP,1))
POINTER (PNTJQS,JQSA(NNNWP,3,1))
POINTER (PNJCUP,JCUPA(NNNWP,1))
```

Associated with the pointer arrays are the common blocks

```
COMMON/ORB2/NCF,NW,PNTRIQ
COMMON/STAT/PNTJQS,PNJCUP
```

The packing makes things a little more complicated. To write out the occupation number for CSFs number `J` we can call the function `iq.f`

```
DO I = 1,NW
  WRITE(*,*) I,IQ(I,J)
END DO
```

To change the occupation number for CSFs number `J` according to what we define in an array `IOCCI`, where

```
INTEGER IOCCI(NNNW),
```

we must use the `pack.f` subroutine

```
IAQ(:,J) = 0
DO I = 1,NW
   CALL PACK(IOCCI(I),I,IQA(1,J))
END DO
```

(HERE I AM UNCERTAIN IF IT SHOULD BE `CALL PACK(IOCCI(I),I,IQA(1,J))` or `CALL PACK(IOCCI(I),I,IQ`
PLEASE CHECK)

Additional variables to count CSFs

Consider our list above with one CSF from $\{\mathrm{CSF}_{\mathrm{GEN}}\}$ and two symbolic CSFs that span $\{\mathrm{CSF}_{\mathrm{SD\text{-}MR}}\}$

```
  1s ( 2)  2s ( 1)  2p-( 1)
               1/2      1/2
                           0-
  1s ( 2)  2s ( 1)  7p-( 1)
               1/2      1/2
                           0-
  1s ( 2)  7s ( 1)  7p-( 1)
               1/2      1/2
                           0-
```

The variables `IC` and `IR` in `setham_gg.f` refer to these 3 CSFs. Since CSF 2 and 3 in this list represent several CSFs we need to introduce additional variables `ICS` and `IRS` that keeps track of the actual CSFs.

All matrix elements must be buffered differently

In the ordinary version all the matrix elements `EMT` and row positions `IROW` of a column are buffered before written to disk (`rci.res`)

```
*   This column is done; write it to disk
*
         WRITE (imcdf) NELC, ELSTO, (EMT(IR), IR = 1, NELC),
     :                          (IROW(IR), IR = 1, NELC)
```

Since all CSFs of a symbolic CSF are treated together this means that matrix elements `EMT` and row positions `IROW` of several columns must be buffered before written to disk. Say that there are N CSFs in a symbolic CSF then N columns must be buffered.

Matrix elements between CSF1 and symbolic CSF2

Here `IC=2` and `IR=1`. We call

```
      CALL ONESCALAR(IC,IR,IA,IB,TSHELL)
```

Only two cases are possible `IA = 0`, matrix element is zero (implying that all matrix elements with CSFs spanned by the symbolic CSFs will be zero) or `IA ~= 0`. In the latter case `IA ~= IB` (convince yourself) and we have access to

```
     TCOEFF = DBLE(TSHELL(1))
```

Then each of the one-particle matrix elements will be obtained by

```
      CALL IABINT (IA, IB, TEGRAL)
      ELEMNT = TCOEFF*TEGRAL
```

where IB is modified according to which orbital that is occupied in the CSFs of the symbolic CSF. We call

```
      CALL RKCO_GG (IC, IR, CORD, INCOR, 1)
```

If NVCOEF ~= 0 we have access to

```
      NVCOEF, COEFF(I), LABEL(1,I), LABEL(2,I), LABEL(3,I),
                LABEL(4,I), LABEL(5,I), I = 1,NVCOEF
```

Then each of the two-particle matrix elements will be obtained by

```
      ELEMNT = 0.DO
      DO 7 I = 1, NVCOEF
         VCOEFF = COEFF(I)
         IF (DABS (VCOEFF) .GT. CUTOFF) THEN
               CALL RKINTC (LABEL(1,I), LABEL(2,I),
     :                          LABEL(3,I), LABEL(4,I),
     :                          LABEL(5,I), TEGRAL)
               ELEMNT = ELEMNT + TEGRAL*VCOEFF
         ENDIF
 7    CONTINUE
      END DO
```

where LABEL(3,I), LABEL(4,I) are modified according to which orbitals that are occupied in the CSFs of the symbolic CSF.

Matrix elements within symbolic CSF2

Now IC=2 and IR=2. This is the diagonal case. We call

```
      CALL ONESCALAR(IC,IR,IA,IB,TSHELL)
```

Then we save, in a vector, TCOEFF from the loop

```
      DO IA = 1,NW
         TCOEFF = DBLE(TSHELL(IA))
         IF (DABS (TCOEFF) .GT. CUTOFF) THEN
      END DO
```

We now have all the information to compute all the diagonal one-electron matrix elements. For the two-electron matrix elements we call

```
      CALL RKCO_GG (IC, IR, CORD, INCOR, 1)
```

and proceed as above. We now have all the information to compute also the diagonal two-electron matrix elements.

To compute the off-diagonal interactions we need to rely on a trick. We set IC=1 (or more generally to IR-1) and assign CSF1 the angular quantum number of CSF2. This can be done by

```
JQSADUM1(:) = JQSA(:,1,1)
JQSADUM2(:) = JQSA(:,2,1)
JQSADUM3(:) = JQSA(:,3,1)
JCUPADUM(:) = JCUPA(:,1)
IQADUM(:) = IQA(:,1)
```

and

```
JQSA(:,1,1) = JQSA(:,1,2)
JQSA(:,2,1) = JQSA(:,2,2)
JQSA(:,3,1) = JQSA(:,3,2)
JCUPA(1,:) = JCUPA(:,2)
```

We than have to modify the occupation number array `IQA` (use `iq.f` and `pack.f` as described above) so that we save the CSF

```
1s ( 2)  2s ( 1)  6p-( 1)
             1/2      1/2
                         0-
```

in `IC=1`. Proceed to compute and save the angular data from the one- and two-particle operators with calls to

```
CALL ONESCALAR(IC,IR,IA,IB,TSHELL)
```

and

```
CALL RKCO_GG (IC, IR, CORD, INCOR, 1)
```

We now have all the information to compute all the off-diagonal one-electron and two-electron matrix elements. After the computation is done we have to restore data for `IC=1` by

```
JQSA(:,1,1) = JQSADUM1(:)
JQSA(:,2,1) = JQSADUM2(:)
JQSA(:,3,1) = JQSADUM3(:)
JCUPA(:,1) = JCUPADUM(:)
IQA(:,1) = IQADUM(:)
```

Matrix elements within symbolic CSF3

Proceed as above. The only difference is that we have to modify the occupation number array `IQA` and issue calls to

```
CALL ONESCALAR(IC,IR,IA,IB,TSHELL)
```

and

```
CALL RKCO_GG (IC, IR, CORD, INCOR, 1)
```

more than once since we have more cases to account for.

Matrix elements between symbolic CSF2 and CSF3

In this case the angular quantum numbers are OK and we only have to modify the occupation numbers before repeated calls to

```
CALL ONESCALAR(IC,IR,IA,IB,TSHELL)
```

and

```
CALL RKCO_GG (IC, IR, CORD, INCOR, 1)
```

We should now have all the information to compute all the one-electron and two-electron matrix elements.