Description of i/o file format for $\mathcal{CI}$tool and related codes

Andrea Bertoni, January 2011 pdflatex'ed January 26, 2011

Introduction and notation

$\mathcal{CI}$tool is a full-Configuration-Interaction solver for a multi-particle (MP) system, namely electrons and holes. It computes eigenvalues/vectors of the Hamiltonian

$$H = H_0 + H_I \tag{1}$$

on the basis of Slater determinants of single-particle (SP) states. The free-particle and interaction Hamiltonians read (in second-quantization formalism)

$$H_0 = \sum_n \epsilon_n e_n^\dagger e_n + \sum_m \bar{\epsilon}_m h_m^\dagger h_m \tag{2}$$

$$H_I = \frac{1}{2}\sum_{\substack{n_1,n_2\\n_3,n_4}} V^{(ee)}_{\substack{n_1,n_2\\n_3,n_4}} e^\dagger_{n_1} e^\dagger_{n_2} e_{n_3} e_{n_4} + \frac{1}{2}\sum_{\substack{m_1,m_2\\m_3,m_4}} V^{(hh)}_{\substack{m_1,m_2\\m_3,m_4}} h^\dagger_{m_1} h^\dagger_{m_2} h_{m_3} h_{m_4} + \frac{1}{2}\sum_{\substack{m_1,n_2\\n_3,m_4}} V^{(he)}_{\substack{m_1,n_2\\n_3,m_4}} h^\dagger_{m_1} e^\dagger_{n_2} e_{n_3} h_{m_4},$$

where ϵ_n and $\bar{\epsilon}_m$ are the SP energies for the SP electron state n and hole state m, respectively. The subscripts n and m run on the SP states included in the calculation, for electrons and holes: S and $\bar{S}$, respectively. So $n = 1 \dots S$ and $m = 1 \dots \bar{S}$. The annihilation and creation operators e_n and $e_n^\dagger$ act on the SP electron state n. The operators h_m and $h_m^\dagger$ act on the SP hole state m.

The interaction elements $V^{(ee)}_{\substack{n_1,n_2\\n_3,n_4}}$, $V^{(hh)}_{\substack{m_1,m_2\\m_3,m_4}}$ and $V^{(he)}_{\substack{m_1,n_2\\n_3,m_4}}$ are an input of $\mathcal{CI}$tool and are obtained from the SP eigenstates, usually from their real-space representation, or from the real-space representation of a component of the SP state.
The SP states are obtained by solving the SP Schrödinger equations

$$H^{(e)}_{SP}|\phi_n\rangle = \epsilon_n|\phi_n\rangle \tag{3}$$

$$H^{(h)}_{SP}|\bar{\phi}_m\rangle = \bar{\epsilon}_m|\bar{\phi}_m\rangle. \tag{4}$$

If the form of the interaction V and SP Hamiltonians are known in a given representation, say $|x\rangle$, then the interaction elements can be obtained from

$$V^{(he)}_{\substack{m_1,n_2\\n_3,m_4}} = \langle\bar{\phi}_{m_1}\phi_{n_2}|V|\phi_{n_3}\bar{\phi}_{m_4}\rangle = \tag{5}$$

$$= \sum_x \sum_{x'} \langle\bar{\phi}_{m_1}|x\rangle\langle\phi_{n_2}|x'\rangle V^{(he)}_{x,x'}\langle x'|\phi_{n_3}\rangle\langle x|\bar{\phi}_{m_4}\rangle. \tag{6}$$

Similarly for (ee) and (hh). Here the interaction $V_{x,x'}$ has two indexes alone since it is diagonal on $|x, x'\rangle$. In case the interaction is the electrostatic one, the above expression gives the *Coulomb elements*. Note that x is a cumulative index containing all the coordinates of the representation, as real-space position, spin, etc.

Let us suppose now that the SP states can be factorized into two components (same for holes),

$$\langle x|\phi_n\rangle = \langle \mathbf{r}|\psi_{p(n)}\rangle\langle q'|\omega_{q(n)}\rangle, \tag{7}$$

with $|x\rangle = |\mathbf{r}\rangle|q'\rangle$, and that the SP Hamiltonian is separable into the two corresponding hamiltonians depending on $\mathbf{r}$ and q', respectively. For simplicity we use a specific example, with the first component only depending on the real-space coordinates, and the second component that does not need to be computed explicitly in a different representation (e.g. spin states). The latter means that we use the representation q' with the basis consisting of the vectors $|\omega_q\rangle$ themselves. As a consequence, $\langle q'|\omega_{q(n)}\rangle = \delta(q'-q)$. In other words, the above value is a function of q' (q is fixed by n) different from zero only when $q' = q$.
In this case, the Coulomb elements (with an interaction depending on $\mathbf{r}$ alone) are given by

$$\begin{aligned} V^{(he)}_{\substack{m_1,n_2\\ n_3,m_4}} &= \langle\bar{\psi}_{\bar{p}(m_1)}\psi_{p(n_2)}|V^{(he)}_{\mathbf{r}-\mathbf{r}'}|\psi_{p(n_3)}\bar{\psi}_{\bar{p}(m_4)}\rangle\,\langle\bar{\omega}_{\bar{q}(m_1)}\omega_{q(n_2)}|\omega_{q(n_3)}\bar{\omega}_{\bar{q}(m_4)}\rangle & (8)\\ &= \int d\mathbf{r}\int d\mathbf{r}'\,\bar{\psi}^*_{\bar{p}(m_1)}(\mathbf{r})\psi^*_{p(n_2)}(\mathbf{r}')V^{(he)}(\mathbf{r}-\mathbf{r}')\psi_{p(n_3)}(\mathbf{r}')\bar{\psi}_{\bar{p}(m_4)}(\mathbf{r}) & (9)\\ &\quad \delta\left[\bar{q}(m_1)-\bar{q}(m_4)\right]\,\delta\left[q(n_2)-q(n_3)\right], \end{aligned}$$

where each of the indexes $p, \bar{p}, q, \bar{q}$ can represent a whole set of quantum numbers. Since the interaction operator V is independent from q, the δ's have appeared. Note that $p = p(n)$ and $q = q(n)$, as can be gathered from Eq. (7), i.e. a single value of p or q is determined by a value of n but the opposite does not hold. That is, you can have the same wave function ψ_p for different values of n.

Usually, you solve the SP real-space Schrödinger equations

$$\begin{aligned} H^{(e)}_{SP}(\mathbf{r})\psi_j(\mathbf{r}) &= f_j\psi_j(\mathbf{r}) & (10)\\ H^{(h)}_{SP}(\mathbf{r})\bar{\psi}_i(\mathbf{r}) &= \bar{f}_i\bar{\psi}_i(\mathbf{r}) & (11) \end{aligned}$$

and obtain a set (or two sets, considering holes) of wave functions and energies. Note that the energies f_j are never used alone, rather they are included in the SP-state energy ϵ_n, possibly including the $|\omega_q\rangle$ contribution (e.g. spin splitting). Let us call R and $\bar{R}$ the number of SP wave functions ψ_j used in the calculation: $j = 1\ldots R$ and $i = 1\ldots\bar{R}$.
The *Coulomb Integrals* are

$$U^{(he)}_{\substack{ij\\ kl}} = \int d\mathbf{r}\int d\mathbf{r}'\,\bar{\psi}^*_i(\mathbf{r})\psi^*_j(\mathbf{r}')\,\mathcal{U}^{(he)}(\mathbf{r}-\mathbf{r}')\,\psi_k(\mathbf{r}')\bar{\psi}_l(\mathbf{r}), \tag{12}$$

where the specific form of the Coulomb interaction $\mathcal{U}$ has been used in place of the generic V. The Coulomb elements must be obtained from

$$V^{(he)}_{\substack{m_1,n_2\\ n_3,m_4}} = U^{(he)}_{\substack{p(m_1)p(n_2)\\ p(n_3)p(m_4)}}\,\delta\left[\bar{q}(m_1)-\bar{q}(m_4)\right]\,\delta\left[q(n_2)-q(n_3)\right]\ , \tag{13}$$

Fed with the SP states and interaction elements, $\mathcal{CI}$tool computes the multi-particle states of $N^{(e)}$ electrons and $N^{(h)}$ holes as a linear combination of Slater determinants

$$|\Psi_n\rangle = \sum_{l_e} \sum_{l_h} C^n_{l_e,l_h} |\Phi^{(e)}_{l_e}\rangle |\Phi^{(h)}_{l_h}\rangle, \tag{14}$$

where $|\Psi_n\rangle$ is the n-th multi-particle state, $|\Phi^{(e)}_{l_e}\rangle$ and $|\Phi^{(h)}_{l_h}\rangle$ are the Slater determinants for electrons and holes, with $l_e = 1..L_e$ and $l_h = 1..L_h$, respectively.
We rewrite the above formula with a single sum, in order to represent the basis vector of the Hilber space with a single ket

$$|\Psi_n\rangle = \sum_{l=1}^{L} C^n_l |\Phi_l\rangle, \tag{15}$$

where L is the dimension of the Hilbert space, i.e. the number of double (electron + hole) Slater determinants considered. Here, $|\Phi_l\rangle = |\Phi^{(e)}_{l_e}\rangle |\Phi^{(h)}_{l_h}\rangle$, with l_e and l_h fixed by l.
The coefficients C^n_l are the output of $\mathcal{CI}$tool

In order to represent the electron (same for holes) density distribution of a multi-particle state, we consider the number-density operator in a coordinate point x

$$\hat{n}^{(e)}(x) = \sum_{s,s'=1}^{S} \phi^*_{s'}(x)\phi_s(x)\, e^\dagger_{s'} e_s, \tag{16}$$

where $\phi_s(x)$ is a SP state and e_s ($e^\dagger_s$) the annihilation (creation) operator of an electron in that state. S is the number of single-particle electron states.
In order to obtain the electron number-density in x for the state Ψ_n, we must compute the expectation value (the symbol $n^{(e)}$ is used for this real-valued function)

$$\begin{aligned} n^{(e)}(x) = \langle\Psi_n|\hat{n}^{(e)}(x)|\Psi_n\rangle &= \left[\sum_{l'=1}^{L} (C^n_{l'})^* \langle\Phi_{l'}|\right] \sum_{s,s'=1}^{S} \phi^*_{s'}(x)\phi_s(x) e^\dagger_{s'} e_s \left[\sum_{l=1}^{L} C^n_l |\Phi_l\rangle\right] \\ &= \sum_{l,l'=1}^{L} \sum_{s,s'=1}^{S} \left[(C^n_{l'})^* C^n_l \phi^*_{s'}(x)\phi_s(x) \langle\Phi^{(h)}_{l'_h}|\langle\Phi^{(e)}_{l'_e}|e^\dagger_{s'} e_s|\Phi^{(e)}_{l_e}\rangle|\Phi^{(h)}_{l_h}\rangle\right] \\ &= \sum_{l,l'=1}^{L} \left[(C^n_{l'})^* C^n_l\, \delta[l'_h - l_h] \sum_{s,s'=1}^{S} \phi^*_{s'}(x)\phi_s(x) \langle\Phi^{(e)}_{l'_e}|e^\dagger_{s'} e_s|\Phi^{(e)}_{l_e}\rangle\right], \end{aligned}$$

where, $l_e = l_e(l)$ and $l_h = l_h(l)$ and δ is a Kronecker delta coming from the orthogonality of the hole Slater determinants. In the above derivation, x represent a full set of coordinates for the SP state.

If one is interested in the density for a subset of coordinates, like $\mathbf{r}$ –see Eq. (7)– the number-density operator is

$$\begin{aligned}\hat{n}^{(e)}(\mathbf{r}) &= \sum_{q'} n^{(e)}(\mathbf{r},q') = \sum_{q'}\sum_{s,s'=1}^{S} \psi^*_{p(s')}(\mathbf{r})\psi_{p(s)}(\mathbf{r})\,\omega^*_{q(s')}(q')\omega_{q(s)}(q')\,e^\dagger_{s'}e_s \qquad (17)\\ &= \sum_{s,s'=1}^{S} \psi^*_{p(s')}(\mathbf{r})\psi_{p(s)}(\mathbf{r})\,\delta[q(s')-q(s)]\,e^\dagger_{s'}e_s\,.\end{aligned}$$

Now one can compute the real-space density

$$\begin{aligned}n^{(e)}(\mathbf{r}) &= \langle\Psi_n|\left[\sum_{s,s'=1}^{S} \psi^*_{p(s')}(\mathbf{r})\psi_{p(s)}(\mathbf{r})\,\delta[q(s')-q(s)]\,e^\dagger_{s'}e_s\right]|\Psi_n\rangle \qquad (18)\\ &= \sum_{l,l'=1}^{L}\left[(C^n_{l'})^* C^n_l\,\delta[l'_h-l_h]\sum_{s,s'=1}^{S}\psi^*_{p(s')}(\mathbf{r})\psi_{p(s)}(\mathbf{r})\,\delta[q(s')-q(s)]\langle\Phi^{(e)}_{l'_e}|e^\dagger_{s'}e_s|\Phi^{(e)}_{l_e}\rangle\right].\end{aligned}$$

In the following, the input/output files for the software suite will be described. The configuration-interaction code $\mathcal{CI}$tool , the density code, the single-particle code and the Coulomb integral code will be considered although the last two programs are not part of $\mathcal{CI}$tool distribution: they are needed in order to prepare $\mathcal{CI}$tool input files.
Empty circles ∘ indicate files not directly used by $\mathcal{CI}$tool or by the density code (part of the $\mathcal{CI}$tool distribution), while bullets • indicate files used/produced by the $\mathcal{CI}$tool codes. (ASC) are "formatted" plain text files, (BIN) are "unformatted Fortran binary files.

Single-particle states

This is the code that computes the SP wave functions $\psi_j(\mathbf{r})$ and writes the SP states (in terms of quantum numbers) and energies: ϕ_n and ϵ_n. It is not part of $\mathcal{CI}$tool . We will use stic2Dhexagon as an example. It has the following i/o files.
∘ INPUT (ASC) : stic2Dhexagon.nml
It is the namelist describing the physical structure and its discretization via a real-space mesh. Its name is hard-coded. It can indicate additional INPUT files describing the structure itself, as the material profile.
∘ OUTPUT (BIN) : psi_e.bin
It is the "unformatted" file containing the SP electron wave functions $\psi_j(\mathbf{r})$. A similar or the same code is needed for holes, producing OUTPUT (BIN) : psi_h.bin. Since it is not used by $\mathcal{CI}$tool and, it is highly dependent on the physical problem and on the mesh, its format is not defined. In fact, it is used by the Coulomb-elements code (also not part of $\mathcal{CI}$tool) and

Table 1 Scrap of SP-states code for writing psi_e.bin. The variables numh, numev, dh are: number of grid points in a certain direction, number of SP eigenfunctions included, distance between two grid points (or something similar, since here we have an hexagonal mesh).

```
INTEGER :: numh, numev
REAL*8 :: dh
REAL*8, ALLOCATABLE :: psi(:,:,:)
ALLOCATE(psi(-numh:numh, -numh:numh, numev))
 ...
OPEN(32, FILE=fileoutBIN_psi_e, FORM="UNFORMATTED")
WRITE(32) numh, numev, dh
WRITE(32) psi
```

by the Density code, through the user-developed module `mod_inoutrs.f90` (see later). In general, it should contain infos on the mesh and the real-space wave functions $\psi_1, \psi_2, \dots \psi_S$. As an example, stic2Dhexagon writes it with the code reported on Table 1.

• OUTPUT (ASC) : spstates.dat

It contains the description – in terms of quantum numbers (QN) – and the energies ϵ_n of the SP states $|\phi_n\rangle$. For electrons it has the following format (Table 2 shows an example with the whole file and with spaces substituted by + signs.)

```
 E
          24           2
NAME:          rank         spin
TYPE:           psi          ext
   1             1            0  0.942294659338679
   2             2            0  0.942558787377977
 ...
```

The first character indicates the type of particles (E or e for electrons, H or h for holes). In the second line, the first number is the number of SP states $|\phi_n\rangle$, namely S, and the second is the number of QNs used to label them. In the third line, after the string "NAME: ", the names of the QNs are reported, with a max length of 12 characters. In the following line, the type of QNs is reported. Both name and type refer to the QNs in the same column. The type can be

psi : the QN labels the ψ_j wave functions, i.e. it is the index j;

ext : the QN labels the ω_q component, i.e. it is the index q;

int : the QN is already inside ψ_j, i.e. it represents a QN that can be gathered from $\psi_j(\mathbf{r})$ – this QN is only used as a label to possibly limit the ψ_j set.

The following lines contain, in order, the index n of $|\phi_n\rangle$, the values of the QNs named above and the energy of the SP-state ϵ_n. Note that the energy can be expressed in any unit (e.g.

Table 2 Example of the file spstates.dat with spaces sustituted by + signs

```
+E
++++++++++24++++++++++2
NAME:+++++++++rank++++++++spin
TYPE:++++++++++psi+++++++++ext
+++1++++++++++++1++++++++++0++0.942294659338679
+++2++++++++++++2++++++++++0++0.942558787377977
+++3++++++++++++3++++++++++0++0.942558787377981
+++4++++++++++++4++++++++++0++0.943290473090732
+++5++++++++++++5++++++++++0++0.943290473090738
+++6++++++++++++6++++++++++0++0.943933118334573
+++7++++++++++++7++++++++++0++0.946407009705940
+++8++++++++++++8++++++++++0++0.947625130416417
+++9++++++++++++9++++++++++0++0.947625130416417
++10++++++++++++10++++++++++0++0.950050742466169
++11++++++++++++11++++++++++0++0.950050742466169
++12++++++++++++12++++++++++0++0.952342721210452
++13++++++++++++1++++++++++1++0.942294659338679
++14++++++++++++2++++++++++1++0.942558787377977
++15++++++++++++3++++++++++1++0.942558787377981
++16++++++++++++4++++++++++1++0.943290473090732
++17++++++++++++5++++++++++1++0.943290473090738
++18++++++++++++6++++++++++1++0.943933118334573
++19++++++++++++7++++++++++1++0.946407009705940
++20++++++++++++8++++++++++1++0.947625130416417
++21++++++++++++9++++++++++1++0.947625130416417
++22++++++++++++10++++++++++1++0.950050742466169
++23++++++++++++11++++++++++1++0.950050742466169
++24++++++++++++12++++++++++1++0.952342721210452
```

Table 3 Scrap of SP-states code for writing spstates.dat.

```
WRITE(32,*) "E"
WRITE(32,*) 2*numev, 2
! name of quantum numbers (CHANGE THE FORMAT TO nA12 FOR n QN)
WRITE(32,"(A6,2A12)") "NAME: ", "rank", "spin"
! type of quantum numbers (CHANGE THE FORMAT TO nA12 FOR n QN)
WRITE(32,"(A6,2A12)") "TYPE: ", "psi",  "ext"
! SP quantum numbers and energy (in eV !) for each state
DO ne= 1, numev
  WRITE(32,"(I4,XX)",ADVANCE="NO") ne
  WRITE(32,"(2I12)",ADVANCE="NO") ne, 0
  WRITE(32,*) energy(ne) - spinenergysplit/2  ! in eV
END DO
DO ne= 1, numev
  WRITE(32,"(I4,XX)",ADVANCE="NO") ne+numev
  WRITE(32,"(2I12)",ADVANCE="NO") ne, 1
  WRITE(32,*) energy(ne) + spinenergysplit/2  ! in eV
END DO
```

eV, Joule, Hartree), however it must be consistent with the unit of the interaction elements. The resulting energies of the MP states, produced by $\mathcal{CI}$tool , will be in the same unit.
The file spstates.dat is written with the code reported on Table 3.

Coulomb elements

This code computes the Coulomb elements $V^{(ee)}_{\substack{n_1,n_2\\n_3,n_4}}$ $V^{(hh)}_{\substack{m_1,m_2\\m_3,m_4}}$ and $V^{(he)}_{\substack{m_1,n_2\\n_3,m_4}}$ from the ψ_j wave functions computed by the previous code. It is not part of $\mathcal{CI}$tool . We will use coulombel for an hexagonal grid as an example. It has the following i/o files.
◦ INPUT (ASC) : coulombel.nml
It is the namelist with some infos on the SP wave functions and on the physical system Its name is hard-coded. It also indicates the name of the other INPUT/OUTPUT files.
• OUTPUT (ASC) : Vee.dat Vhh.dat Vhe.dat
These files contain the Coulomb elements for electron-electron, hole-hole, hole-electron interactions, needed by $\mathcal{CI}$tool . Be careful to put in them the Coulomb elements $V^{(ee)}_{\substack{n_1,n_2\\n_3,n_4}}$ and not the Coulomb integrals $U^{(ee)}_{\substack{ij\\kl}}$. This will be changed in a future version. These files have the following structure:

```
      12944
  1   2      1    2      2.438049847439864E-003
```

Table 4 Scrap of Coulomb-elements code for writing Vee.dat.

```
TYPE ci_type_real8
  INTEGER*1  ::  n1, n2, n3, n4
  REAL*8 ::  v
END TYPE ci_type_real8
TYPE( ci_type_real8 ), ALLOCATABLE :: ce_ee(:)
ALLOCATE(ce_ee(numce_ee))
 ...
WRITE(21,*) numce_ee
DO nn= 1, numce_ee
  WRITE(21,"(1X,2(I3,1X),2X,2(I3,1X),2X)",ADVANCE="NO")   &
       &  ce_ee(nn)%n1, ce_ee(nn)%n2, ce_ee(nn)%n3, ce_ee(nn)%n4
  WRITE(21,*)  ce_ee(nn)%v
END DO
```

```
   1   2     1  10     9.859311940431132E-005
   1   2     1  11    -2.040113596402228E-004
 ...
```

with the first number indicating how many Coulomb elements are available in the file, and the following lines containing the four indexes and the value (energy-like) of the interaction element. Specifically, the four integers in each line are n_1, n_2, n_3 and n_4, in this very same order. In the code coulombel, the file Vee.dat is written with the code reported on Table 4

$\mathcal{CI}$tool

The full-configuration-interaction solver. Using the SP-states energies and interaction elements it computes the correlated multi-particle (MP) state of $N + \bar{N}$ electrons and holes on the basis of Slater determinants. It has the following i/o files.

• INPUT (ASC) : citool.nml

Namelist indicating the number of electrons and holes, the number of SP states (must match with S and $\bar{S}$ used in the previous calculations), the names of in/out files. An example is reported in Table 5 and a brief explanation of the keywords is reported in Table 6.

• INPUT (ASC) : spstates.dat

See above.

• INPUT (ASC) : Vee.dat Vhh.dat Vhe.dat

See above.

• `INPUT (ASC) : constrains_e.dat`

It is the optional file describing constrains on the Hilbert space according sums of SP quantum numbers. Its name is given in citool.nml via the `filein_hconstrains_e` string. Only

Table 5 Example of the file namelist citool.nml.

```
&indata_singleparticle
                numspstates_e= 24 ,
                numspstates_h= 0            /

&indata_multiparticle
                num_e= 3 ,
                num_h= 0 ,
                nummpenergies= 6 ,
                nummpstates= 6 ,
                complexrun= .FALSE.              /


&indata_inoutput
                citoolnml_version= "0.9" ,
                filein_spstates_e= "spstates.dat" ,
                filein_spstates_h= "input_1h.dat" ,
                fileinformat_coulomb= "dat real8" ,
                filein_coulomb_ee= "Vee.dat" ,
                filein_coulomb_hh= "Vhh.dat" ,
                filein_coulomb_eh= "Veh.dat" ,
                filein_hconstrains_e = "constrains_e.dat" ,
                fileoutBIN_hspace= "hspace.bin" ,
                fileoutASC_hspace= "hspace.txt" ,
                fileoutBIN_mpstates= "mpstates.bin" ,
                cutoff_fileoutBIN_mpstates = 0. ,
                fileoutASC_mpstates= "mpstates.txt" ,
                cutoff_fileoutASC_mpstates = 1e-6 ,
                loglevel= 0 ,
                statusfile= "status.txt",
                runname= "test20dec10.2"             /
```

Table 6 Keywords in citool.nml file for $\mathcal{CI}$tool version 0.91

`numspstates_e`	number of electrons SP states S
`numspstates_h`	number of holes SP states $\bar{S}$
`num_e`	number of electrons
`num_h`	number of holes
`nummpenergies`	number of multi-particle energy levels to be computed
`nummpstates`	number of multi-particle states to be computed
`complexrun`	are the interaction elements complex ?
`citoolnml_version`	version of this nml: must match the code version
`filein_spstates_e`	name of file with SP states for electrons
`filein_spstates_h`	name of file with SP states for holes
`fileinformat_coulomb`	format of file with interaction elements - it contains: `dat|cit` for ascii or CItool binary format `real8|complex16` for real or complex double prec.
`filein_coulomb_ee`	name of file with the ee interaction elements
`filein_coulomb_hh`	name of file with the hh interaction elements
`filein_coulomb_eh`	name of file with the eh interaction elements
`filein_hconstrains_e`	name of file with description of constrains
`fileoutBIN_hspace`	unformatted file with the Hilbert space (Slater dets)
`fileoutASC_hspace`	formatted file with the Hilbert space
`fileoutBIN_mpstates`	unformatted file with the resulting multi-particle states
`cutoff_fileoutBIN_mpstates`	– not implemented: must be 0 –
`fileoutASC_mpstates`	formatted file with the resulting multi-particle states
`cutoff_fileoutASC_mpstates`	only Slater dets weight$\geq$cutoff are included in file
`loglevel`	level of infos in statusfile: with 0 maximum info
`statusfile`	name of logfile
`runname`	name of this run: is included in the output files

available for electrons at the moment. In this file, you specify a number of subsets of the total Hilbert space, each containing a restricted number of Slater determinants. The Slater determinants included in each subset are determined by the *sum* of SP quantum numbers. The Hilbert subspaces of different constrains can overlap and do not need to fully cover the total Hilbert space.
The constrains file has the following structure:

```
 E
           2           2
NAME:          rank         spin
TYPE:           psi          ext
   1              *            *           *           *
   2              *            0           3           3
 ...
```

that is similar to spstates.dat except for the energy column, not present here, and two extra 12-position columns with two integers (without `NAME` or `TYPE` entry). Starting from scratch, one can take the spstates.dat file and use it as a template, removing the energies and introducing the two extra integers. The first character, "E" or "H", indicates if the following constrains are on the electron or hole quantum numbers, respectively. The two integers in the second line are the number of constrains and the number of quantum numbers. Note that constrains in excess to the first number will not be considered. The number of quantum numbers must match the one in spstates.dat. In the third and fourth line the name and type of quantum numbers are reported (see the description of spstates.dat above). In the following lines the constrains are described, with the character "∗" meaning "any". Let us consider a single constrains line. If a number is present in a field corresponding to a given quantum number, then the Slater determinants that will be included in the basis need to have populated SP states such that the sum of the specific quantum number matches the given value. Let us take, as an example, the constraints 2 reported above. While no condition is present for the rank quantum number (∗= any), only Slater determinants where the sum of spin quantum numbers of occupied SP states is 0 are included in the basis. This correspond to fix the total spin as far as the interaction is spin independent. Be careful that this is true only with total quantum numbers that are the sum of SP quantum numbers. Also note that the number 0 in the spin constrain does not mean total S=0, rather the total S you are imposing depends on how you codified the spin quantum number in spstates.dat. For example, if you decided spin-up=1 and spin-down=0 (as in Table 2), then a constrain with spin value 0 (as in the above example) means that the sum of SP spins must be zero, i.e. only electrons with spin down are allowed in a Slater determinants for this constrain. In general, the constrain is dependent on the number of particles. For example, with a constrain spin value 1 and a run with two electrons, you are imposing that Slater determinants must have one spin-up and one spin-down electron, while in a run with five electrons you will have one spin-up and four spin-down electrons. Again, this depends on the SP quantum number codification.

The two last integers in a constrain line, after the requested sum of SP quantum numbers, indicate how many multi-particle eigenenergies and eigenstates must be computed for that constrain. The maximum numbers allowed are nummpenergies and nummpstates specified in citool.nml. The above values are also the default, used to replace *. So, in the first constrain of the example above, nummpenergies eigenvalues and nummpstates eigenvectors will be computed, while in the second constrain, 3 eigenvalues and 3 eigenvectors will be computed.
If no `filein_hconstrains_e` is specified in citool.nml, no constrain is imposed, and the full Hilbert space of every possible Slater determinant is used. This is equivalent to have a constrain with * in all fields.
The file `constrains_e.dat` is read by $\mathcal{CI}$tool with a code like the one reported on Table 7.
• `OUTPUT (BIN) : hspace.bin`
Contains the Hilbert space (or spaces, if more constrains are used) on which the multi-particle states are calculated and represented. Each basis vector is a couple of Slater determinants (one for electron states and one for hole states), each encoded in a 68-bit (i.e. Fortran INTEGER*8) integer. The internal structure of this unformatted file can be guessed from the code in Table 8. This file is needed by the density code, and, in general, to interpret the unformatted file with the multi-particle states since it does not contain the basis.
• `OUTPUT (ASC) : hspace.txt`
Same as above, in plain text.
• `OUTPUT (BIN) : mpstates.bin`
Contains the result, i.e. the multi-particle states expressed in terms of the coefficients C_l^n of Eq. 15. Each coefficient is a REAL*8 (or COMPLEX*16 if `complexrun= .TRUE.` in citool.nml). The internal structure of this unformatted file can be guessed from the code in Table 9. This file is needed by the density code, together with the file with the basis hspace.bin.
• `OUTPUT (ASC) : mpstates.txt`
Same as above, in plain text.

Particle density distribution

This is the code that computes the particle density as a function of a given coordinate set according Eq. (18). It is named density4$\mathcal{CI}$tool and it is part of $\mathcal{CI}$tool .
It has the following i/o files:
• INPUT (ASC) : density4CItool.nml
Namelist with the description of the MP state(s) for which the density distribution will be computed and other infos. The name of this file is hard-coded. An example of this namelist is reported on Table 10 and a brief description of the variables is given on Table 11
In order to decide the density of which MP(s) you want, you first need to set the specific constrain (see the $\mathcal{CI}$tool section) containing the MP state. If the run was performed without explicit constrains, put `want_cons=1`. then, you can choose the MP state either via its position in the increasing-energy list (the status file reports this list for the selected constrain)

Table 7 Scrap of $\mathcal{CI}$tool code for reading the constrains file constrains_e.dat.

```
INTEGER, ALLOCATABLE :: hcons(:,:)
CHARACTER(LEN=12) :: namespqn, typespqn, cons
READ(31,*) partype
READ(31,*) numhcons, numspqn
ALLOCATE(hcons(numhcons,numspqn+2))
READ(31,"(A6)",ADVANCE="NO") string6
IF (string6/="NAME: " .AND. string6/="name: ") STOP "error"
DO nqn= 1, numspqn
  READ(31,"(A12)",ADVANCE="NO") namespqn
END DO
READ(31,*)
READ(31,"(A6)",ADVANCE="NO") string6
IF (string6/="TYPE: " .AND. string6/="type: ") STOP "error"
DO nqn= 1, numspqn
  READ(31,"(A12)",ADVANCE="NO") typespqn
END DO
READ(31,*)
DO nc= 1, numhcons
  READ(31,"(I4,XX)",ADVANCE="NO") nc
  DO nqn= 1, numspqn + 2
    READ(31,"(A12)",ADVANCE="NO") cons
    IF (TRIM(ADJUSTL(cons)) /= "*") THEN
      READ(cons,*) hcons(nc,nqn)
    END IF
  END DO
  READ(31,*)
END DO
```

Table 8 Scrap of $\mathcal{CI}$tool code for writing the unformatted Hilbert space file hspace.bin.

```
CHARACTER(80), PARAMETER :: citool_version= "0.9"
CHARACTER(80) :: runname = "citoolrun"
INTEGER, ALLOCATABLE :: blockstart(:)
INTEGER*8, ALLOCATABLE :: ket(:,2)
OPEN(22, FILE=TRIM(fileoutBIN_hspace), ACTION="WRITE", FORM="UNFORMATTED")
WRITE(22) citool_version
WRITE(22) runname
WRITE(22) dimhspace_e, dimhspace_h
WRITE(22) numhcons_e, numhcons_h
DO ncons_e= 1, numhcons_e
DO ncons_h= 1, numhcons_h
  WRITE(22) ncons_e, ncons_h
  WRITE(22) dimhspacecons
  WRITE(22) numblock
  WRITE(22) blockstart(1:numblock+1)
  WRITE(22) ket
END DO
END DO
```

Table 9 Scrap of $\mathcal{CI}$tool code for writing the unformatted file mpstates.bin, with the computed multi-particle states.

```
CHARACTER(80), PARAMETER :: citool_version= "0.9"
CHARACTER(80) :: runname = "citoolrun"
REAL*8, ALLOCATABLE :: mpstates(:,:)
REAL*8, ALLOCATABLE :: mpenergies(:,:)
OPEN(22, FILE=TRIM(fileoutBIN_mpstates), ACTION="WRITE", FORM="UNFORMATTED")
WRITE(22) citool_version
WRITE(22) runname
WRITE(22) cutoff_fileoutBIN_mpstates
WRITE(22) dimhspace_e, dimhspace_h
WRITE(22) numhcons_e, numhcons_h
DO ncons_e= 1, numhcons_e
DO ncons_h= 1, numhcons_h
  WRITE(22) ncons_e, ncons_h
  WRITE(22) dimhspacecons
  WRITE(22) numblock
  WRITE(22) nummpenergiescons, nummpstatescons
  DO nb= 1, numblock
    WRITE(22) mpenergies
    WRITE(22) blocknummpenergies
    WRITE(22) mpstates_x
    WRITE(22) blocknummpstates
  END DO
END DO
END DO
```

Table 10 Example of the file namelist density4CItool.nml.

```
&indatadensity_wantmpstate
                want_cons= 1 ,
                want_energylevel= 3 ,
                want_block= 5, 5 ,
                want_rank=  1, 3              /

&indatadensity_inoutput
                density4citoolnml_version= "0.9" ,
                fileinBIN_psi_e= "psi_e.bin" ,
                fileinBIN_psi_h= "" ,
                filein_densdescription_e= "densdescription_e.dat" ,
                filein_densdescription_h= ""     /
```

Table 11 Keywords in density4citool.nml file for $\mathcal{CI}$tool version 0.91

`want_cons`	number of the constrain containinf the MP state(s)
`want_energylevel`	n. of the MP state (in increasing energy order)
`want_block`	if `want_energylevel=0` this select one or more MP block
`want_rank`	if `want_energylevel=0` this select one or more MP rank
`density4citoolnml_version`	ver. of this file: should match `citoolnml_version`
`fileinBIN_psi_e`	"unformatted" file with the electron SP wave functions
`fileinBIN_psi_h`	"unformatted" file with the hole SP wave functions
`filein_densdescription_e`	describes how many and which densities to compute
`filein_densdescription_h`	same, for holes

or via its block and rank number. In the first case put `want_energylevel=1` for the ground state, `want_energylevel=2` for the first-excited state, and so on. The following two variables will be ignored. In the second case, namely to choose the MP state(s) through its(their) block/rank, put `want_energylevel=0`. Then you can select one or more (up to 20) MP states by setting the `want_block` and `want_rank` variables. They are integers arrays, so you can put one or more (up to 20) integers values in the namelist. For example, on Table 10 two MP states are selected, namely (block,rank)= (5,1) and (5,5). Note that in order to make this selection effective one should set `want_energylevel=0` on Table 10. When more than one MP state is selected, each calculated density will be the average of the density of each MP state. So, in the example of Table 10, each density file (as listed in "densdescription_e.dat", see below) would contain the sum of the density of the two MP states selected, divided by 2.

- INPUT (ASC) : citool.nml

The code density4$\mathcal{CI}$tool reads the same namleist of the configuration-interaction run. This file should be unaltered from the $\mathcal{CI}$tool run that produced the results whose density will be computed.

- INPUT (ASC) : densitydescription_e.dat

In this file you list the kind of electron densities you want, by possibly listing only specific values of SP quantum numbers and giving names to the output files. The file with the description of the densities has the following structure:

```
 E
           3           2
NAME:          rank         spin
TYPE:           psi          ext
   1              *            *        densTOTe.hdat
   2              *            1        densUPe.hdat
   3              *            0        densDNe.hdat
```

that is similar to spstates.dat except for the energy (last) column, here substituted by a 80-character string, containing the file name of the density described bu the quantum numbers on the same line. The first four lines are read with a code similar to the first part of Table 7. The remaining lines, with the density descriptions, are read with the code sketched on Table 12, where numdensdesc is the number of such lines.

In densitydescription_e.dat, the character in the first line indicates the type of particles (E or e for electrons, H or h for holes). In the second line, the first number is te number of densities to be computed, the second is the number of quantum numbers describing a SP state. The meaning of the third and fourth lines have been already described (see spstates.dat. In the following lines you can describe the densities you are interested in. The first number is only an integer label, starting from 1 and increasing by one each line. then, in the integers describing the quantum number you can put $*$ in order to include all the SP states for this density (also in this case $*$ ="any"), or you can specify a value. In the latter case only the SP states with that value for that particular quantum number will be considered. In the

Table 12 Scrap of density4$\mathcal{CI}$tool code for reading the density-description file densitydescription_e.dat.

```
INTEGER :: densdesc(numdensdesc,numspqn)
CHARACTER(80) :: densfiles(numdensdesc)
densdesc(:,:)= 9999
DO nd= 1, numdensdesc
  READ(31,"(I4,XX)",ADVANCE="NO") nd_read
  IF ( nd_read /= nd )  STOP "DENSDESCRIPTION: densdesc wrong #"
  DO nqn= 1, numspqn
    READ(31,"(A12)",ADVANCE="NO") dede_read
    IF (TRIM(ADJUSTL(dede_read)) /= "*") THEN
      READ(dede_read,*) densdesc(nd,nqn)
    END IF
  END DO
  READ(31,*) densfiles(nd)
END DO
```

example of Table 12 the first description line computes the total electron density (i.e. with all the SP states included), and the following lines compute the spin-UP and spin-DOWN densities, respectively, since the spin quantum number is fixed to 1 (UP) or 0 (DOWN) so that only the proper SP states are used.

• `OUTPUT (ASC) : status.txt`

The code density4$\mathcal{CI}$tool uses the same log file of $\mathcal{CI}$tool , whose name is read from citool.nml.

• `FORTRAN SOURCE (F90) :` mod_inoutrs.f90

This is not a true i/o file, but the used must provide it in order to read the file textsfpsi_e.bin and to write the density files in a user-specified format. It must contain a module `MODULE mod_inoutrs` and two subroutines, namely `INSPWF` and `OUTDENS`.

`SUBROUTINE INSPWF(numspwf, numx, psi, filename)` fills the 2D `psi` array. Its first index labels the real-space mesh point $\mathbf{r}$, its second index is the wave-function number j, see Eq. (10). `numx` and `numspwf` are the number of mesh points and SP wave functions, respectively. `filename` is the name of file containing the wave functions and is specified in the density-code namelist.

`SUBROUTINE OUTDENS(numx, dens, filename, denssum)` writes the content on the 1D array `dens` (with `numx` elements) on the file `filename`, specified in the density-code namelist. `denssum` is `REAL*8` result of the total integral of `dens`.

Note that the need to have a user-supplied mod_inoutrs.f90 will be changed in the following versions of $\mathcal{CI}$tool . A couple of examples are provided, for a rectangular 2D grid and a triangular grid (hexagonal domain).