

# TAURUS<sub>pav</sub>: Manual for the input files

21/05/2023

## 1 Structure of the standard input file (STDIN)

The input file is read by the code as STDIN and therefore has no fixed naming convention. On the other hand, the format of the file is fixed. We describe in this manual how to write a proper input file for TAURUS<sub>pav</sub> and the different options that the code offers. Because the input file is somewhat lengthy, we will present its different section separately (but remember that they are all part of the same file).

Before going further, a few remarks are in order:

- We use here the Fortran convention for the format of variables, e.g. `1i5` or `1a30`, assuming that the reader knows their meaning. If it is not the case, we encourage the reader to search for a tutorial somewhere else. There is a large body of Fortran documentation available online therefore it should not be too difficult.
- All lines start with the reading of an element of an array, `input_names` or `input_block`, made of `character(30)` variables. These variables only play a cosmetic/descriptive role when reading or printing the input parameters and can be set to any desired value by the user. Therefore, we are not going to comment them further.

### 1.1 Section about the Hamiltonian

#### Description

Line	Format	Data
1	1a	<code>input_names(1)</code>
2	1a	<code>input_names(2)</code>
3	1a30, 1a100	<code>input_names(3), hamil_file</code>
4	1a30, 1i1	<code>input_names(4), hamil_com</code>
5	1a30, 1i1	<code>input_names(5), hamil_read</code>
6	1a30, 1i5	<code>input_names(6), paral_teamssize</code>
7	1a	<code>input_names(7)</code>

where

- `hamil_file`: common name of the Hamiltonian input files. It is used by the code to determine all the possible hamiltonian files: `hamil_file.sho`, `hamil_file.01b`, `hamil_file.2b`, `hamil_file.com`, `hamil_file.red`
- `hamil_com`: option to take into account the center-of-mass correction when doing calculations with general Hamiltonians (`hamil_type = 3` or `4`).  
= 0 w/o center-of-mass correction.  
= 1 with center-of-mass correction, whose two-body matrix elements are read from `hamil_file.com`.
- `hamil_read`: option to read the Hamiltonian from the normal files or from a “reduced” file.  
= 0 reads the Hamiltonian from the normal Hamiltonian files (`hamil_file.sho`, `hamil_file.01b`, `hamil_file.2b`, `hamil_file.com`)  
= 1 reads the Hamiltonian from the reduced file `hamil_file.red` containing the *m*-scheme matrix elements written in binary (to be faster and have a smaller size). Note that the file

`hamil_file.sho` is still needed and read by the code. By contrast, `hamil_file.com` will not be read even if `hamil_com = 1`, so the center-of-mass correction has to be already included in the reduced file.

- **paral\_teamssize**: number of MPI processes in a team. Within a team, each process stores only a certain subset of the two-body matrix elements of the Hamiltonian and computes with them the fields  $h$ ,  $\Gamma$  and  $\Delta$ . Then they all synchronize through a `mpi_reduce`. This option is needed for large model spaces when the storage of the two-body matrix elements in the SHO basis becomes a problem.

= 0 or 1 all processes have access to all matrix elements.

> 1 the number of matrix elements managed by a process, `hamil_H2dim`, is obtained by performing the euclidean division of the total number of matrix elements, `hamil_H2dim_all`, by the number of processes within the team, `paral_myteamsize`. The rest of the division is distributed among the first processes in the team (i.e. the processes with a rank in the team, `paral_myteamrank`, strictly lower than the rest). Note that if `paral_teamssize` is not a divisor of the total number of MPI processes, `paral_worldsize`, then the last team will only be made of `modulo(paral_worldsize,paral_teamssize)` processes. Therefore, the actual size of the team is not necessarily equal to the input `paral_teamssize`. Obviously, `paral_teamssize` has to be lower than or equal to `paral_worldsize`.

## Example

```
Hamiltonian
-----
Master name hamil. files      usdb
Center-of-mass correction     0
Read reduced hamiltonian      0
No.  of MPI proc per H team   0
```

## 1.2 Section about miscellaneous paramters

### Description

Line	Format	Data
8	1a	input_names( 8)
9	1a	input_names( 9)
10	1a30, 1i1	input_names(10), misc_phys
11	1a30, 1i1	input_names(11), misc_part
12	1a30, 1i1	input_names(12), misc_oper
13	1a30, 1i1	input_names(13), misc_frot
14	1a30, 1es10.3	input_names(14), misc_cutrot
15	1a30, 1i1	input_names(15), seed_text
16	1a30, 1es10.3	input_names(16), seed_occeps
17	1a30, 1i1	input_names(17), seed_allem
18	1a	input_names(18)

where

- **misc\_phys**: option to select the physics case computed.
  - = 0 computes the observables related to the spectroscopy (energy, radii, electromagnetic transitions, ...).
  - = 1 computes the single-beta decay transitions. Not available in this version of the code.
  - = 2 computes the neutrinoless double beta decay nuclear matrix element (initial and final states have  $J^\pi = 0^+$ ). Not available in this version of the code.
- **misc\_part**: option to select the part of the calculation to perform.
  - = 0 computes all the observables.
  - = 1 computes only the norm overlap.
  - = 2 computes the norm overlap as well as the one-body observables. Does not compute the two-body observables (e.g. the energy).
- **misc\_oper**: option to read custom matrix elements of some operators from a file. See next section for more details.
  - = 0 the matrix elements of the operators are defined using the textbook definitions.
  - = 1 reads the matrix elements of the operators from files (if detected).
- **misc\_frot**: option to write/read the rotated matrix elements into/from a file.
  - = 0 does not write the matrix elements.
  - = 1 writes the rotated matrix elements.
  - = 2 reads the rotated matrix elements and performs the integration using them.
- **misc\_cutrot**: cutoff for rotated overlap  $\langle \Phi | R(\Omega) | \Phi \rangle$ .
  - > 0.0 the code will skip angles with a rotated overlap  $\langle \Phi | R(\Omega) | \Phi \rangle < \text{misc\_cutoff}$ .
  - = 0.0 the code will assume the value **misc\_cutoff** =  $10^{-16}$ .
- **seed\_text**: option to read the left and right wave functions as a binary (.bin) or text (.txt) file.
  - = 0 reads the files **left\_wf.bin** and **right\_wf.bin**.
  - = 1 reads the files **left\_wf.txt** and **right\_wf.txt**.
- **seed\_occeps**: cutoff to determine the fully occupied/empty single-particle states in the canonical basis. This is needed to compute the overlap using the Pfaffian method. We recommend changing this parameter only when observing serious numerical problems or inaccuracies.
  - > 0.0 the code will consider single-particles states with  $v^2 \geq 1.0 - \text{seed\_occeps}$  as fully occupied and the ones with  $v^2 \leq \text{seed\_occeps}$  as fully empty.
  - = 0.0 the code will assume the value **seed\_occeps** =  $10^{-8}$ .
- **seed\_allemp**: option to take into account the fully empty single-particle states in the calculation of the overlap. As for the previous option, we recommend to switch on this option only if you encounter numerical problems or if the code crashes with an error message linked to the evaluation of the overlap.
  - = 0 eliminates the maximum number of empty states possible.
  - = 1 takes into account all the empty states.

## Example

```

Miscellaneous
-----
Physics case studied           0
Part of the calc. performed   0
Read mat. elem. of operators  0
Write/read rotated mat. elm.  0
Cutoff for rotated overlaps    0.000E-00
Read wavefunctions as text     0
Cutoff occupied s.-p. states  0.000E-00
Include all empty sp states    0

```

## 1.3 Section about the particle number projection

### Description

Line	Format	Data
<b>19</b>	1a	input_names(19)
<b>20</b>	1a	input_names(20)
<b>21</b>	1a30, 1i5	input_names(21), valence_Z
<b>22</b>	1a30, 1i5	input_names(22), valence_N
<b>23</b>	1a30, 1i5	input_names(23), phiZ_dim
<b>24</b>	1a30, 1i5	input_names(24), phiN_dim
<b>25</b>	1a30, 1i5	input_names(25), phiA_dim
<b>26</b>	1a30, 1i1	input_names(26), pnp_nosimp
<b>27</b>	1a	input_names(27)

where

- **valence\_Z**: number of active protons. For no-core calculations, this is the total number of protons in the nucleus.
- **valence\_N**: same for the number of active neutrons.
- **phiZ\_dim**: number of points in the discretization (à la Fomenko) of the integral over proton gauge angles  $\phi_Z$ .  
 = 0 no particle-number projection is performed for this particle species. Note that this is equal to the case **phiZ\_dim** = 1.  
 > 0 and **odd** use a Fomenko discretization with the  $n$ -th point being located at the angle  $(\pi \text{ or } 2\pi)(n - 1)/(\text{phiZ\_dim})$ .  
 > 0 and **even** use a Fomenko discretization with the  $n$ -th point being located at the angle  $(\pi \text{ or } 2\pi)(n - 1/2)/(\text{phiZ\_dim})$
- **phiN\_dim**: same for the integral over neutron gauge angles  $\phi_N$ .
- **phiA\_dim**: same for the integral over nucleon gauge angles  $\phi_A$ .
- **pnp\_nosimp**: option to disable the simplifications related to the symmetries of the input wave functions.  
 = 0 simplifies the projections if a good quantum number is detected (e.g. no integral over  $\phi_Z$ )

tttvalence\_Z if one of the states has a good number of protons  $Z$ ).  
= 1 disables the simplifications.

### Example

Particle Number	
-----	
Number of active protons	4
Number of active neutrons	5
No. gauge angles: protons	7
No. gauge angles: neutrons	7
No. gauge angles: nucleons	7
Disable simplifications NZA	0

## 1.4 Section about the angular momentum projection

### Description

Line	Format	Data
<b>28</b>	1a	input_names(28)
<b>29</b>	1a	input_names(29)
<b>30</b>	1a30, 1i5	input_names(30), amp_2jmin
<b>30</b>	1a30, 1i5	input_names(30), amp_2jmax
<b>31</b>	1a30, 1i5	input_names(31), alpJ_dim
<b>32</b>	1a30, 1i5	input_names(32), betJ_dim
<b>33</b>	1a30, 1i5	input_names(33), gamJ_dim
<b>34</b>	1a30, 1i1	input_names(34), amp_nosimp
<b>35</b>	1a	input_names(35)

where

- **amp\_2jmin**: smallest value of the angular momentum considered.
- **amp\_2jmax**: largest value of the angular momentum considered.
- **alpJ\_dim**: number of points in the discretization of the integral over Euler angles  $\alpha_J$ .  
= 0 no projection on  $M_J$  is performed.  
> 0 use a Fomenko-like discretization with the  $n$ -th point being located at the angle  $2\pi(n - 1/2)/(\text{alpJ\_dim})$ .
- **betJ\_dim**: same for the integral over Euler angles  $\beta_J$  and  $J$ . In this case, the calculation is based on a Gauss-Legendre discretization of order **betJ\_dim**.
- **gamJ\_dim**: same for the integral over Euler angles  $\gamma_J$  and  $K_J$ .
- **amp\_nosimp**: option to disable the simplifications related to the symmetries of the input wave functions.  
= 0 simplifies the projections if a good quantum number is detected (e.g. no integral over  $\gamma_J$  if the right state has a good  $K_J$ ).  
= 1 disables the simplifications.

## Example

```
Angular Momentum
-----
Minimum angular momentum 2J  0
Maximum angular momentum 2J 12
No. Euler angles:  alpha    0
No. Euler angles:  beta     0
No. Euler angles:  gamma    0
Disable simplifications JMK  0
```

## 1.5 Section about the parity projection

### Description

Line	Format	Data
<b>36</b>	1a	input_names(36)
<b>37</b>	1a	input_names(37)
<b>38</b>	1a30, 1i1	input_names(38), parP_dim
<b>39</b>	1a30, 1i1	input_names(39), pap_nosimp
<b>40</b>	1a	input_names(40)

where

- **parP\_dim**: switch to project onto a good parity  $P$ .  
= 0 no projection on  $P$  is performed.  
= 1 projects on  $P = \pm 1$ .
- **pap\_nosimp**: option to disable the simplifications related to the symmetries of the input wave functions.  
= 0 simplifies the projections if a good quantum number is detected.  
= 1 disables the simplifications.

## Example

```
Parity
---
Project on parity P          0
Disable simplifications P    0
```

## 2 Other input files

In addition to the the STDIN, the code requires several input files:

- The wave function of the left quasi-particle state, written in the format of TAURUS<sub>vap</sub>, with the name `left_wf.bin`.
- The wave function of the right quasi-particle state, written in the format of TAURUS<sub>vap</sub>, with the name `right_wf.bin`.

- The hamiltonian files (.sho, .01b, .2b, .red) written in the same format as the one described in the manual of TAURUS<sub>vap</sub>.

Finally, the code can take as input optional files containing custom one-body and two-body matrix elements for certain operators:

- The files containing the one-body matrix elements of the electric multipole operators:  
matelem\_multipole\_QL\_1b.bin with L = 1, 2 or 3.
- The files containing the one-body matrix elements of the magnetic multipole operators:  
matelem\_multipole\_ML\_1b.bin with L = 1 or 2.
- The files containing the one- and two-body matrix elements of the square radius operator:  
matelem\_radius\_r2\_1b.bin and matelem\_radius\_r1r2\_2b.bin.

The precise format of these files can be determined by looking directly at the code TAURUS<sub>pav</sub>.