

RelaxSE

User's Manual

Elisa Rebolini¹ and Marie-Bernadette Lepetit^{1,2}

¹ Institut Laue Langevin, Grenoble, France

² Institut Néel, CNRS, Grenoble, France

June 18, 2021

1 Code description

The RelaxSE code does a fully-decontracted Multireference Configuration Interaction (MRCI) calculation using one of the following methods

- **SAS+S** : single excitations on all determinants of a reference space built as in the SAS+S method [1], that is
 - a selected set of determinants (included into a formal Complete Active Space), denoted as REF0,
 - all CAS \rightarrow CAS single excitations from all REF0 determinants,
 - all LIGO \rightarrow CAS single excitations from all REF0 determinants,
 - all CAS \rightarrow LIGV single excitations from all REF0 determinants,
 - all LIGO \rightarrow LIGV single excitations from all REF0 determinants,
- **MRSCI** : single excitations on all determinants of a selected reference space (included into a formal Complete Active Space);
- **MRSDCI** : single and double excitations on all determinants of selected reference space (included into a formal Complete Active Space);
- **CAS+S** : single excitations on all determinants of a Complete Active Space;
- **CAS+SD** : single and double excitations on all determinants of a Complete Active Space;
- **CAS+DDCI** : single and double excitations from the Difference Dedicated Configuration Interaction space [2–4], on all determinants of a Complete Active Space.

2 Code citation

Please cite the RelaxSE code as

Elisa Rebolini and Marie-Bernadette Lepetit, The Journal of Chemical Physics **154**, 164116 (2021).

3 Input description

3.1 Input files

The files required by the RelaxSEcode are the following.

<i>Fichiers</i>	<i>Contient</i>
<i>INPUT</i>	: the input file
<i>prefix.ref0</i>	: the REF0 short list of determinants. From each determinant in <i>prefix.ref0</i> all other determinants to get the desired spin configurations will be generated.
<i>prefix.TraOne</i>	: the <i>TraOne</i> frile issued from MOTRA.
<i>prefix.TraInt</i>	: the <i>TraInt</i> frile issued from MOTRA.
<i>prefix.restart</i>	: (optional) the wave-function file for restarts.

3.2 Output files

The files generated by the code are

<i>Fichiers</i>	<i>Contient</i>
<i>prefix.out</i>	: the output file.
<i>prefix.det</i>	: (optional) the list of determinants in human readable format.
<i>prefix.bdet</i>	: the list of determinants in binary format.
<i>prefix.sass</i>	: sass information file.
<i>prefix.restart</i>	: the wave-function file for restarts.

3.3 The *INPUT* File

This file is divided in several Fortran namelists. Let us remember that fortran is insensitive to case, except for file names. The character case in the following namelist are only for better understanding of the variable purpose. A Fortran namelist should be given as follow

```
&namelist_name  
  param1= x, param2=y, ...  
&end
```

The required namelists are the following ones

- SassInp
- InfomMolcasInp
- OrbInp

- VecInp
- DavidInp
- PropInp

SassInp : dedicated to general informations.

<i>Mot clef</i>	<i>Signification</i>
PREFIX	: prefix for all code files except <i>INPUT</i> .
PRINT_DET	: (optional, default <code>.false.</code>) to print the determinants list in <i>prefix.det</i> .
IPRINT	: (optional, default 0) printing options. <ul style="list-style-type: none"> • 0: no debug print • 1: print individual block timings • 2: print the Fock matrix
METHOD	: (optional, default SAS+S) method for determinant generation. <ul style="list-style-type: none"> • SAS+S : SASS method. • CAS+S : complete active space + single excitations on all determinants of the CAS. • CAS+SD : complete active space + single and double excitations on all determinants of the CAS. • CAS+DDCI : complete active space + single and double excitations belonging to the difference dedicated configuration interaction method.
SIZEBATCH	: (optional, default 40) size of blocks for MPI calculations. Recommended size for large calculations = number of OpenMP threads.
RESTART	: (optional, default <code>.false.</code>) to restart a calculation from a previous job, file <i>prefix.restart</i> needs to be present.

Typical exemple.

```
&sassinp
prefix="cuo",
sizebatch=20,
method="SAS+S"
&end
```

InfoMolcasInp : dedicated to provide informations on group symmetry, usually contained in the MOLCAS *RunFile*.

<i>Mot clef</i>	<i>Signification</i>
NIRREP	: number of irreducible representations in the symmetry group.
NTOT	: total number of orbitals.
IIRTB	: group multiplication table.
ICHTB	: group character table.

Typical exemple.

```

&infomolcasinp
nirrep=1,
ntot=118,
iIrTb(1,1) = 1,
iChTb(1,1) = 1
&end

```

OrbInp : dedicated to orbitals information.

<i>Mot clef</i>	<i>Signification</i>
NORB_GEL	: vector, number of frozen orbitals.
NORB_OCC	: vector, number of occupied orbitals.
NORB_LIGO	: vector, number of occupied bridging orbitals.
NORB_ACT	: vector, number of active/magnetic orbitals.
NORB_LIGV	: vector, number of virtual bridging orbitals.
NORB_VIRT	: vector, number of virtual orbitals.
NORB_DEL	: vector, number of deleted orbitals.

Typical exemple.

```

&orbinp
norb_gel = 2,
norb_occ = 38,
norb_ligo = 2,
norb_act = 2,
norb_ligv = 0,
norb_virt = 70,
norb_del = 4,
&end

```

VecInp : dedicated to information on the seaked vectors.

<i>Mot clef</i>	<i>Signification</i>
NREF0	: number of REF0 determinants to be read in <i>prefix.ref0</i> .
NELACT	: number of active/magnetic electrons.
NVEC	: number of desired eigenstates.
STOT	: $2S + 1$, S spin of the seaked states (default STOT=1). Used only to generate the complete REF0 determinants list.
SZ	: $2S_z$, the spin sector in wich the diagonalisation takes place.
VEC_IRREP	: (default 1) the irreducible representation of the seaked vectors.

Typical exemple.

```

&vecinp
stot=1,
sz=0,
vec_irrep=1,
nvec=2,
nref0=1,
nelact = 2
&end

```

DavidInp : (experts) dedicated to the Davidson procedure fine tuning.

<i>Mot clef</i>	<i>Signification</i>
SIZEHEFFDAVIDSON	: (optional, default 10) SIZEHEFFDAVIDSON \times NVEC is the size of the Davidson matrix. The code memory allocation should be at least $2 \times \text{SIZEHEFFDAVIDSON} \times \text{NVEC} \times 8$ bytes.
NITERDAVIDSON	: (optional, default 100) maximum number of Davidson iterations.
TOL_ORTH	: (optional, default 10^{-15}) Schmidt's orthogonalisation maximum error.
TOL_NORM	: (optional, default 10^{-15}) normalisation maximum error.
TOL_CONV	: (optional, default 10^{-8}) convergence criterium (energy equivalent).
ITER0	: (optional, default 0) last iteration number of the previous run in case of a restart.
CONV_ENER	: (optional, default .false.) activate the convergence on energy in addition to vectors.

Typical exemple.

```
&davidinp
&end
```

4 PROP input description

The PROP computes properties from the wave functions issued from the RelaxSE code.

It always prints out the projection of the RelaxSE wave functions on the CAS.

This is presently the only property implemented.

4.1 Input files

The files required by the PROP code are the following.

<i>Fichiers</i>	<i>Contient</i>
INPUT	: the input file (same as the RelaxSEcode + &propinp namelist
prefix.bdet	: the list of determinants in binary format.
prefix.sass	: sass information file.
prefix.restart	: (optional) the wave-function file for restarts.

4.2 Output files

The files generated by the code are

<i>Fichiers</i>	<i>Contient</i>
prefix.outprop	: the output file.

4.3 The *INPUT* File

This file is divided in several Fortran namelists. The same namelists as for the `RelaxSE` calculation and an additional namelist specific for properties calculations.

SassInp : identical to the `RelaxSE` calculation one.

InfoMolcasInp : identical to the `RelaxSE` calculation one.

OrbInp : identical to the `RelaxSE` calculation one.

VecInp : identical to the `RelaxSE` calculation one.

DavidInp : identical to the `RelaxSE` calculation one.

PropInp :

<i>Mot clef</i>	<i>Signification</i>
NPROP	: number of properties to be computed in addition to the printing of the <code>RelaxSE</code> wave functions projection on the CAS.
WHICHPROP	: list of the NPROP KEYWORDS of the properties to be computed

Typical exemple.

```
&propinp
nprop=2
whichprop = 'lcoef', 'dens'
&end
```

5 Code performances

The `RelaxSE` code uses a direct algorithm in order to treat large numbers of determinants (typically up to 10^8 , 10^9), i.e. each matrix element is recomputed on-the-fly.

The determinants are divided in 9 blocks, denoted $D_{\text{Nel}}^{\text{Nexc}}$, where `Nel` is the number of additional electrons in the active part with respect to the `REF0` configurations, and `Nexc` is the number of additional excitations outside of the active space. The `RelaxSE` code has both MPI and OpenMP parallelization driven both by integral and determinant blocks. The MPI parallelisation is done on the pair integral-kind / determinant-block. When the determinant blocks are very large, it is possible to divide the calculation into smaller MPI parts by batching the determinants. However, the gain is not systematic as the integrals are read from disk for each MPI process and the overhead can be large.

Within each MPI process an OpenMP parallelisation is done on the outer particle or hole loop.

For an optimal use of the code the user should adjust the number of MPI processes, OpenMP threads and the size of the batches according to her/his problem and computer system. In order to guide the user for such choices we provide below performance testing for the most typical cases. These data are taken from ref. [5].

Table 1: Orbital partitioning for the test calculations.

Test set	nocc	nligo	nact	nligv	nvirt	ndet
LIGO	49	2	8	0	140	30 267 828
	47	4	8	0	140	53 017 324
	45	6	8	0	140	74 811 684
	43	8	8	0	140	95 650 908
LIGV	51	0	8	2	138	30 721 372
	51	0	8	4	136	54 531 036
	51	0	8	6	134	77 992 188
	51	0	8	8	132	101 104 828

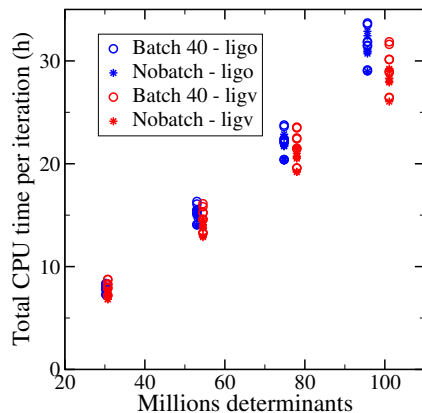


Figure 1: Total CPU-time as a function of the number of determinants for either occupied (ligo) or virtual (ligv) ligand orbitals. Calculations were performed with 4 or 6 MPI processes, 10 to 40 OpenMP threads and with (circles) and without (stars) batching of the determinants.

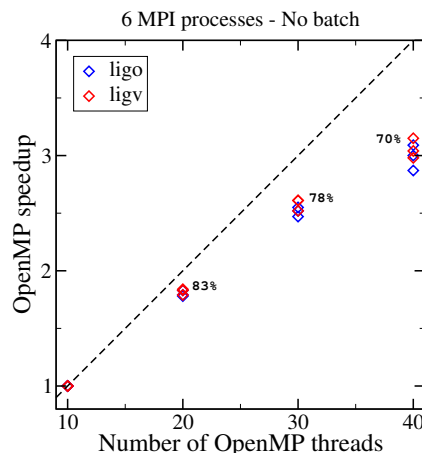


Figure 2: OpenMP speedup compared to calculations with 10 OpenMP threads. The number of MPI processes is 6, the batching of the virtual orbitals was turned off.

References

- [1] A. Gellé, J. Varignon, and M.-B. Lepetit, EPL (Europhysics Letters) **88**, 37003 (2009). 1
- [2] J. Miralles, J. P. Daudey, and R. Caballol, Chem. Phys. Lett. **198**, 555 (1992). 1
- [3] V. M. G. *et al.*, Chem. Phys. Lett. **238**, 222 (1995).
- [4] V. M. García, M. Reguero, and R. Caballol, Theor. Chem. Acc. **98**, 50 (1997). 1
- [5] E. Rebolini and M.-B. Lepetit, The Journal of Chemical Physics **154**, 164116 (2021), URL <https://doi.org/10.1063/5.0045672>. 6

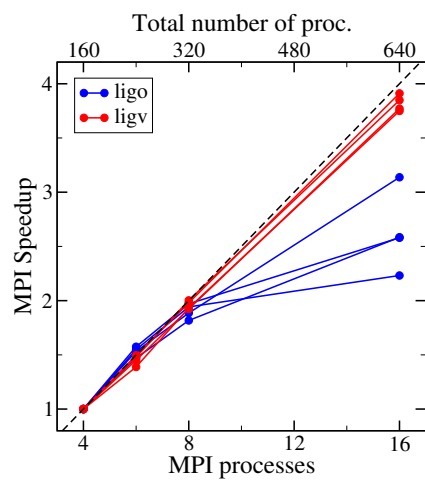


Figure 3: MPI speedup compared to a calculation with 4 MPI processes. The number of OpenMP threads is 40, as well as the size of the batches.