For an ab initio calculation of the magnetic excitations: RelaxSE!

Elisa Rebolini, Marie-Bernadette Lepetit

▶ **To cite this version:**

Elisa Rebolini, Marie-Bernadette Lepetit. For an ab initio calculation of the magnetic excitations: RelaxSE!. Journal of Chemical Physics, 2021, 154 (16), pp.164116. 10.1063/5.0045672 . hal-03812402

HAL Id: hal-03812402
https://hal.science/hal-03812402

Submitted on 11 Aug 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

The Journal of Chemical Physics ARTICLE scitation.org/journal/jcp

For an *ab initio* calculation of the magnetic excitations: RelaxSE!

Cite as: J. Chem. Phys. 154, 164116 (2021); doi: 10.1063/5.0045672
Submitted: 28 January 2021 • Accepted: 10 April 2021 •
Published Online: 29 April 2021

View Online Export Citation CrossMark

Elisa Rebolini[1,a] and Marie-Bernadette Lepetit[1,2,b]

AFFILIATIONS

[1]Institut Laue Langevin, Grenoble, France
[2]Institut Néel, CNRS UPR, 2940 Grenoble, France

Note: This paper is part of the JCP Special Collection in Honor of Women in Chemical Physics and Physical Chemistry.
[a]**Electronic mail:** rebolini@ill.fr
[b]**Author to whom correspondence should be addressed:** Marie-Bernadette.Lepetit@neel.cnrs.fr

ABSTRACT

In this paper, we present a novel efficient and parallel implementation, RelaxSE, for the calculation of the low-lying excited states and energies of strongly correlated systems. RelaxSE is based on the fully uncontracted multi-reference method of Selected Active Space + Single excitations. This method has been specifically designed to be able to tackle systems with numerous open shells per atoms. It is, however, computationally challenging due to the rapid scaling of the number of determinants and their non-trivial ordering induced by the selection process. We propose a combined *determinant-driven* and *integral-driven* approach designed for hybrid OpenMP/MPI parallelization. The performances of RelaxSE are evaluated on a controlled test set and show linear scaling with respect to the number of determinants and a small overhead due to the parallelization. Systems with up to 1×10^9 determinants are successfully computed.

Published under license by AIP Publishing. https://doi.org/10.1063/5.0045672

11 August 2023 11:07:55

I. INTRODUCTION

Strongly correlated systems present the peculiar characteristic that their low-energy physical and chemical properties are not driven by the kinetic energy of their Fermi-level electrons. In these systems, the electron–electron correlation, sub-dominant or even negligible in most systems, is much larger than the kinetic energy. As a consequence, degrees of freedom usually hidden by the kinetic aspects (charge, spin, orbital, etc.) can express themselves, resulting in a large variety of competing ground states and fascinating low-energy properties. It is, for instance, in strongly correlated compounds that phenomena such as high-temperature superconductivity,[1,2] colossal magneto-resistance,[3] large magneto-electric coupling,[4] and exotic magnetic orders[5,6] appear.

The consequences of strong correlation on the electronic structure are that

- the electrons responsible for the low-energy properties are localized, both spatially and energetically, and
- the wave functions of their ground and low-lying excited states are fundamentally multi-configurational.

It is, thus, not surprising that *ab initio* single-determinant based methods (such as density functional theory or single-reference post-Hartree-Fock methods) encounter difficulties in accurately describing the low energy properties of such systems. For this purpose, one relies on multi-reference configuration interaction (MRCI) methods. Unfortunately, MRCI methods can only be used on finite and relatively small systems. For infinite compounds, physicists, thus, use effective model Hamiltonians (such as Hubbard, Heisenberg, and related models), describing explicitly only the Fermi-level electrons. Due to the locality of the latter, the parameters of such models can be determined from spectroscopic calculations on embedded fragments.[7,8] One can then use *ab initio* MRCI calculations on the ground and low-lying excited states of such fragments and extract from them the pertinent degrees of freedom and the amplitude of the effective interactions of the effective models.[9]

To be able to be predictive with respect to experiments, effective models should be determined with a great accuracy (typically

Published under license by AIP Publishing

1 meV for effective magnetic interactions). Specific *ab initio* MRCI methods have been designed to reach this accuracy. For systems with a few unpaired electrons per active or magnetic atoms, the CAS + DDCI[10–12] (Complete Active Space + Difference Dedicated Configurations Interaction) or the LCAS + S[13] (Large CAS + single excitations) method proved their high reliability and efficiency. For instance, on copper[14–16] or vanadium oxides,[17] they allowed the determination of effective magnetic couplings within inelastic neutron scattering accuracy. The precision of these methods is even more obvious in molecular systems as the magnetic couplings can be experimentally evaluated with a much greater accuracy. For instance, the effective magnetic exchange was computed with an error smaller than 5 cm^{-1} in oxalato-bridged Cu(II) binuclear complexes.[18]

For systems with a large number of open shells per atom, these methods unfortunately reach a computational wall as their cost increases exponentially with the number of Fermi level orbitals to correlate (open-shell orbitals such as magnetic orbitals). For such systems, one has to rely on the SAS + S method developed in our group a few years ago.[19] The basic idea of this multi-reference method is to treat

- the strong correlation within the Fermi-level electrons,
- specific charge transfers if needed, and
- dynamical correlation (screening effect).

This method, described in detail in Sec. II, has shown its ability to predict magnetic couplings with similar accuracy as the standard CAS + DDCI method on nickel (Ni^{2+}: $3d^2$) and cobalt (Co^{2+}: $3d^7$) compounds[19] and within inelastic neutron scattering experimental accuracy on manganese (Mn^{3+}: $3d^4$) compounds.[20] While the SAS + S method has proven accurate, no efficient, parallel code existed until now. The purpose of the present paper is to describe our implementation of such a code and its efficiency.

The SAS + S method is detailed in Sec. II. The challenges to be met and the adopted solutions are described in Sec. III, while a few examples and their scalability are given in Sec. IV.

II. THEORETICAL BACKGROUND

Prior to describing the SAS + S method, let us shortly recall the principle of the more standard CAS + DDCI[10] and LCAS + S[13] methods, which constitute special cases of the SAS + S method.

These variational methods are based on the fully uncontracted expansion of the wave functions for the low-energy states of the system under consideration. They treat exactly the static correlation effects within the magnetic orbitals. Each configuration involved in this process is then used as a reference configuration for the inclusion of the dynamical correlation (screening effects).

One characteristic of these methods is that the reference space is a Complete Active Space[21] (CAS) based on the partitioning of the orbital space into three subsets, namely,

- the doubly occupied orbitals (doubly occupied in all the determinants of the CAS) and
- the active orbitals (that take in the CAS all possible occupations and spins, within a sector of fixed S_z and a number of electrons), and
- the virtual orbitals (empty in all the determinants of the CAS).

The CAS wave functions can, thus, be written as follows:

$$|\Psi_m^{\mathrm{CAS}}\rangle = \sum_I C_{I,m} |\Phi_I^{\mathrm{CAS}}\rangle$$

with

$$|\Phi_I^{\mathrm{CAS}}\rangle = \left| \prod_{\substack{o=1 \\ o\,\mathrm{dbl\ occ}}}^{nocc} o^{\sigma} o^{\bar{\sigma}} \prod_{\substack{a_I=1 \\ a_I\,\mathrm{act}}}^{nelact} a_I^{\sigma_{a_I}} \right\rangle, \tag{1}$$

where `nelact` is the number of active electrons and σ is the orbital spin ($S_z = \sum_i \sigma_i$). The CAS + DDCI or LCAS + S wave functions take the following form:

$$\begin{aligned} |\Psi_m\rangle &= \overbrace{|\Psi_m^{ref}\rangle}^{\mathrm{reference=CAS}} + \overbrace{|\Psi_m^{*}\rangle}^{\mathrm{screening}} \\ &= \sum_I C_{I,m} |\Phi_I^{\mathrm{CAS}}\rangle + \sum_{J^*} C_{J^*,m} |\Phi_{J^*}\rangle. \end{aligned} \tag{2}$$

In the LCAS + S method, all orbitals associated with the leading charge-transfer configurations, such as the ligand orbitals bridging the super-exchange terms in magnetic interactions, are included in the active space (see Ref. 13 and Fig. 1 for more details). The remaining dynamical correlation effects (screening effects on the CAS and charge transfer configurations) can, thus, be described by adding all determinants obtained by a single excitation on each determinant of this large CAS (see Fig. 1).

In the CAS + DDCI method,[10–12] on the contrary, the CAS is reduced to the main open-shell orbitals (such as magnetic orbitals). The leading charge-transfer configurations are, thus, treated as dynamical correlation terms. In order to get the screening effects on these configurations (in addition to the CAS ones), one has to include in the screening configurations not only the single but also the double excitations on each of the CAS determinants (two holes and two particles). The CAS + DDCI method differs from CAS + single and double excitations (CASSD) as it does not include the two occupied toward two virtual excitations (see Fig. 1 for more details). Indeed, it can be shown that, while these excitations largely contribute to the total energies, their contribution is essentially a global shift on the low-energy spectrum (provided that the sought states are defined by different CAS contributions as in magnetic excitations).

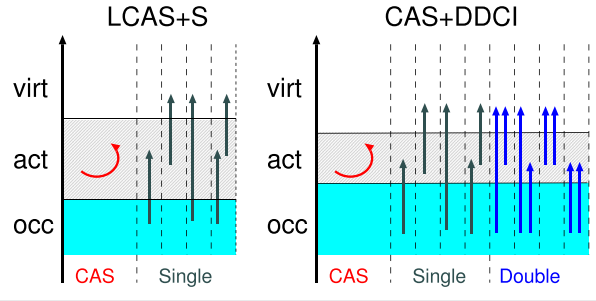

FIG. 1. Schematic representation of the LCAS + S and CAS + DDCI methods.

11 August 2023 11:07:55

Published under license by AIP Publishing

While it has been shown that these two methods provide equivalent low-energy spectra in correlated systems,[13] one sees from the previous discussion that the MRCI space to diagonalize could be further pruned without much loss of accuracy. Such a reduction becomes mandatory when the number of open shells per atom increases. This is the aim of the SAS + S method.[19] The price for such a reduction of the computational cost is an increase in the complexity. Indeed, we need now to define five classes of orbitals (seven including frozen occupied and deleted virtual orbitals that do not contribute in the calculation) and two levels of reference configurations. The first idea of the SAS + S method resides in the fact that, when the number of open shells per atom is large, most of the CAS configurations have a negligible weight in the sought states. The zeroth-order reference space for the SAS + S method is, thus, a selected sub-space of the minimal CAS. The second set of references contains the CAS configurations needed to correlate the previous set and the main charge-transfer configurations. The screening effects are then included by the addition of all single excitations on each determinant of the two reference spaces. The SAS + S wave function can, thus, be written as follows:

$$|\Psi_m\rangle = \overbrace{\underbrace{\sum_I C^0_{I,m}|\Phi^0_I\rangle}_{\substack{\text{zeroth-order}\\ \text{ref0}}} + \underbrace{\sum_J C^1_{J,m}|\Phi^1_J\rangle}_{\substack{\text{charge transfer + static corr}\\ \text{ref1}}}}^{\text{references}} + \overbrace{\sum_{J^*} C_{J^*,m}|\Phi_{J^*}\rangle}^{\text{screening}}$$
$$= |\Psi^0_m\rangle + |\Psi^1_m\rangle + |\Psi^*_m\rangle. \quad (3)$$

The orbital space is partitioned into

- the doubly occupied orbitals (named `occ`),
- the occupied ligand orbitals (named `ligo`),
- the active orbitals (named `act`),
- the virtual ligand orbitals (named `ligv`), and
- the virtual orbitals (named `virt`).

The zeroth-order configurations (`ref0`) have all `occ` and `ligo` orbitals doubly occupied, all `ligv` and `virt` orbitals empty, and various occupations and spins within the active orbitals. The charge transfer, correlation, and reference configurations (`ref1`) are single excitations

- within the CAS,
- from the `ligo` to the actives,
- from the `ligo` to the `ligv` space, and
- from the actives to the `ligv` space

(see Fig. 2 for more details). Finally, the screening effects are built from all possible single excitations on each of these references.

As can be seen from Fig. 2, the SAS + S space is a sub-space of both the CAS + DDCI and LCAS + S methods. Indeed, it includes up to double excitations with respect to the selected configuration of the minimal CAS (`ref0`); however, one of the two excitations does not involve the `occ` and `virt` orbitals that are usually the far most numerous. It results in a large reduction of the variational space to diagonalize without loss of accuracy (see Ref. 19 for more details).

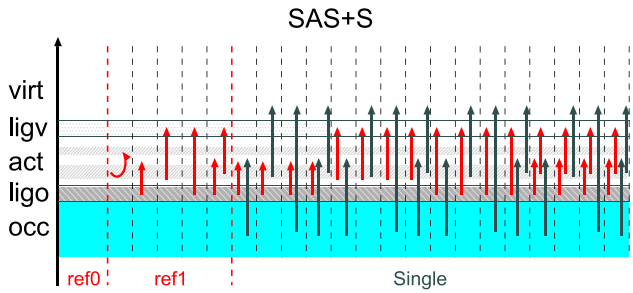

FIG. 2. Schematic representation of the SAS + S method. The blue screening-excitation lines can start either from the `occ` or `ligo` spaces and end either in the `ligv` or `virt` spaces.

Let us finally point out that the requirement for the variational space to be stable by the $\hat{S}^2$ operator (so that to insure that the sought states are eigenfunctions of both $\hat{S}_z$ and $\hat{S}^2$) is the inclusion of spin exchanges within the CAS. This requirement is responsible for some formally higher-level excitations in the SAS + S method, as it involves spin exchanges within the active space, on single and double excitations from the `occ`/`ligo` spaces toward `ligv`/`virt` ones.

III. CHALLENGES AND IMPLEMENTATION

All Configuration Interaction (CI) methods eventually reduce to the diagonalization of the Hamiltonian matrix on a given determinant space. When applying such methods on systems of increasing size, two computational bottlenecks are usually encountered:

- the number of determinants on which the matrix is expanded upon and
- the number of two-electron integrals used to compute each of the matrix elements.

For most flavors of CI methods, one of these challenges is usually dominant, and therefore, either a *determinant-driven* or an *integral-driven* implementation is used in order to circumvent it. In the SAS + S case, both challenges are faced simultaneously, with an additional complexity due to the selection of the determinants, as they no longer have a straightforward ordering.

In order to tackle this issue, we propose a hybrid MPI-OpenMP parallel implementation driven both by integral and determinant blocks. This approach 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. To distribute the calculation across several nodes (MPI parallel programming), the Hamiltonian matrix and the integrals are divided into blocks, and the contribution of each integral block to a given matrix block is computed independently. Moreover, for each pair integral/matrix block, a dedicated set of instructions is written within which OpenMP multi-threading is used. This allows us to fully exploit the sparsity of the Hamiltonian matrix and to minimize the number of central processing unit (CPU) operations, more specifically the number of tests. A simplified flowchart of the algorithm is depicted in Fig. 3.

11 August 2023 11:07:55

Published under license by AIP Publishing

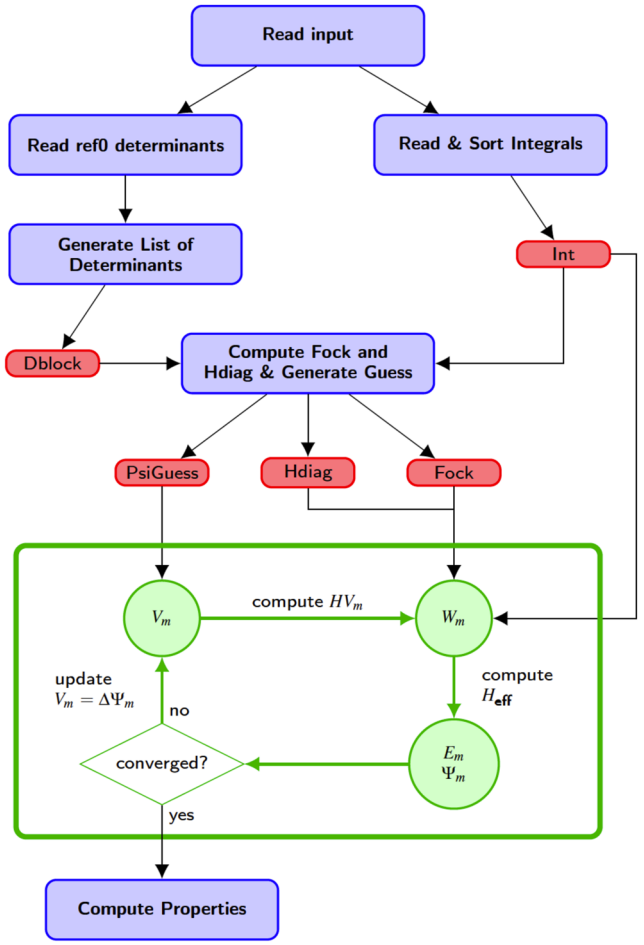

FIG. 3. Flowchart of the RELAXSE code.

Another important issue of the code is its built-in ability to accurately deal with (quasi-)degenerate eigenstates. Indeed, as we are interested in the low-energy excitations of strongly correlated systems and more specifically in computing low-energy magnetic excitations with great accuracy, the chosen diagonalization method should be able to handle the frequent (quasi-) degeneracy of these magnetic states. As a consequence, one cannot rely on simple Krylov space methods, such as the Lanczos[22] method (known to encounter difficulties with degenerate eigenstates) or the state-by-state Davidson method.[23] We, therefore, use the effective Hamiltonian variant of the Davidson method that treats on an equal footing all quasi-degenerate states[24,25] (see below).

A. Determinant generation and storage

A set of `ref0` configurations is given as input by specifying the occupancy of the `nact` active orbitals by the `nelact` active electrons. In order to ensure that the obtained determinant space is stable with respect to the spin operators $\hat{S}^2$ and $\hat{S}_z$, all possible spin permutations are applied on the input reference configurations. The obtained set of determinants is hereinafter denoted as the `ref0` determinants.

The complete list of determinants is then generated depending on the number of occupied (`nocc`), virtual (`nvirt`), ligand occupied (`nligo`), and ligand virtual (`nligv`) orbitals, as well as the method name (`method` = `CAS+S`, `CAS+SD`, `SAS+S`, `CAS+DDCI`, or `SAS+DDCI`). This list is not explicitly used during the Davidson procedure as only a few pointers and the ordering of the active parts are needed for its complete (and unique) description.

B. Block partitioning

The determinants are divided into nine blocks, denoted $D^{\text{Nexc}}_{\text{Nel}}$, where `Nel` is the number of additional electrons in the active part with respect to the reference configurations and `Nexc` is the number of additional excitations outside of the active space. This partitioning results in a block representation of the Hamiltonian matrix shown in Fig. 4.

For each block of the matrix, the number of orbital differences between the determinants and the nature of these differences (occupied, virtual, or active orbitals) determine the components needed to compute that block (Fock matrix elements, two-electron integrals, and pseudo-core energy).

The one- and two-electron integrals on the fragment orbitals are not recomputed. They are read from the `TraOne` and `TraInt` files generated by a MOLCAS/MOTRA calculation. The two-electron integrals $(ij|kl)$ are then sorted in 20 different categories depending on the virtual (virt + ligv) "v," active "a," or occupied (occ + ligo) "o" status of the orbitals involved ("aaaa," "aaao," "vaaa," "aaoo," "vaao," "vvaa," "vaoo," "vvao," "vvoo," "vava," "vvvo," "vvva," "vvvv," "oooo," "vovo," "aooo," "vavo," "vooo," "aoao," and

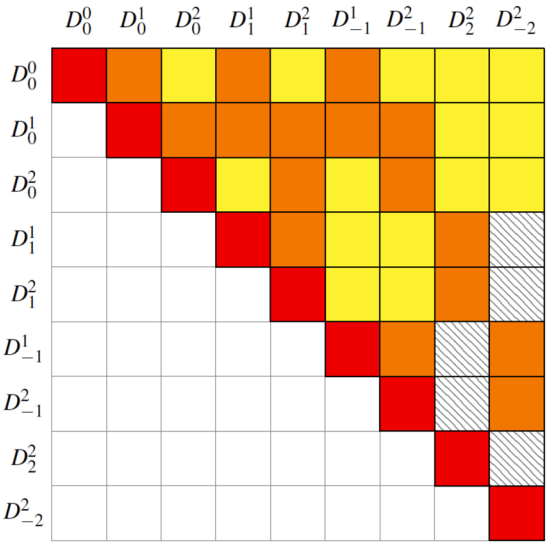

FIG. 4. Block representation of the upper half of the Hamiltonian matrix $\langle A|\hat{H}|B\rangle$. In the striped blocks, all determinants differ by at least three orbitals, and these blocks are, therefore, 0. In the yellow blocks, they show at least two differences, in the orange blocks at least 1, and in the red none.

11 August 2023 11:07:55

Published under license by AIP Publishing

"voao") and stored to disk in separate files. Only the integral block of interest is loaded in memory at a given time.

In order to speed up the calculations, the pseudo-Fock matrix (taking only into account the effect of the occupied and ligo orbitals) and the diagonal of the Hamiltonian matrix are computed and stored prior to the Davidson procedure.

C. Effective Hamiltonian Davidson's diagonalization

The effective-Hamiltonian Davidson diagonalization works on all M sought eigenstates simultaneously and builds over the iterations an effective Hamiltonian matrix (see Fig. 5).

At the first iteration, the effective Hamiltonian is defined by the Hamiltonian block over the guess vectors only,

$$H^1_{\text{eff}} = \hat{P}_1 \hat{H} \hat{P}_1,$$

where $\hat{P}_1 = \sum_{m=1}^{M} |V^1_m\rangle\langle V^1_m|$ is the projector on the $|V^1_m\rangle$ guess vectors. At iteration n, the effective Hamiltonian is built as

$$H^n_{\text{eff}} = \left(\hat{P}_1 + \sum_{i=2}^{n} \hat{P}_i\right) \hat{H} \left(\hat{P}_1 + \sum_{i=2}^{n} \hat{P}_i\right),$$

where $\hat{P}_i = \sum_{m=1}^{M} |V^i_m\rangle\langle V^i_m|$ is the projector on the ortho-normalized Davidson's correction vectors $|V^i_m\rangle$. The correction vectors $|V^i_m\rangle$ are defined as follows: The expansion of the raw correction vectors, $|\tilde{V}^n_m\rangle$, on a determinant basis set $\{|\Phi_k\rangle\}$ can be written as follows:

$$|\tilde{V}^n_m\rangle = \sum_i \tilde{C}^n_{k,m} |\Phi_k\rangle$$

with

$$\tilde{C}^n_{k,m} = \frac{\sum_h H_{k,h} C^{n-1}_{h,m} - H_{kk} C^{n-1}_{k,m}}{E^{n-1}_m - H_{kk}},$$

where E^n_m is the energy of the m-th eigenvector Ψ^n_m of $\hat{H}^n_{\text{eff}}$ and $C^i_{k,m}$ are the coefficients of the $|V^i_m\rangle$ expansion on the determinant basis set. The final correction vectors $|V^n_m\rangle$ are obtained from $|\tilde{V}^n_m\rangle$ by Schmidt's orthogonalization of $|\tilde{V}^n_m\rangle$ with respect to

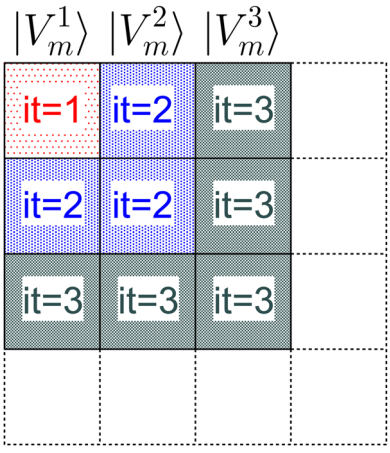

FIG. 5. Effective Hamiltonian matrix as built over Davidson's iterations.

ALGORITHM 1. Effective Hamiltonian Davidson's diagonalization.

Data: Guess vectors V^1_m
Result: E_m, Ψ_m
while *convergence not reached* **do**
  for $i <$ *maxit* **do**
    compute $\forall m$, $W^i_m = \hat{H} V^i_m$
    compute $\hat{H}^i_{\text{eff}} = [V^j_m \cdot W^k_n]$
    diagonalize $\hat{H}^i_{\text{eff}} \longrightarrow E^i_m, \Psi^i_m$
    if *conv. on energies* **then** exit
    compute raw corrections vectors
    if *conv. on vectors* **then** exit
    for $j < i$ **do**
      Schmidt-orthog. V^{i+1}_m with V^j_m
    Löwdin orthog. V^{i+1}_m
  $V^1_m = \Psi^{maxit}_m$

$\{|V^i_m\rangle, i = 1, n-1\}$, followed by Löwdin's orthonormalization between them.

Every *maxit* iteration, the effective Hamiltonian is contracted, i.e., the guess vectors are replaced by the eigenvectors of $\hat{H}^{maxit}_{\text{eff}}$, and the effective Hamiltonian is reset to the $H^1_{\text{eff}} = \hat{P}_1 \hat{H} \hat{P}_1$ block (see Algorithm 1).

D. Computing HV

At each iteration i of the Davidson procedure, the time-consuming step is the evaluation of $|W^i_m\rangle = H|V^i_m\rangle$. For simplicity, the iteration and state indices (i, m) are dropped in what follows. The computation of $W = H \cdot V$ is done block-wise at the integral and the determinant level. For each integral kind *intkind* in the list given in Sec. III B and for the Fock integrals, we determine the pairs of determinant blocks (D_A, D_B) for which these integrals have a non-zero contribution, $H^{intkind}_{AB}$, to the Hamiltonian block $H_{AB} = \langle D_A|H|D_B\rangle$. The tuples $(intkind, D_A, D_B)$ constitute the items on which the MPI parallelization is performed as shown in Algorithm 2. When the determinant blocks are very large, it is possible to divide the calculation in smaller parts by batching the determinants. However, the gain is not systematic as the overhead due to the reading of the integrals from disk can be large.

ALGORITHM 2. External loop for the MPI parallelization.

Data: Vectors V_m
Result: Vectors W_m
mpi_init()
mpi_comm_rank(mpi_comm_world, id_cpu)
mpi_comm_size(mpi_comm_world, nb_cpu)
nblock = nb_item/(nb_cpu −1)
for $i = 1$, *nblock* **do**
  indx = (i − 1) ∗ nb_cpu + id_cpu +1
  (intkind, A, B) = MPI_item_list(indx)
  load intkind from file
  compute_HVm(Vm, Wm, intkind, A, B)
mpi_allreduce(Wm, mpi_sum, mpi_comm_world)

11 August 2023 11:07:55

Published under license by AIP Publishing

ALGORITHM 3. External loops for the OpenMP parallelization.

```
Data: Blocks A and B of V_m, intkind
Result: Blocks A and B of W_m
!$OMP PARALLEL
!$OMP DO reduction(+:WB) reduction(+:WA)
for particles of A do
    for holes of A do
        for active of A do
            for particles of B do
                for holes of B do
                    for active of B do
                        compute diff. in the active parts
                        compute matrix element elm
                        W_m^A += elm* V_m^B
                        W_m^B += elm* V_m^A
!$OMP END DO
!$OMP END PARALLEL
```

For each MPI item ($intkind$, D_A, D_B), specific instructions were generated in order to minimize the number of CPU operations. The code generator uses all the information on the determinant blocks (number of holes and particles), the ordering and spin of the particles and holes, and the number of differences between the active parts of the determinants, to produce an optimal code, with no or very little branching due to if/then/else statements. Moreover, the position of each matrix element is computed on-the-fly, and the explicit determinant ordering is not needed. At this level, OpenMP multi-threading is used on the most external loop, chosen to be on the particle indices (or on the holes when there are no particles). The hermiticity of the Hamiltonian is fully exploited for both the diagonal blocks (only the upper-half is computed) and the off-diagonal blocks (simultaneous computation of both blocks choosing the most favorable external loop in terms of OpenMP performances). The OpenMP loops are shown in Algorithm 3.

When determinant batching is turned on, the outer loop runs only on a batch of particles instead of the complete set so that the workload can be shared on several nodes at the MPI level.

IV. PERFORMANCES AND EXAMPLES

The calculations presented in this section were performed on the CPU partition of the Jean-Zay supercomputer of the CNRS national computer center: IDRIS. Jean-Zay CPU partition is a HPE SGI 8600 computer of 1528 nodes with 2 Intel Cascade Lake 6248 processors (20 cores at 2.5 GHz), that is, 40 cores per node, and 192 GB of memory per node. It uses an Omni-PAth interconnection network of 100 Gb/s: 1 link per scalar node.

The code was compiled with gcc/9.1.0 and openmpi/3.1.5 at the -O2 optimization level, enabling OpenMP and MPI parallelizations.

Most of the test calculations were performed on a middle size system in order to be able to run a large number of calculations with different characteristics. We chose the calculation of the magnetic integrals in the $YMnO_3$ compound used in Ref. 20 as a testing system. We, thus, used a Mn_2O_9 quantum cluster, embedded in a bath of total ion pseudopotentials[26] and renormalized[27] point charges that reproduce the Madelung potential.

The total number of orbitals is 199 with 8 active orbitals. The calculations were done in the $S_z = 3$ sector computing the two eigenvectors required for the calculation of the magnetic interaction.

In order to evaluate the performances of the code with respect to the total number of determinants, calculations were performed for different numbers of ligand occupied and ligand virtual orbitals (LIGO and LIGV test sets, respectively), keeping the total number of occupied and virtual orbitals constant, as shown in Table I.

The last test BIG was designed to reach a total number of determinants greater than 1×10^9. Due to the 192 GB memory limitation of the nodes of the Jean-Zay supercomputer, this calculation was performed on a local node boosted with 2TB of memory. The node is a Lenovo ThinkSystem SR645 with AMD EPYC 7402 3.0 GHz processors. As we had a unique node, the calculation was done using OpenMP parallelization only and 40 OpenMP-threads.

On this architecture, the code was compiled with gcc/8.3.1 and openmpi/4.0.2 at the -O2 optimization level, enabling only OpenMP parallelizations.

A. Code stability

Prior to detailing the efficiency of the code with respect to the different parallelization schemes, let us say a work on the stability of the code with respect to the number of OpenMP threads and MPI precesses. At each iteration, the energies of the two states that are computed do not exhibit any difference on the 15 significative numbers expected in a Fortran double precision code, whether as a function of the number of OpenMP threads or MPI processes.

B. Scaling with respect to the number of determinants

The total CPU time per iteration as a function of the number of determinants is shown in Fig. 6. For each, except the last orbital configuration of Table I, a total of 16 calculations were performed with different parallelization options: with 4 or 6 MPI processes, with 10, 20, 30, or 40 OpenMP threads, without batching, and with batches of 40 particles. One can immediately see that the code scales linearly with the number of determinants. The typical CPU time per determinant is 10^{-3} s.

TABLE I. Orbital partitioning in the $YMnO_3$ 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
BIG	47	4	8	6	134	1 097 706 172

11 August 2023 11:07:55

Published under license by AIP Publishing

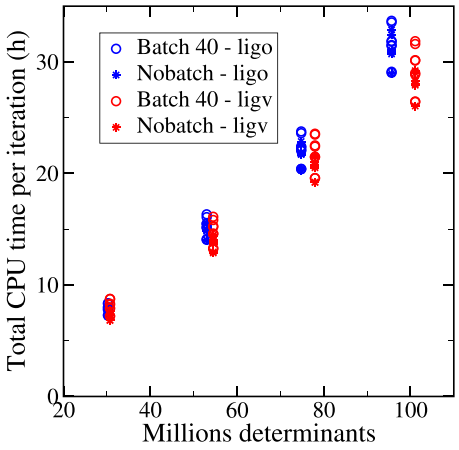

FIG. 6. Total CPU time as a function of the number of determinants for either occupied (ligo) or virtual (ligv) ligand orbitals. The number of ligo (ligv) orbitals is 2, 4, 6, and 8. Calculations were performed with 4 or 6 MPI processes, with 10–40 OpenMP threads, and with (circles) and without (stars) batching of the determinants.

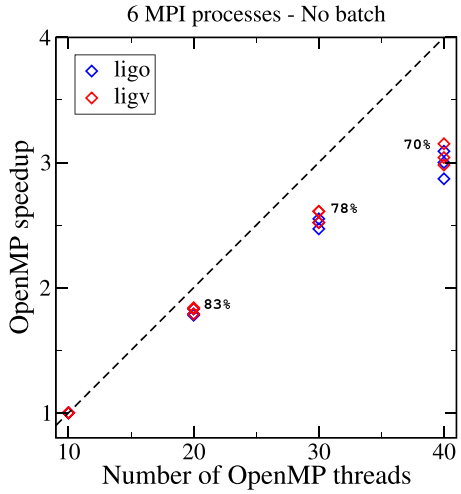

FIG. 7. OpenMP speedup compared to calculations with ten OpenMP threads. The number of MPI processes is 6, and the batching of the virtual orbitals was turned off. The number of ligo (ligv) orbitals is 2, 4, 6, and 8.

Depending on the orbital partitioning (ligo vs ligv) and the computational conditions (batching), the slope is slightly different, as expected due to the parallelization overhead. Indeed, as the OpenMP parallelization is done preferably on the particle loops, at a constant number of determinants, it is more effective to have numerous but shorter tasks (long particle loops and short hole loops). This is the reason why the "ligv" configurations show a smaller CPU time per determinant than the "ligo" configurations. For the same reason, the dynamic distribution of the OpenMP tasks is more effective when the full particle loop is treated in one go, rather than in batches.

The CPU time is independent of the number of MPI processes, which means that the MPI reduction has a negligible overhead. Concerning the dependence with respect to the number of OpenMP threads, an increase of only 6% in the CPU time is observed, independently of the calculation, when going from 20 to 40 threads.

C. OpenMP speedup

As shown in the previous paragraph, the total CPU time is mostly constant with respect to the number of OpenMP threads. However, the performance of the OpenMP implementation needs to be evaluated on the wall time. In Fig. 7, the wall-time speedup as a function of the number of OpenMP threads is shown for the different orbital configurations of Table I and 6 MPI processes. Calculations with ten threads were taken as references. Perfect scaling is drawn as a dashed line.

As the number of OpenMP threads increases, one can see that the speedup is going away from optimal scaling (83% for 20 threads to 70% for 40 threads) and is slightly better for the "ligv" orbital configurations than the "ligo" ones, as discussed in Sec. IV B. The sub-optimal behavior of the OpenMP speedup is due to the fact that (i) within a given ($intkind$, D_A, D_B) block, the external loop on which the parallelization is done is too short with respect to the number of threads, and (ii) for some blocks [the diagonal cases ($intkind$, D_A, D_A) and two-particle blocks], the OpenMP tasks have different sizes, which impairs the load balancing.

D. MPI speedup

The MPI speedup on the wall time is shown in Fig. 8 as a function of the number of MPI processes. Calculations are performed for the different orbital configurations of Table I with 40 OpenMP threads, enabling the determinant batching (batches of 40 on the

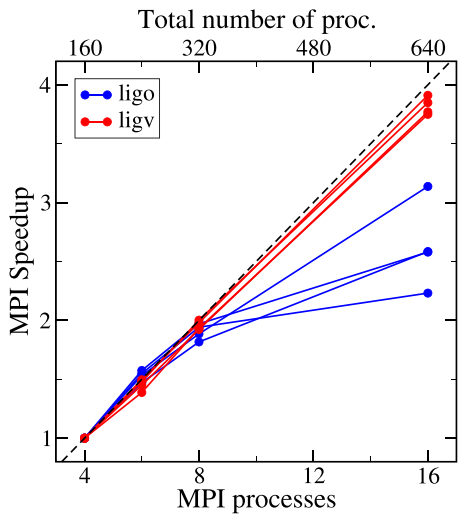

FIG. 8. MPI speedup compared to a calculation with four MPI processes. The number of OpenMP threads, as well as the size of the batches, is 40. The number of ligo (ligv) orbitals is 2, 4, 6, and 8.

11 August 2023 11:07:55

Published under license by AIP Publishing

particle loops). Calculations with four MPI processes were taken as references. Optimal speedup is drawn as a dashed line.

A very different behavior is observed for the "ligo" and "ligv" calculations. The "ligv" calculations exhibit an almost perfect scaling up to 16 MPI processes (16 × 40 = 640 threads), while the gain becomes quite limited after eight processes in the case of the "ligo" calculations.

The MPI load balancing is done after the first full iteration by distributing the (*intkind*, D_A, D_B) blocks on the different MPI processes according to their wall time in the previous iteration. As the number of determinants per block and the number of integrals can be very different according to the system under consideration, the absolute (and relative) timings of the blocks can vary by several orders of magnitude, effectively bounding the wall time to one of the largest blocks. In order to circumvent this issue, determinant batching was introduced to divide the largest block calculations into several smaller ones. The cost is the reading of the integrals from files for each batch. As the batching is currently only coded on the particles, when the blocks are getting larger due to the "ligo," there is no option to divide the work, and the gain of adding more CPUs becomes negligible. This is the reason for the poor performances of these calculations with more than eight MPI processes. This issue should be fixed in the next version of the code.

E. 10^9 dets

When the number of open shells per atom increases—for instance, if one aims at evaluating the magnetic interaction between two rare-earth atoms—or when the basis set increases—as in systems with non-trivial ligands—the number of determinants increases very fast. One of our goals for the RelaxSE code was to be able to tackle such extreme cases and be able to diagonalize matrices up to 10^9 determinants. In the BIG test case, we, thus, increased the number of ligo and ligv orbitals in order to push the RelaxSE code to its limits. Four orbitals in the ligo spaces and six in the ligv orbital spaces lead to 1 097 706 172 determinants with the SAS + S method.

The wall time per iteration was 10 h 45 min. As the effective Hamiltonian Davidson procedure requires between 20 and 25 iterations to converge, a complete calculation takes 10 days, making such calculations accessible even if expensive. The total CPU time per determinant is comparable with the calculations performed on Jean-Zay: 0.96×10^{-3} s and the OpenMP speedup: $t_{\text{total CPUtime}}/t_{\text{Walltime}} \simeq 27.3$. This OpenMP speedup is comparable with the one observed in the smaller calculations (see Fig. 7), confirming the total independence of the OpenMP speedup with the size of the calculation.

This large calculation confirms the complete linear scaling of our code as a function of the number of determinants.

V. CONCLUSION

The RelaxSE code is an efficient OpenMP + MPI code dedicated to compute low energy excitations with a high accuracy. It is a totally decontracted MRCI code specifically designed for magnetic excitations in strongly correlated systems, even if standard SDCI calculations can also be performed. Indeed, RelaxSE is able to perform MR-SCI, MR-SDCI, CAS + DDCI,[10] and SAS + S[19] calculations. The specificity of the SAS + S method is to be able to compute the low energy spectrum of systems containing atoms with numerous open shells, with an accuracy better than meV. With the ability to reach diagonalization up to 10^9 determinants, RelaxSE opens new possibilities for the study of magnetic or multiferroic systems and even heavy fermion systems with an accuracy comparable with inelastic neutron scattering measurements. Similarly, it opens the possibility for relatively easy *ab initio* calculation of the magneto-electric coupling tensor as the accurate determination of the magnetic excitations as a function of an applied electric field is the main bottleneck (see Ref. 28).

Concerning the current implementation, some further developments are being considered. For instance, an efficient batching scheme needs to be implemented for "ligo" systems in order to fully exploit the MPI parallelization. Another issue is the memory requirements for the very big systems. As mentioned above, we had to switch to a high-memory node in order to run the calculations on the BIG system. Such nodes are not common in the High Performance Computing centers, and a more memory-friendly implementation needs to be designed for these cases. One possibility would be to write the Davidson vectors to disk and only load to memory the required blocks or sub-blocks (using determinant batching).

Finally, we plan to explore additional methodological developments, such as the perturbative treatment of the double excitations within effective Hamiltonian theory.

The RelaxSE code will be distributed under LGPL license and shortly be accessible at the url https://code.ill.fr/relaxse/relaxse-code.

11 August 2023 11:07:55

ACKNOWLEDGMENTS

The authors thank the IDRIS and CINES computer centers as the calculations presented in this work were performed on the IDRIS under Project No. 91842. The 2TB node used for the BIG calculation was bought through the ANR project HTHPCM.

We specifically would like to thank Patrick Corde from IDRIS and Gabriel Hautreux from CINES for their help in the debugging and optimization of the OpenMP parallelization.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request. A public release of the RelaxSE code is planned at the end of 2021.

REFERENCES

[1] J. G. Bednorz and K. A. Müller, Z. Phys. B: Condens. Matter **64**, 189 (1986).
[2] D. Johrendt and R. Pöttgen, Angew. Chem., Int. Ed. **47**(26), 4782 (2008).
[3] D. Saurel, C. Simon, A. Pautrat, C. Martin, C. Dewhurst, and A. Brûlet, Phys. Rev. B **82**, 054427 (2010).
[4] N. Hur, S. Park, P. A. Sharma, J. S. Ahn, S. Guha, and S.-W. Cheong, Nature **429**, 392–395 (2004).
[5] I. Affleck, J. Phys.: Condens. Matter **1**, 3047 (1989).
[6] L. Balents, Nature **464**, 199 (2010).
[7] F. Illas, I. P. R. de Moreira, C. de Graaf, O. Castell, and J. Casanovas, Phys. Rev. B **56**, 5069–5072 (1997).

Published under license by AIP Publishing

[8]I. P. R. de Moreira, F. Illas, C. J. Calzado, J. F. Sanz, J.-P. Malrieu, N. B. Amor, and D. Maynau, Phys. Rev. B **59**, R6593–R6596 (1999).
[9]M.-B. Lepetit and A. Gellé, Comput. Theor. Chem. **1116**, 59 (2017).
[10]J. Miralles, J.-P. Daudey, and R. Caballol, Chem. Phys. Lett. **198**, 555 (1992).
[11]V. M. García *et al.*, Chem. Phys. Lett. **238**, 222 (1995).
[12]V. M. García, M. Reguero, and R. Caballol, Theor. Chem. Acc. **98**, 50 (1997).
[13]A. Gellé, M. L. Munzarová, M.-B. Lepetit, and F. Illas, Phys. Rev. B **68**, 125103 (2003).
[14]C. J. Calzado and J.-P. Malrieu, Eur. Phys. J. B **21**, 375 (2001).
[15]C. Graaf, I. P. R. de Moreira, F. Illas, O. Iglesias, and A. Labarta, Phys. Rev. B **66**, 014448 (2002).
[16]A. Gellé and M.-B. Lepetit, Phys. Rev. B **74**, 235115 (2006).
[17]N. Suaud and M.-B. Lepetit, Phys. Rev. B **62**, 402–409 (2000).
[18]J. Cabrero, N. Ben Amor, C. de Graaf, F. Illas, and R. Caballol, J. Phys. Chem. A **104**(44), 9983 (2000).
[19]A. Gellé, J. Varignon, and M.-B. Lepetit, Europhys. Lett. **88**, 37003 (2009).
[20]J. Varignon, S. Petit, A. Gellé, and M.-B. Lepetit, J. Phys.: Condens. Matter **25**, 496004 (2013).
[21]B. O. Roos, P. R. Taylor, and P. E. M. Sigbahn, Chem. Phys. **48**(2), 157–173 (1980).
[22]B. N. Parlett, *The Symmetric Eigenvalue Problem* (Prentice-Hall, 1980).
[23]E. R. Davidson, J. Comput. Phys. **17**, 87 (1975).
[24]M. Pélissier, private communication (1988).
[25]B. Liu, "The simultaneous expansion method for the iterative solution of several of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices," in *Numerical Algorithms in Chemistry: Algebraic Methods* (Lawrence Berkeley Laboratory, University of California, 1978), pp. 49–53.
[26]N. W. Winter, R. M. Pitzer, and D. K. Temple, J. Chem. Phys. **86**, 3549 (1987).
[27]A. Gellé and M.-B. Lepetit, J. Chem. Phys. **128**, 244716 (2008).
[28]M. B. Lepetit, Theor. Chem. Acc. **135**, 91 (2016).

11 August 2023 11:07:55

Published under license by AIP Publishing