Spin-adapted selected configuration interaction in a determinant basis

Vijay Gopal Chilkuri, Thomas Applencourt, Kevin Gasperich, Pierre-Francois Loos, Anthony Scemama

Spin-adapted selected configuration interaction in a determinant basis

Vijay Gopal Chilkuri,[1] Thomas Applencourt,[2] Kevin Gasperich,[3] Pierre-François Loos,[1] and Anthony Scemama[1, a)]

[1)]*Laboratoire de Chimie et Physique Quantiques, Université de Toulouse, CNRS, UPS, France*

[2)]*Argonne Leadership Computing Facility, Argonne National Laboratory, Argonne, Illinois 60439 USA*

[3)]*Computational Science Division, Argonne National Laboratory, Argonne, Illinois 60439 USA*

Selected configuration interaction (SCI) methods, when complemented with a second-order perturbative correction, provide near full configuration interaction (FCI) quality energies with only a small fraction of the Slater determinants of the FCI space. However, a selection criterion based on determinants alone does not ensure a spin-pure wave function. In other words, such SCI wave functions are not eigenfunctions of the $\hat{S}^2$ operator. In many situations (bond breaking, magnetic system, excited state, etc), having a spin-adapted wave function is essential for a quantitatively correct description of the system. Here, we propose an efficient algorithm which, given an arbitrary determinant space, generates all the missing Slater determinants allowing one to obtain spin-adapted wave functions while avoiding manipulations involving configuration state functions. For example, generating all the possible determinants with 6 spin-up and 6 spin-down electrons in 12 open shells takes 21 CPU cycles per generated Slater determinant. The selection is still done with individual determinants, and one can take advantage of the basis of configuration state functions in the diagonalization of the Hamiltonian to reduce significantly the memory footprint.

Keywords: Selected Configuration Interaction ; spin-adaptation ; configuration state functions

I. INTRODUCTION

In recent years, selected configuration interaction (SCI) methods[1–3] have become more and more popular,[4–31] especially for the accurate calculation of electronic excitation energies.[32–47] *Determinant-based SCI* refers to configuration interaction in a truncated space of determinants. For instance, a SCI with singles and doubles (SCISD) refers to the diagonalization of the CISD Hamiltonian with only a subset of chosen (or selected) determinants belonging to the CISD space. There exists many variants of SCI methods differing in two major aspects. The first one is the nature of the target space: the most common spaces are the multi-reference CI (MRCI) space,[5,10,48] the (frozen-core) full CI (FCI) space[4,8,9,20,34] and the complete active space (CAS).[49] The second aspect in which SCI methods differ are in the rules used to select the determinants, thus affecting the convergence with respect to the number of determinants and the computational cost. The various permutations of such rules results in a plethora of SCI methods.

Discussing the different kinds of selection rules is beyond the scope of the present article. The reader who is not acquainted with SCI methods only needs to be aware of a few key aspects: (i) the selection criterion is chosen to include the most energetically relevant determinants in the variational space; (ii) SCI methods produce wave functions that are potentially expanded in an *arbitrary* set of determinants; (iii) it is of common practice to compute the Epstein-Nesbet second-order perturbative correction (E_{PT2}) to the variational energy, in order to estimate the lowest eigenvalues of the CI Hamiltonian matrix defined by the method (CAS, MRCI, FCI, ...); (iv) as the number of determinants grows, $E_{\text{PT2}} \to 0$ and the variational SCI energy converges monotonically to the exact energy of the CI Hamiltonian.

A balanced description of excited states, magnetic systems, and bond breakings require the wave functions to be spin-adapted, i.e., eigenfunctions of the $\hat{S}^2$ operator. The Slater determinant many-particle representation is, by construction, only strictly an eigenfunction of the $\hat{S}_z$ operator and therefore does not ensure a spin-pure wave function. The usual way to enforce the wave function to be an eigenfunction of $\hat{S}^2$ is to work in a basis where each element of the basis is an eigenfunction of $\hat{S}^2$ with the desired eigenvalue. These basis functions are built as linear combinations of Slater determinants, and are known as *configuration state functions* (CSF).

A natural option would be to express SCI in terms of CSFs. However, due to the complexity in the calculation of matrix elements in the CSF basis, many SCI implementations still rely on determinants. Opting for the CSF representation would require a major effort for re-writing the software, such as the important work that was done in the NECI FCIQMC code which now uses the graphical unitary group approach[50,51] and the ORCA program which uses the angular-momentum coupling based approach[52,53]. In the present paper we follow a different route and present simple recipes to ensure that the selected wave functions are spin-adapted without requiring too many modifications in a determinant-based code.

II. MANY-PARTICLE BASIS REPRESENTATIONS

A *configuration* is a vector of molecular orbital occupation numbers. For example, the configuration $(2,1,1,1,1)$ can be written as a linear combination of six determinants

$$\begin{pmatrix} \uparrow\downarrow \\ \uparrow \\ \uparrow \\ \uparrow \\ \uparrow \end{pmatrix} = a \begin{pmatrix} \uparrow\downarrow \\ \uparrow\downarrow \\ \uparrow \\ \uparrow \\ \uparrow\downarrow \end{pmatrix} + b \begin{pmatrix} \uparrow\downarrow \\ \uparrow\downarrow \\ \uparrow \\ \uparrow \\ \uparrow\downarrow \end{pmatrix} + c \begin{pmatrix} \uparrow\downarrow \\ \uparrow\downarrow \\ \uparrow \\ \uparrow \\ \uparrow\downarrow \end{pmatrix} + d \begin{pmatrix} \uparrow\downarrow \\ \uparrow\downarrow \\ \uparrow \\ \uparrow \\ \uparrow\downarrow \end{pmatrix} + e \begin{pmatrix} \uparrow\downarrow \\ \uparrow\downarrow \\ \uparrow \\ \uparrow \\ \uparrow\downarrow \end{pmatrix} + f \begin{pmatrix} \uparrow\downarrow \\ \uparrow\downarrow \\ \uparrow \\ \uparrow \\ \uparrow\downarrow \end{pmatrix} \tag{1}$$

[a)]Electronic mail: scemama@irsamc.ups-tlse.fr

or of two CSFs with coefficients A and B

$$\begin{pmatrix} \dagger \\ \dagger \\ \dagger \\ \dagger \\ \ddagger \end{pmatrix} = A \times \frac{1}{2}\left[\begin{pmatrix} \uparrow- \\ -\downarrow \\ -\downarrow \\ \uparrow- \\ \uparrow\downarrow \end{pmatrix} + \begin{pmatrix} -\downarrow \\ \uparrow- \\ \uparrow- \\ -\downarrow \\ \uparrow\downarrow \end{pmatrix} - \begin{pmatrix} \uparrow- \\ -\downarrow \\ \uparrow- \\ -\downarrow \\ \uparrow\downarrow \end{pmatrix} - \begin{pmatrix} -\downarrow \\ \uparrow- \\ -\downarrow \\ \uparrow- \\ \uparrow\downarrow \end{pmatrix}\right] + B \times \frac{\sqrt{3}}{6}\left[-2\begin{pmatrix} -\downarrow \\ -\downarrow \\ \uparrow- \\ \uparrow- \\ \uparrow\downarrow \end{pmatrix} + \begin{pmatrix} \uparrow- \\ -\downarrow \\ -\downarrow \\ \uparrow- \\ \uparrow\downarrow \end{pmatrix} + \begin{pmatrix} \uparrow- \\ -\downarrow \\ \uparrow- \\ -\downarrow \\ \uparrow\downarrow \end{pmatrix} - 2\begin{pmatrix} \uparrow- \\ \uparrow- \\ -\downarrow \\ -\downarrow \\ \uparrow\downarrow \end{pmatrix} + \begin{pmatrix} -\downarrow \\ \uparrow- \\ \uparrow- \\ -\downarrow \\ \uparrow\downarrow \end{pmatrix} + \begin{pmatrix} -\downarrow \\ \uparrow- \\ -\downarrow \\ \uparrow- \\ \uparrow\downarrow \end{pmatrix}\right] \tag{2}$$

By definition, all the determinants belonging to the same CSF are associated with the same configuration, and the determinants associated with a given configuration may be involved in multiple CSFs. Expressing Eq. (1) in terms of CSFs is an overdetermined problem: six parameters (a, b, c, d, e, f) for determinants *vs* two parameters (A, B) for the CSFs, so it has no unique solution in the general case. Only eigenfunctions of the $\hat{S}^2$ operator possess the necessary constraints to enable the exact transformation.

A few years ago, Bytautas and Ruedenberg proposed a simple scheme to truncate large determinant-based wave functions while maintaining the spin purity.[54] The squared coefficients of the determinants within the same configuration are summed together to produce the so-called *space-product weights*, which are then used to truncate the wave function. As the truncation occurs by removing configurations, one can understand from Eqs. (1) and (2) that the removal of all the determinants associated with a configuration is equivalent to a removal all the CSFs associated with the same configuration, hence keeping the spin purity of the wave function.

Following this idea, imposing spin adaptation in determinant-based SCI methods can be done by (i) identifying all the configurations of the determinants composing the variational space, (ii) generating all the determinants of the required multiplicity corresponding to these configurations, and (iii) diagonalizing the Hamiltonian in this spin-complete determinant space. Since the Hamiltonian commutes with the $\hat{S}^2$ operator, the obtained eigenfunction is automatically spin-adapted. An efficient algorithm to carry out this procedure is presented in this paper. Because the obtained wave functions are spin-adapted, they can be exactly expressed in terms of CSFs.[55–57] Then, we take advantage of the reduction of the number of parameters to reduce the memory requirement of the Davidson diagonalization, which is the main bottleneck in today's SCI algorithms. All the presented algorithms are implemented in the open-source *Quantum Package* software.[58]

III. ALGORITHM

The wave function of a given electronic state is expressed as

$$|\Psi\rangle = \sum_I c_I |D_I\rangle \tag{3}$$

where each Slater determinant D_I is represented as a Waller-Hartree double determinant,[59]

$$D_I = d_i^{\uparrow}\, d_j^{\downarrow} \tag{4}$$

i.e., the product of a determinant of spin-up ($\uparrow$) orbitals $d_i^{\uparrow}$ and a determinant of spin-down ($\downarrow$) orbitals $d_j^{\downarrow}$. Such a representation can be encoded as a pair of bit strings $(\mathtt{d}_i, \mathtt{d}_j)$, where each bit string is of length N_{orb}, the number of molecular orbitals. The spin-up and spin-down orbitals originate from a restricted Hartree-Fock or a CAS self-consistent field (CASSCF) calculation, so that the spatial part of these orbitals are common for both spin manifolds. Within a bit string, each bit corresponds to a spin-orbital; the bit is set to `1` if the orbital is occupied, and it is set to `0` if the orbital is empty. In low-level languages such as Fortran or C, a bit string may be stored as an array of N_{int} 64-bit integers, where

$$N_{\text{int}} = \left\lfloor \frac{N_{\text{orb}} - 1}{64} \right\rfloor + 1 \tag{5}$$

This representation allows for efficient determinant comparisons using bit-wise operation capabilities of modern processors[60] and will be convenient in the following.

All the CPU cycle measurements were performed on an Intel(R) Xeon(R) Gold 6140 CPU@2.30GHz with the GNU Fortran compiler 7.3.0, by reading the time stamp counter of the CPU with the `rdtsc` instruction.

A. Identification of the configurations

The configuration $\mathbf{p}_I$ associated with determinant D_I in Eq. (4), is a vector of integers defined as

$$[\mathbf{p}_I]_k = \begin{cases} 0 & \text{when the } k\text{-th orbital is unoccupied} \\ 1 & \text{when the } k\text{-th orbital is singly occupied} \\ 2 & \text{when the } k\text{-th orbital is doubly occupied} \end{cases} \tag{6}$$

If $\mathbf{p}_I$ is encoded as a pair of bit strings $(\mathtt{p}_I^{(1)}, \mathtt{p}_I^{(2)})$, where $\mathtt{p}_I^{(1)}$ and $\mathtt{p}_I^{(2)}$ encode respectively the singly and doubly occupied orbitals, the configuration can be computed as

$$\begin{cases} \mathtt{p}_I^{(1)} & = \mathtt{d}_i \oplus \mathtt{d}_j \\ \mathtt{p}_I^{(2)} & = \mathtt{d}_i \wedge \mathtt{d}_j \end{cases} \tag{7}$$

where $\oplus$ and $\wedge$ denote respectively the `xor` and the `and` binary operators.

```
function COMPUTE_PERMUTATIONS(n, m)
    /* n: input, number of bits set to 1 */
    /* m: input, number of bits set to 0 */
    /* v: output, an array of permutations */
    /* u, t, t', t'' and v are encoded in at least n + m + 1 bits */
    k ← 0
    u ← (1 ≪ n) − 1
    while u < (1 ≪ (n + m)) do
        v[k] ← u
        k ← k + 1
        t ← u ∨ (u − 1)
        t' ← t + 1
        t'' ← ((¬t ∧ t') − 1) ≫ (ctz(u) + 1)
        u ← t' ∨ t''
    end while
    return v
end function
```

FIG. 1. Anderson's algorithm. All the configurations of n bits set to 1 are generated in an integer of n + m bits in lexicographic order. ctz(i) counts the number of trailing zeros, i ≪ n shifts i by n bits to the left, i ≫ n shifts i by n bits to the right, ∧ is the bit-wise and operator, and ∨ is the bit-wise or operator.

Transforming all the selected determinants into a list of unique configurations can be done in linear time if a hash value is associated with each configuration.[61] Hence, the time for this transformation is negligible.

B. Generating all the determinants associated with a configuration

Given a configuration, one must generate all the possible determinants by considering either a spin-up or a spin-down electron in the singly occupied molecular orbitals, keeping the numbers of spin-up and spin-down electrons fixed. One can notice that, by doing so, all the generated determinants only differ by these singly occupied orbitals, so from now on we can consider a more compact representation: a bit string of $n_\uparrow + n_\downarrow$ bits, where $n_\uparrow$ and $n_\downarrow$ denote the numbers of spin-up and spin-down unpaired electrons. The bit is set to 1 when the orbital is occupied by a spin-up electron, and 0 when it is occupied by a spin-down electron. The indices of the singly occupied orbitals are kept in a look-up table **m** for later use.

To generate all the determinants keeping the numbers of spin-up and spin-down electrons constant, we need to build all the possible bit strings with $n_\uparrow$ bits set to 1 and $n_\downarrow$ bits set to 0. This compact representation allows us to use Anderson's algorithm (see Fig. 1),[62] which generates all the configurations of $n_\uparrow$ bits set to 1 in a bit string of length $n_\uparrow + n_\downarrow$ in lexicographical order. To illustrate how this algorithm proceeds, we show in Table I the step-by-step transformations of the variables with $n_\uparrow = 2$ and $n_\downarrow = 2$ from which the sequence (0011, 0101, 0110, 1001, 1010, 1100) is produced.

Figure 2 gives a pictorial description of the data structures used to generate a determinant. To build a generated determinant $(\mathtt{d}_\uparrow, \mathtt{d}_\downarrow)$ from a permutation u, one must

1. Fill the doubly occupied orbitals by setting both $\mathtt{d}_\uparrow$ and $\mathtt{d}_\downarrow$ equal to $\mathtt{p}_I^{(2)}$.
2. Iterate over the bits of u. If the k-th bit is set to 1, set the m_k-th orbital of $d_\uparrow$ to 1, otherwise set the m_k-th orbital of $d_\downarrow$ to 1.

TABLE I. Evolution of the values of the variables as the algorithm in Fig. 1 advances.

Iteration	t	t'	t''	u
0				0011
1	0011	0100	0001	0101
2	0101	0110	0000	0110
3	0111	1000	0001	1001
4	1001	1010	0000	1010
5	1011	1100	0000	1100

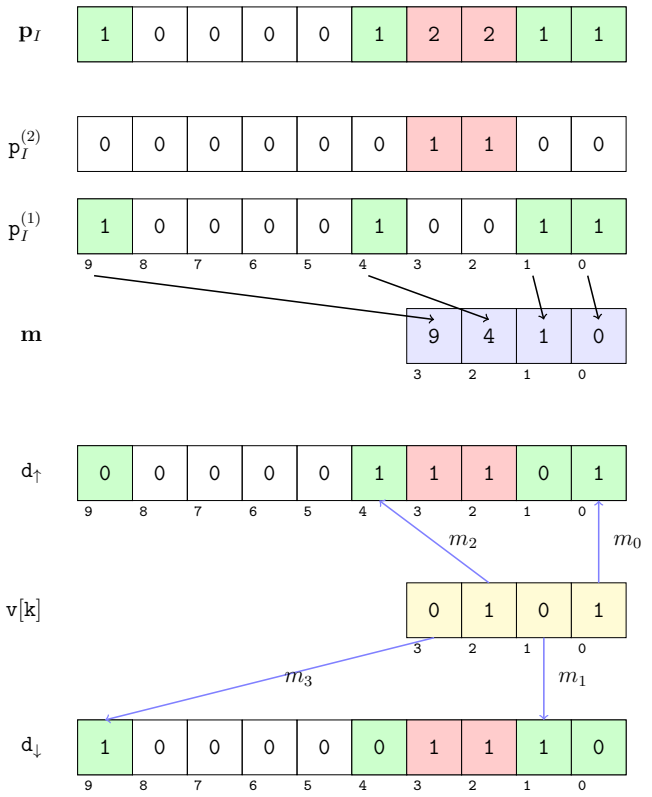

FIG. 2. The configuration $\mathbf{p}_I$ is encoded as in Eq. (7). Singly and doubly occupied orbitals are represented respectively in green and red. The list of indices **m** of the singly occupied orbitals is built (in blue), and this mapping is re-used to build the determinants from permutations (yellow) generated by Anderson's algorithm. Bit strings and arrays are represented from right to left to be consistent with the binary notation of Table I.

C. Further optimizations

As a first optimization, instead of creating each determinant from the permutation as shown in Fig. 2, all the determinants

can be generated iteratively by considering only the orbitals that have changed from the previously generated determinant. This avoids always setting all the $n_\uparrow + n_\downarrow$ bits in the bit strings. The integer obtained by $\mathtt{v[k-1]} \oplus \mathtt{v[k]}$ has bits set to 1 at the positions where the bits differ between $\mathtt{v[k-1]}$ and $\mathtt{v[k]}$. The positions of these bits can be found in a few cycles by

1. Counting the number of trailing zeros. This gives the position of the least significant 1.

2. Setting the least significant 1 to 0 using $\mathtt{v[k]} \leftarrow \mathtt{v[k]} \wedge (\mathtt{v[k]} - 1)$.

and iterating until $\mathtt{v[k]} = 0$.

A second optimization is to consider time-reversal symmetry (i.e., exchanging all spin-up and spin-down electrons in an even electron systems). When $n_\uparrow = n_\downarrow$, one can remark that $\mathtt{v}[\mathtt{n_{det}} - 1 - \mathtt{k}] = \neg\mathtt{v[k]}$, where $\mathtt{n_{det}}$ is the number of determinants generated:

$$\mathtt{n_{det}} = \frac{(n_\uparrow + n_\downarrow)!}{n_\uparrow! n_\downarrow!} \tag{8}$$

Hence, it is sufficient to iterate over the first half of the permutations of Anderson's algorithm, and generate pairs of determinants per iteration.

D. Reduction of the memory requirements

In the latest version of *Quantum Package*, the spin-pure eigenstates were obtained by finding the lowest eigenstate of a linear combination of the Hamiltonian and the $\hat{S}^2$ matrices.[58,63] At iteration n, the Davidson algorithm requires the computation of the matrix $\mathbf{W} = \mathbf{H}\,\mathbf{U}$, where $\mathbf{U}$ and $\mathbf{W}$ are $N_{\text{det}} \times N_{\text{states}}$ matrices, where N_{det} is the number of determinants, and N_{states} is greater than the number of states of interest and adjusted to reduce the number of iterations for the convergence of the algorithm.

In terms of storage, the $\mathbf{W}$ and $\mathbf{U}$ matrices of all n iterations need to be stored. As the storage increases with iterations, it is common practice to define a maximum iteration n_{max} where all the $\mathbf{U}$ matrices are compressed into a single $N_{\text{det}} \times N_{\text{states}}$ improved $\mathbf{U}$ matrix, and the algorithm restarts. If one wants to monitor the expectation value $\left\langle \hat{S}^2 \right\rangle$, one needs also to compute $\mathbf{Y} = \mathbf{S^2}\,\mathbf{U}$, and store the $\mathbf{Y}$ matrices of all iterations. As the computation of $\hat{S}^2$ is made only for monitoring purposes, the $\mathbf{Y}$ matrices can be stored in single precision to limit the increase in the memory requirements. Hence, in the determinant basis, the required space for the diagonalization is $2.5 \times N_{\text{det}} \times N_{\text{states}} \times n_{\text{max}}$.

Since the selected determinant space contains all the determinants of each configuration, we can make an exact transformation from the determinant basis to the CSF basis,[56,57] thus rendering the many-particle basis representation more compact. Hence, we now store the $\mathbf{U}$ and $\mathbf{W}$ matrices in the CSF basis, while the computation $\mathbf{W} = \mathbf{H}\,\mathbf{U}$ is still performed in the determinant basis for simplicity. As the wave function is guaranteed to be an eigenstate of $\hat{S}^2$, it is no longer necessary to compute and store $\mathbf{S^2}$, so the storage requirements are reduced to $2 \times N_{\text{det}} \times N_{\text{states}} + 2 \times N_{\text{CSF}} \times N_{\text{states}} \times n_{\text{max}}$, where $N_{\text{CSF}} \ll N_{\text{det}}$ is the number of CSFs (see below).

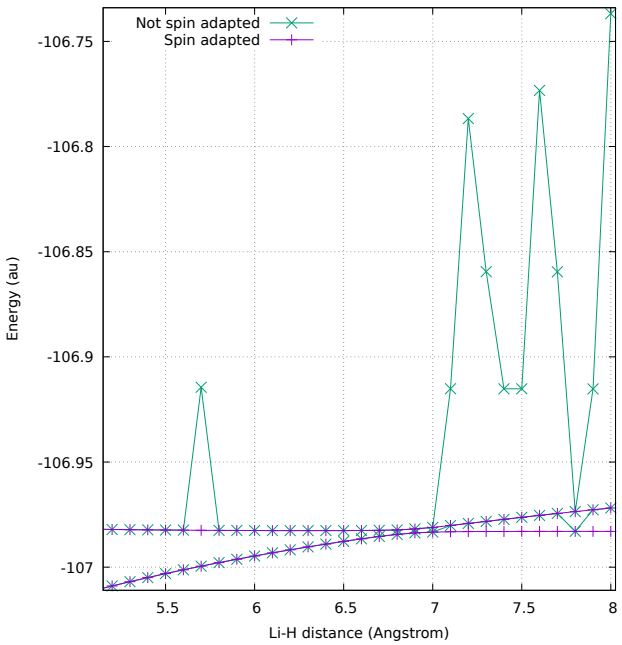

FIG. 3. Avoided crossing of LiF, with and without imposing spin symmetry. The energy (in hartree) of the two lowest singlet states of LiF is represented as a function of the bond length (in Å).

IV. NUMERICAL TESTS

In this section, the *configuration interaction using a perturbative selection made iteratively* (CIPSI) algorithm[3,58] is employed to select determinants of the external space: they are selected by the magnitude of their contribution to the second-order perturbative correction to the energy. The spin-adaptation step is introduced between the selection step and the diagonalization. We would like to emphasize that we use CIPSI because it is the method implemented in *Quantum Package*, but any CI or SCI could be have been considered.

A. Avoided crossing of LiF

The avoided crossing between the ionic and neutral $^1\Sigma^+$ states of LiF is a common benchmark for correlated methods, as the location of the crossing is highly sensitive to the amount of correlation.[64–66] At large distances, the lowest triplet state is very close in energy to the singlet states. If the wave function is not spin-adapted, the triplet state will mix with the singlets during the selection, and the convergence of the CIPSI calculation to the correct states is not guaranteed.

We report in Fig. 3 the potential energy curve of the two lowest singlet states of LiF computed with and without imposing spin adaptation. For all the distances, the CIPSI calcula-

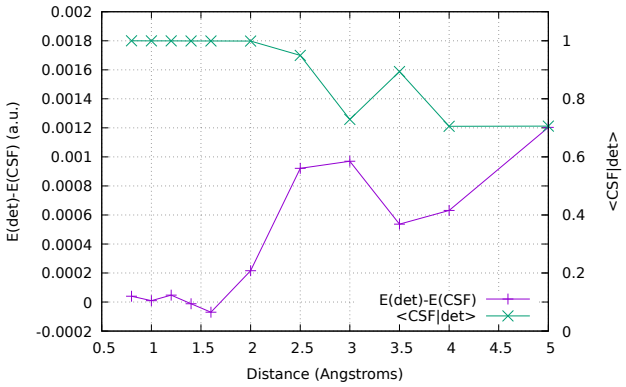

FIG. 4. Difference of extrapolated FCI energy obtained with and without spin-adaptation (purple) along the dissociation path of N_2 (aug-cc-pVDZ). The overlaps of the wave functions obtained with the two schemes are reported in green.

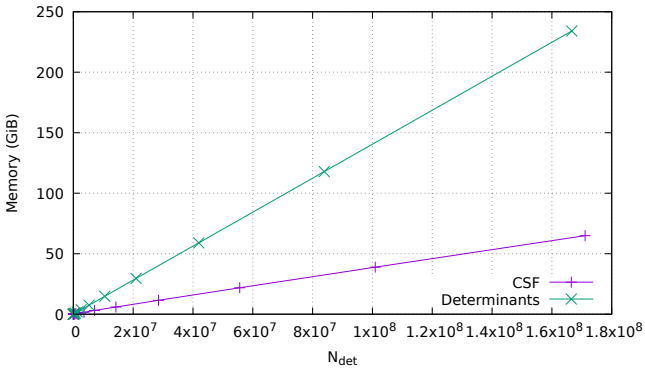

FIG. 5. Memory requirements (in GiB) for the Davidson algorithm with storage in the determinant basis or in the CSF basis as a function of N_{det} in the case of dissociated N_2 with the aug-cc-pVDZ basis.

tions were run *blindly* (with no user interaction), starting with the CASSCF(2,2)/aug-cc-pVDZ wave functions of both states (four determinants). Only the lowest molecular orbital was frozen, corresponding to the $1s$ orbital of the fluorine atom. The calculations were stopped when the second-order perturbative correction was below 0.1 mE_h or when the number of determinants reached 4 million.

Figure 3 shows that for large distances, without spin adaptation, there are multiple erratic points for which the two obtained states are not the desired ones. This curve also shows that all the points obtained with spin-adaptation converged to the correct states, giving a smooth potential energy curve.

B. Dissociation of N_2

Selected CI methods provide not only the energies of the states of interest, but also the corresponding wave functions which can be used for post-processing. For instance, wave functions computed with CIPSI have shown to be excellent choices of trial wave functions for quantum Monte Carlo calculations.[19,25,26,36,46,67] When a wave function is used for further calculations, the spin-adapted characteristic is particularly important because it can enforce a continuous behavior of wave function along a dissociation curve, especially when different spin states become quasi-degenerate.

To illustrate the importance of this feature, we compute the dissociation curve of the singlet ground state of the N_2 molecule with the aug-cc-pVDZ basis set, and estimate the frozen-core FCI energy by extrapolating the variational energy with respect to the renormalized second-order perturbative correction.[58] The curve is first computed using a simple determinant selection, minimizing the energy without considering the spin operator. Then, the curve is computed using the spin-adapted determinant selection, and we report in Fig. 4 the difference in extrapolated FCI energies, as well as the overlap between the two wave functions at each point of the curve. For each point of the curve, the calculation was stopped when the wave function was expanded on more than a million determinants. This corresponds to second-order perturbative energy corrections smaller than 0.012 hartree.

When the triple bond is broken, N_2 dissociates into two nitrogen atoms, each in its high-spin configuration. At dissociation, the two nitrogens can be combined in a singlet, in a triplet or in a quintet state, all with the same energy. Hence, without any particular treatment all these spin states mix together and produce spin contaminated wave functions. As the determinant-based Epstein-Nesbet perturbation theory is not invariant with respect to the magnetic quantum number m_s, spin contamination in the reference wave function affects the second-order perturbative correction and makes the extrapolations less accurate. This effect can be observed in Fig. 4 where for distances larger than 2 Å the overlap between the spin-adapted singlet wave function and the determinant wave function shows a significant spin contamination, leading to fluctuations in the extrapolated energies as large as a millihartree.

Figure 5 shows the memory requirements of the Davidson routines in the determinant-based and the CSF-based storage. From this figure, it is clear that storing the matrices in the CSF basis makes a big difference in terms of memory requirements, with a reduction by a factor 4 in the case of dissociated N_2 at an internuclear distance of 5 Åwith the aug-cc-pVDZ basis.

Figure 6 displays the ratio $N_{\text{det}}/N_{\text{CSF}}$ as a function of N_{det} for the same system, which clearly tells us that the number of determinants increases faster than the number of CSFs. This can be explained by the fact that upon starting with a closed shell reference, during the CIPSI selection, determinants with a large number of open shells appear later than determinants with mostly closed shells. Hence, we expect the reduction in terms of memory requirements to be increasingly notable as the number of CIPSI iterations increases.

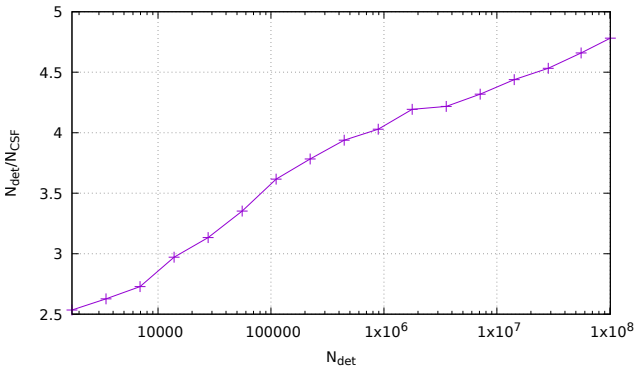

FIG. 6. Ratio of the number of determinants N_{det} over the number of CSFs N_{CSF}, as a function of the number of selected determinants in the case of dissociated N_2 with the aug-cc-pVDZ basis.

V. CONCLUSION

We have presented a general algorithm to complement an arbitrary wave function with all the required Slater determinants to obtain eigenstates of the $\hat{S}^2$ operator when the Hamiltonian is diagonalized, with negligible computational overhead. This spin adaptation step is introduced after the determinant selection of the SCI algorithm, and enables the possibility to switch to the CSF basis for the diagonalization of the Hamiltonian to reduce the memory requirements which is one of the limiting step is today's SCI algorithms. We would like to emphasize that this spin-adaptation procedure can be applied to any SCI-type method: CIPSI, SHCI, FCIQMC, *etc*. We hope to report further algorithmic improvements in the near future following the same philosophy.

ACKNOWLEDGMENTS

The authors gratefully acknowledge Sean Eron Anderson for creating the *Bit Twiddling Hacks*[62] web page.

This work is supported by the European Centre of Excellence in Exascale Computing TREX - Targeting Real Chemical Accuracy at the Exascale. This project has received funding from the European Union's Horizon 2020 - Research and Innovation program - under grant agreement no. 95216. AS and PFL were also supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 863481). Funding from Projet International de Coopération Scientifique (PICS08310) is acknowledged. KG acknowledges support from grant number CHE1762337 from the U.S. National Science Foundation. This research used resources of the Argonne Leadership Computing Facility, which is a U.S. Department of Energy Office of Science User Facility operated under contract DE-AC02-06CH11357. This work was performed using HPC resources from CALMIP (Toulouse) under allocation 2021-18005 and from GENCI-TGCC (Grant 2020-A0040801738).

[1] J. L. Whitten and M. Hackmeyer, J. Chem. Phys. **51**, 5584 (1969).
[2] C. F. Bender and E. R. Davidson, Phys. Rev. **183**, 23 (1969).
[3] B. Huron, J. P. Malrieu, and P. Rancurel, J. Chem. Phys. **58**, 5745 (1973).
[4] J. Greer, J. Comput. Chem. **146**, 181 (1998).
[5] M. Hanrath and B. Engels, Chem. Phys. **225**, 197 (1997).
[6] P. Stampfuß and W. Wenzel, J. Chem. Phys. **122**, 024110 (2005).
[7] L. Bytautas and K. Ruedenberg, Chem. Phys. **356**, 64 (2009).
[8] G. H. Booth, A. J. W. Thom, and A. Alavi, J. Chem. Phys. **131**, 054106 (2009).
[9] E. Giner, A. Scemama, and M. Caffarel, Can. J. Chem. **91**, 879 (2013).
[10] R. J. Buenker, R. A. Phillips, S. Krebs, H.-P. Liebermann, A. B. Alekseyev, and P. Funke, Theor. Chem. Acc. **133**, 1468 (2014).
[11] A. A. Holmes, N. M. Tubman, and C. J. Umrigar, J. Chem. Theory Comput. **12**, 3674 (2016).
[12] W. Liu and M. Hoffmann, Theor. Chem. Acc. **133**, 1481 (2014).
[13] W. Liu and M. R. Hoffmann, J. Chem. Theory Comput. **12**, 1169 (2016).
[14] Y. Lei, W. Liu, and M. R. Hoffmann, Mol. Phys. **115**, 2696 (2017).
[15] N. Zhang, W. Liu, and M. R. Hoffmann, J. Chem. Theory Comput. **16**, 2296 (2020).
[16] P. M. Zimmerman, J. Chem. Phys. **146**, 104102 (2017).
[17] Y. Ohtsuka and J. ya Hasegawa, J. Chem. Phys. **147**, 034102 (2017).
[18] J. P. Coe, J. Chem. Theory Comput. **14**, 5739 (2018).
[19] A. Scemama, Y. Garniron, M. Caffarel, and P.-F. Loos, J. Chem. Theory Comput. **14**, 1395 (2018).
[20] F. A. Evangelista, J. Chem. Phys. **140**, 124114 (2014).
[21] H.-J. Flad, T. Rohwedder, and R. Schneider, in *Progress in Physical Chemistry Volume 3* (Oldenbourg Wissenschaftsverlag GmbH, 2010) pp. 361–379.
[22] J. J. Eriksen, T. A. Anderson, J. E. Deustua, K. Ghanem, D. Hait, M. R. Hoffmann, S. Lee, D. S. Levine, I. Magoulas, J. Shen, N. M. Tubman, K. B. Whaley, E. Xu, Y. Yao, N. Zhang, A. Alavi, G. K.-L. Chan, M. Head-Gordon, W. Liu, P. Piecuch, S. Sharma, S. L. Ten-no, C. J. Umrigar, and J. Gauss, J. Phys. Chem. Lett. **11**, 8922 (2020).
[23] Y. Garniron, A. Scemama, E. Giner, M. Caffarel, and P. F. Loos, J. Chem. Phys. **149**, 064103 (2018).
[24] P.-F. Loos, Y. Damour, and A. Scemama, J. Chem. Phys. **153**, 176101 (2020).
[25] A. Scemama, E. Giner, A. Benali, and P.-F. Loos, J. Chem. Phys. **153**, 174107 (2020).
[26] A. Benali, K. Gasperich, K. D. Jordan, T. Applencourt, Y. Luo, M. C. Bennett, J. T. Krogel, L. Shulenburger, P. R. C. Kent, P.-F. Loos, A. Scemama, and M. Caffarel, J. Chem. Phys. **153**, 184111 (2020).
[27] J. B. Schriber and F. A. Evangelista, J. Chem. Phys. **144**, 161106 (2016).
[28] J. Li, M. Otten, A. A. Holmes, S. Sharma, and C. J. Umrigar, J. Chem. Phys. **149**, 214110 (2018).
[29] Y. Yao, E. Giner, J. Li, J. Toulouse, and C. J. Umrigar, J. Chem. Phys. **153**, 124117 (2020).
[30] J. Li, Y. Yao, A. A. Holmes, M. Otten, Q. Sun, S. Sharma, and C. J. Umrigar, Phys. Rev. Research **2**, 012015 (2020).
[31] K. T. Williams, Y. Yao, J. Li, L. Chen, H. Shi, M. Motta, C. Niu, U. Ray, S. Guo, R. J. Anderson, J. Li, L. N. Tran, C.-N. Yeh, B. Mussard, S. Sharma, F. Bruneval, M. van Schilfgaarde, G. H. Booth, G. K.-L. Chan, S. Zhang, E. Gull, D. Zgid, A. Millis, C. J. Umrigar, and L. K. Wagner, Phys. Rev. X **10**, 011041 (2020).
[32] J. P. Coe and M. J. Paterson, J. Chem. Phys. **139**, 154103 (2013).
[33] J. B. Schriber and F. A. Evangelista, J. Chem. Theory Comput. **13**, 5354 (2017).
[34] A. A. Holmes, C. J. Umrigar, and S. Sharma, J. Chem. Phys. **147**, 164111 (2017).
[35] P.-F. Loos, A. Scemama, A. Blondel, Y. Garniron, M. Caffarel, and D. Jacquemin, J. Chem. Theory Comput. **14**, 4360 (2018).
[36] A. Scemama, A. Benali, D. Jacquemin, M. Caffarel, and P.-F. Loos, J. Chem. Phys. **149**, 034108 (2018).
[37] M. Dash, S. Moroni, A. Scemama, and C. Filippi, J. Chem. Theory Comput. **14**, 4176 (2018).
[38] A. D. Chien, A. A. Holmes, M. Otten, C. J. Umrigar, S. Sharma, and P. M. Zimmerman, J. Phys. Chem. A **122**, 2714 (2018).
[39] P.-F. Loos, M. Boggio-Pasqua, A. Scemama, M. Caffarel, and D. Jacquemin, J. Chem. Theory Comput. **15**, 1939 (2019).

[40] P.-F. Loos, A. Scemama, M. Boggio-Pasqua, and D. Jacquemin, J. Chem. Theory Comput. **16**, 3720 (2020).
[41] P.-F. Loos, A. Scemama, and D. Jacquemin, J. Phys. Chem. Lett. **11**, 2374 (2020).
[42] P. F. Loos, F. Lipparini, M. Boggio-Pasqua, A. Scemama, and D. Jacquemin, J. Chem. Theory Comput. **16**, 1711 (2020).
[43] M. Véril, A. Scemama, M. Caffarel, F. Lipparini, M. Boggio-Pasqua, D. Jacquemin, and P.-F. Loos, WIREs Comput. Mol. Sci. **n/a**, e1517.
[44] M. Dash, J. Feldt, S. Moroni, A. Scemama, and C. Filippi, J. Chem. Theory Comput. **15**, 4896 (2019).
[45] E. Giner, A. Scemama, J. Toulouse, and P.-F. Loos, J. Chem. Phys. **151**, 144118 (2019).
[46] A. Scemama, M. Caffarel, A. Benali, D. Jacquemin, and P.-F. Loos, Res. Chem. **1**, 100002 (2019).
[47] N. S. Blunt, S. D. Smart, G. H. Booth, and A. Alavi, J. Chem. Phys. **143**, 134117 (2015).
[48] F. Neese, J. Chem. Phys. **119**, 9428 (2003).
[49] J. E. T. Smith, B. Mussard, A. A. Holmes, and S. Sharma, J. Chem. Theory Comput. **13**, 5468 (2017).
[50] W. Dobrautz, S. D. Smart, and A. Alavi, J. Chem. Phys. **151**, 094104 (2019).
[51] G. Li Manni, W. Dobrautz, and A. Alavi, J. Chem. Theory Comput. **16**, 2202 (2020).
[52] V. G. Chilkuri and F. Neese, J. Comput. Chem. (forthcoming 2020).
[53] V. G. Chilkuri and F. Neese, J. Chem. Theory Comput. (forthcoming 2020).
[54] L. Bytautas and K. Ruedenberg, in *ACS Symposium Series* (American Chemical Society, 2007) pp. 103–123.
[55] R. Pauncz, *Spin Eigenfunctions: Construction and Use* (Springer US, 2012).
[56] J. Olsen, J. Chem. Phys. **141**, 034112 (2014).
[57] B. S. Fales and T. J. Martínez, J. Chem. Phys. **152**, 164111 (2020).
[58] Y. Garniron, T. Applencourt, K. Gasperich, A. Benali, A. Ferté, J. Paquier, B. Pradines, R. Assaraf, P. Reinhardt, J. Toulouse, P. Barbaresco, N. Renon, G. David, J.-P. Malrieu, M. Véril, M. Caffarel, P.-F. Loos, E. Giner, and A. Scemama, J. Chem. Theory Comput. **15**, 3591 (2019).
[59] R. Pauncz, Int. J. Quantum Chem. **35**, 717 (1989).
[60] A. Scemama and E. Giner, arXiv (2013), 1311.6244.
[61] D. Bitton and D. J. DeWitt, ACM Trans. Database Syst. **8**, 255 (1983).
[62] "Bit twiddling hacks," (2021), accessed Jan 15 2021.
[63] B. S. Fales, E. G. Hohenstein, and B. G. Levine, J. Chem. Theory Comput. **13**, 4162 (2017).
[64] D. Casanova, J. Chem. Phys. **137**, 084105 (2012).
[65] Ö. Legeza, J. Röder, and B. A. Hess, Mol. Phys. **101**, 2019 (2003).
[66] Y. Garniron, E. Giner, J.-P. Malrieu, and A. Scemama, J. Chem. Phys. **146**, 154107 (2017).
[67] M. Caffarel, T. Applencourt, E. Giner, and A. Scemama, J. Chem. Phys. **144**, 151103 (2016).