

Manual for DMRG code

Laëtitia Bomble

Contents

Table of contents	2
Introduction	3
1 Install, run and test the code	4
2 DMRG theory and algorithm	4
2.1 How does it works [13]	4
2.2 Overview of the algorithm [5, 11]	5
2.3 Some equations	6
2.3.1 All molecular orbitals will be activated by the DMRG algorithm .	6
2.3.2 Some orbitals remain unactivated in the DMRG computation . .	8
2.3.3 Mulliken Population analysis	9
3 How to make an input	9
3.1 dmrg.in input file	9
3.1.1 Format	9
3.1.2 Options and Settings	12
3.2 integral.dat input file	13
3.3 overlap.dat input file	14
3.4 atoms.dat input file	14
3.5 cmohf.dat input file	15
4 Output file	15
5 About the code itself	15
A Generate inputs with the Dalton electronic structure code	16

B	To not modify input files	17
B.1	model.in file	17
B.2	aic_ai.dat file	17
B.3	ai+1c_ai+1.dat file	18
B.4	aic_ai+1.dat file	18
B.5	aic_ai+1c_ai_ai+1.dat file	18
B.6	aic.dat file	18
B.7	ai+1c.dat file	18
B.8	aic_ai+1c.dat file	18
B.9	aic_ai_ai+1.dat file	19
B.10	ai+1c_ai+1_ai.dat file	19
C	Example for system specific files	19
C.1	Dalton input files	19
C.1.1	TMM.mol	19
C.1.2	TMM.dal	21
C.2	DMRG input files	21
C.2.1	integral.dat file	21
C.3	dmrg.in file for singlet state	23
C.4	dmrg.in file for triplet state	23
C.5	dmrg.in file for singlet state with Mulliken population analysis	24
C.6	atoms.dat file for singlet state with Mulliken population analysis	25
C.7	overlap.dat file for singlet state with Mulliken population analysis	26
C.8	cmohf.dat file for singlet state with Mulliken population analysis	26
	References	27

Introduction

The aim of electronic structure methods is to provide accurate approximation to the solutions of the Schrödinger equation to enable the treatment of many-electron systems. A well known approximation is the Hartree-Fock approximation which assumes that the wave function can be expressed as a single Slater determinant and uses a mean field approximation to describe the electronic interaction. The deviation from the true energy arising from this description of electronic interaction, is called the correlation energy. It is made of two parts, the dynamic and the non-dynamic (static) correlation. Static correlation is important in systems with several near-degenerated electronic states. Dynamic correlation deals with improvement of the reference wave-function by accounting for excitation out of the reference to high energy virtual orbitals. DMRG is particularly suited to deal with static correlation. This energy is most of the time small, but in some systems (energy dissociation of stretched H_2 for example) it cannot be neglected. Thus, the multi-electron wave function must be described as a combination of Slater determinants leading to the so called multireference methods. The conceptually simplest way to proceed is the Full Configuration Interaction method (FCI) where the wave function is computed as a combination of all the Slater determinants available in the chosen atomic basis. The FCI solution is exact within this atomic basis. Unfortunately, since the number of determinants grows factorially with the number of atomic orbitals and electrons, this method is only tractable for very small molecules. On the other hand, it is believed that it should be possible to describe a system with a number of parameters proportional to the size of the system, this is known as the locality principle. Then, even if the wave function of the system is expanded on an exponential number of determinants, their coefficients expansion could be parametrized by a reduced number of elements. Density Matrix Renormalization Group theory offers a way of executing this parametrization and thus overcoming the factorial growth issue.

DMRG was developed by White and al. in the 1990's to study solid state physics problem and was applied to quantum chemistry a few years later [13]. It allows us to treat large scale strongly correlated systems. This is done by an integration of the degrees of freedom of the Hamiltonian leading to a reduction of the number of parameters used to characterize the wave function. DMRG is then a variational, size consistent, multi-determinantal, polynomially scaling method. An efficient DMRG algorithm has been proposed by Chan and Head-Gordon in 2001 [5]. The code presented here implements this algorithm. Unfortunately, DMRG is still at a development stage, so this algorithm can not be used as a black box. It is then mandatory to know a bit about DMRG theory before running DMRG computations. Therefore, in this manual a brief overview

of theory is presented before explaining how to use this code.

1 Install, run and test the code

This DMRG code is written in Fortran 90 and thus requires a Fortran compiler. It has no dependencies.

The code can be compiled using the `compile.sh` file provided in the distribution.

DMRG test computation is provided and can be ran using the `run_test` file in `TEST` directory. DMRG code can also be executed using the command:

`./dmrg_qc.x < dmrg.in > dmrg.out`, `dmrg.in` is the master input file described in section 3.1 and `dmrg.out` the output file described in section 4.

A part of the inputs must be generated by an electronic structure code. We propose a patch to include to Dalton electronic structure code [1] to generated the necessary data. These files can be find in `DALTON` directory. How to generate necessary inputs with these is explained in section A.

2 DMRG theory and algorithm

2.1 How does it works [13]

The DMRG wave function is a linear combination of many-particle states. These states are themselves a linear combination of Slater determinant constructed from a given set of orbitals.

$$\Psi = \sum_{i=1,M} C_i \Theta = \sum_{i=1,M} \sum_{j=1,m_{FCI}} B_{ij} \Phi_j \quad (1)$$

All determinants corresponding to the chosen orbital basis are taken into account, the approximation arises in the multi-particle states truncation. The M value is either chosen by the user or dynamicaly determine to tune the energy error (Dynamical Block State Selection see page 8).

The DMRG algorithm uses as a starting point molecular spin-orbitals (SO) generated by an external electronic structure code. The system is then visualized as a 1D-lattice of this spin-orbitals (called sites). In the formalism of second quantization, each active site can be in two states: empty or occupied. The energy of the total system is calculated step by step by increasing the number of active sites at each step. The number of determinants considered (not always selected) increases during this process. We implemented the 2-dots DMRG algorithm, where the active block is enlarged two sites by two sites. The remaining sites form the environment block (see figure 1). An approximate total

Hamiltonian is calculated giving an approximate energy. The decimation process reduces the dimension of the active block before a new enlargement. At the end, all sites have been activated. This constitute one forward sweep. Then, we continue with a backward sweep where environment and active blocks have been exchanged. And so on until convergence.

2.2 Overview of the algorithm [5, 11]

At each step, the total system is divided into four blocks (see figure 1):

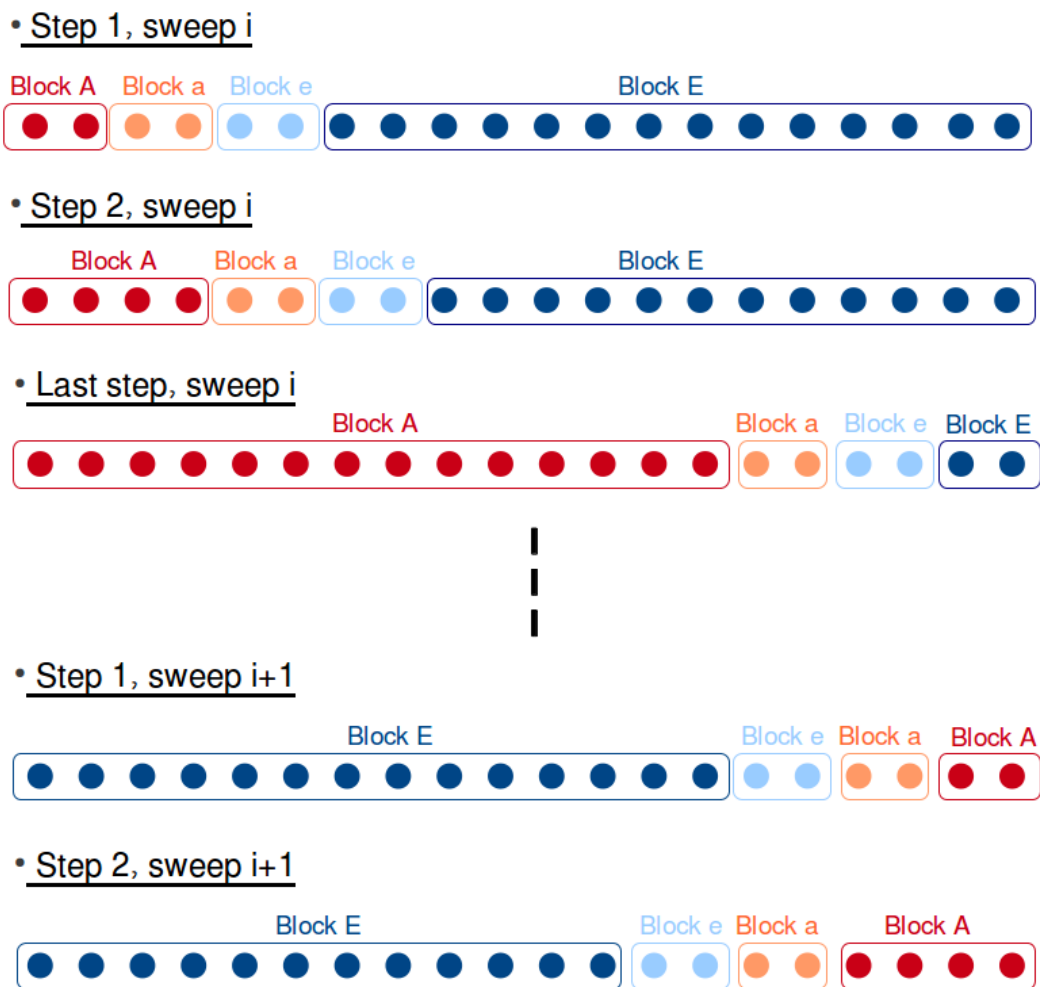


Figure 1: Dots represent spin-orbitals, called sites. The system is divided into four blocks, block A is enlarged step by step.

- Block A: block with active orbitals, constructed in the previous step, described by a renormalized Hamiltonian H_A .
- Block a: block with active orbitals to add to block A, described at FCI level by Hamiltonian H_a .
- Block E: block with environment orbitals, constructed in the previous sweep, described by a renormalized Hamiltonian H_E .
- Block e: block with environment orbitals to add to block E, described at FCI level by Hamiltonian H_e .

Before starting the iterations, an *ordering for orbital activation* must be selected (several options are available see p. 12). Then, step by step, orbitals will be activated until the end of the lattice is reached. This is called a sweep. Then, another sweep is conducted, starting from the opposite side of the lattice (see fig. 1). The main actions performed at each step of each sweep are the following:

1. Initialize the system: Compute exact Hamiltonian for block a and e, load Hamiltonian for block A and E.
2. Enlargement: Block a is added to block A, block e to block E to create blocks A' and E', computation of Hamiltonians for A' and E'.
3. Super-Hamiltonian: Computation and diagonalization of an approximate Hamiltonian describing the total system.
4. Decimation: Computation of a renormalized Hamiltonian for A' that will be used in next step. Includes:
 - Computation and diagonalization of density matrix projected on A'
 - Selection of the M eigenvectors with the M largest eigenvalues
 - Use of these eigenvectors to generate the new multireference states for A' from the ones computed in enlargement step.

2.3 Some equations

2.3.1 All molecular orbitals will be activated by the DMRG algorithm

The second-quantized Hamiltonian is written (see [3] for details about second quantization):

$$H = \sum_{ij} t_{ij} a_i^\dagger a_j + \frac{1}{2} \sum_{ijkl} v_{ijkl} a_i^\dagger a_j^\dagger a_l a_k$$

t_{ij} are the one-electron integrals, v_{ijkl} the two-electron integrals in physicist notation. When block a is added to block A, leading to the enlarged block A', the enlarged active Hamiltonian is: $H_{A'} = H_A + H_a + H_{Aa}$, with H_{Aa} the interaction Hamiltonian between block A and a. In the same way, the Hamiltonian of enlarged block B' (block B + block b) is: $H_{B'} = H_B + H_b + H_{Bb}$. The approximate total Hamiltonian is computed by: $H = H_{A'} + H_{B'} + H_{A'B'}$, with $H_{A'B'}$ the interaction between A' and B'. Interaction Hamiltonian between block P and Q can be written:

$$\begin{aligned}
H_{IJ} = \frac{1}{2} & \left[\sum_{i \in P, j \in Q} t_{ij} a_i^\dagger a_j + \sum_{i \in P, j \in Q} t_{ji} a_j^\dagger a_i \right. \\
& + \sum_{(i,j) \in P, (k,l) \in Q} v_{i,j,k,l} a_i^\dagger a_j^\dagger a_l a_k + \sum_{(i,j) \in Q, (k,l) \in P} v_{i,j,k,l} a_k^\dagger a_l^\dagger a_j a_i \\
& + \sum_{(i,j) \in P, (k,l) \in Q} (v_{i,j,k,l} - v_{kilj} - v_{ikjl} + v_{kijl}) a_i^\dagger a_j a_k^\dagger a_l \\
& + \sum_{i \in P, (j,k,l) \in Q} (v_{i,j,k,l} - v_{j,i,k,l}) a_i^\dagger a_j^\dagger a_l a_k + \sum_{i \in P, (j,k,l) \in Q} (v_{i,j,k,l} - v_{j,i,k,l}) a_k^\dagger a_l^\dagger a_j a_i \\
& \left. + \sum_{i \in Q, (j,k,l) \in P} (v_{i,j,k,l} - v_{j,i,k,l}) a_i^\dagger a_j^\dagger a_l a_k + \sum_{i \in Q, (j,k,l) \in P} (v_{i,j,k,l} - v_{j,i,k,l}) a_k^\dagger a_l^\dagger a_j a_i \right]
\end{aligned}$$

We can see that to construct the block Hamiltonians and to enlarged then, we need the expression of several products of elementary creation and annihilation operators. These operators are the keys of DMRG algorithm. Indeed, the DMRG wave function is never explicitly given in the original molecular orbital basis, thus DMRG is a purely numerical method, and the operators contains the information relative to the combination of states and will be improved step by step.

All necessary operators will be computed from 9 ones that need to be initialized for a two sites (i and i+1) space (in a 2-dots DMRG algorithm case): a_i^\dagger , a_{i+1}^\dagger , $a_i^\dagger a_i^\dagger$, $a_i^\dagger a_{i+1}^\dagger$, $a_i^\dagger a_{i+1}^\dagger a_i^\dagger$, $a_{i+1}^\dagger a_{i+1}^\dagger$, $a_{i+1}^\dagger a_{i+1}^\dagger a_i^\dagger$, $a_i^\dagger a_i a_{i+1}^\dagger$ and $a_i^\dagger a_{i+1}^\dagger a_i a_{i+1}^\dagger$. To have more information about how these operators are computed and store see [5, 9].

Once the super (total) Hamiltonian has been computed and diagonalized, the projected density matrix on active block is constructed from the ground state wave function Ψ_0 . Ψ_0 can be expressed in the product basis of the active block and environment block basis:

$$\Psi_0 = \sum_{i \in A', j \in E'} c_{ij} |A'_i\rangle |E'_j\rangle$$

Thus, the projected density matrix expression is:

$$D_{A'} = Tr_{E'} |\Psi_0\rangle \langle \Psi_0| D_{A'}(i, j) = \sum_{i,j \in A, k \in E} c_{ik} c_{jk} |A_i\rangle \langle A_j|$$

$D_{A'}$ is diagonalized and M eigenvectors are kept to form the projection matrix P . M is either given by user or dynamically selected (Dynamical Block State Selection) to have an error on DMRG energy below a threshold chosen by user [7]. To maximize the number of M selected (*i.e.* to maximize the non-zero eigen values of $D_{A'}$), a random noise can be added to elements of $D_{A'}$ [8].

Finally, the Hamiltonian and operators of the enlarged active block are decimated using P matrix: $O_A^{\text{new}} = P^T O^{\text{old}} P$. The M new states are combination of old ones but only states with the same quantum numbers (spin and number of electrons) are mixed. The renormalized block A' is the block A to used in the following step.

The Hamiltonian and operators of the environment block are guessed for the first sweep: the monodeterminantal Hamiltonian from the external electronic structure computation is used. In a sweep $s > 1$, Hamiltonian from step $s-1$ is loaded and used.

2.3.2 Some orbitals remain unactivated in the DMRG computation

User can decided a part of the spin-orbitals will not be activated by DMRG algorithm and that their contribution will remain at a monodeterminantal level (the one chosen in the external electronic structure code). Then the total Hamiltonian can be written:

$$H = H_{DMRG} + E_{unact} + H_{DMRG-unact}$$

H_{DMRG} is the Hamiltonian calculated by DMRG ansatz and described above (see section 2.3.1), E_{unact} is the energy of the frozen part which is fixed and given by the external electronic structure code:

$$E_{unact} = 2 \sum_{i \in unact} h_{ii} + \sum_{i,j \in unact} 2J_{ij} - K_{ij}$$

$H_{DMRG-unact}$ is the interaction between the activated and unactivated parts, in second quantization:

$$H_{DMRG-unact} = \sum_{i,j \in act, k \in unact} (v_{ikjk} - v_{ikkj}) a_i^\dagger a_j \quad (2)$$

$$= \sum_{i,j \in act} V_{ij}^{unact} a_i^\dagger a_j, \quad \text{with} \quad V_{ij}^{unact} = \sum_{k \in unact} (v_{ikjk} - v_{ikkj}) \quad (3)$$

E_{unact} and V_{ij}^{unact} have to be generated by the external electronic structure code.

2.3.3 Mulliken Population analysis

To get Mulliken charges, we first need to compute the 1-electron Reduced Density-Matrix. In second quantization it is expressed as:

$$1\text{RDM} = \langle \Psi_0 | A^\dagger A | \Psi_0 \rangle, \text{ with } A^\dagger A(i, j) = a_i^\dagger a_j, \text{ } i, j \text{ are SO.}$$

Then, 1RDM is expressed in the atomic orbital basis:

$$D(\mu, \nu) = \sum_{i,j} b_{\mu i} 1\text{RDM}_{ij} b_{\nu j}$$

Finally, Mulliken electronic charge on atom A (MC_A) is:

$$MC_A^e = \sum_{\mu \in A} (DS)_{\mu\mu}$$

S is the overlap matrix in AO basis.

If some orbitals are not activated by DMRG algorithm, Mulliken electronic charges arising from frozen orbitals must be added to these ones.

3 How to make an input

To run, the DMRG algorithm needs integrals and orbitals generated by another electronic structure code. For a DMRG computation without Mulliken Population Analysis, one and two electrons integrals between the molecular orbitals selected (t_{ij} and v_{ijkl}) as well as some between selected and unactivated ones (V_{ij}^{unact}) must be generated. For Mulliken population analysis, overlap integrals between AO's as well as MO's coefficient are demanded.

A summary of input files can be found in table 1. The following sections describe with more details how these ones must be set.

3.1 dmrg.in input file

`dmrg.in` contains all the settings and options, as well as the location of the other input files. The structure of this file is given in this section, as well as the settings and options.

3.1.1 Format

`dmrg.in` is organized as follow:

- * `lprint`: If `.true.`, print details, mainly for debugging purpose.

INPUTS TO NOT MODIFY (see appendix B)	
model.in	List of to be read Hamiltonians
aic.dat	a_i^\dagger
ai+1c.dat	a_{i+1}^\dagger
aic_ai.dat	$a_i^\dagger a_i^\dagger$
aic_ai+1.dat	$a_i^\dagger a_{i+1}^\dagger$
aic_ai+1c.dat	$a_i^\dagger a_{i+1}^\dagger$
ai+1c_ai+1.dat	$a_{i+1}^\dagger a_{i+1}$
ai+1c_ai+1_ai.dat	$a_{i+1}^\dagger a_{i+1} a_i$
aic_ai_ai+1.dat	$a_i^\dagger a_i a_{i+1}$
aic_ai+1c_ai_ai+1.dat	$a_i^\dagger a_{i+1}^\dagger a_i a_{i+1}$
INPUTS GENERATED BY EXTERNAL ELECTRONIC STRUCTURE CODE	
integral.dat	One and two electron integrals and integrals V_{ij}^{unact} .
overlap.dat (for MPA)	Overlap integrals between AO's
cmohf.dat (for MPA)	MO's coefficients
atoms.dat (for MPA)	Cross reference between atoms and AO's
MASTER INPUT	
dmrg.in	See section 3.1

Table 1: List of input files to provided DMRG code

- * `ldbss`: If `.true.` use DBSS (see p.8), if `.false.` Use a given renormalization number.
- * `check_order`: Orbital ordering, see section 3.1.2.
- * `ldavidson`: If `.true.`, use Davidson algorithm for diagonalization[12]
- * If DBSS is:
 - $\left\{ \begin{array}{l} \text{.true.: } \text{rho_cut, err_min, emme_min: threshold for } D_{A'} \text{ terms, maximal error} \\ \text{and minimal } M \text{ value} \\ \text{else: } \text{rho_cut, emme, threshold for terms of } D_{A'}, \text{ maximal } M \text{ value} \end{array} \right.$
- * `noise`: noise added to $D_{A'}$
- * `e111`: System size (number of spin orbitals)
- * `d`: Dimension of an elementary block

- * `nsweep`: Maximun sweep number
- * `NumHamLoc`: Number of read operators

- * `spin_tot`: $2 \times$ Total spin
- * `ne_tot`: Number of electrons
- * `locc_auto`: If `.true.`, initial orbital population generated by DMRG code (only for singlet), if `.false.`, initial population to be given by user.
- * If no auto initial population:
 - $\left\{ \begin{array}{l} \text{n_doubly, n_simply_up, n_simply_down :} \\ \text{Number of: doubly occupied, simply occupied with spin up,} \\ \text{simply occupied with spin down orbitals.} \\ \text{occ_doubly: List of doubly occupied orbitals (if none put zero).} \\ \text{occ_simply_up: List of simply occupied with spin up orbitals (if none put zero).} \\ \text{occ_simply_down: List of simply occupied with spin down orbitals (if none put zero).} \end{array} \right.$
- * `lmpa`: If `.true.` compute Mulliken Population Analysis.
- * If MPA, `natom, nbasis`: Number of atoms, number of atomic orbitals

- * `dirname`: Directory adress
- * `FolderIn`: Name of input folder in `/home/DMRG/`
- * `filemodel`: Name of model file

- * `fileintegral`: Name of integral file
 - * If MPA:
 - `fileatoms`: Name of cross reference file
 - `fileoverlap`: Name of overlap file
 - `filecmohf`: Name of MO's coefficient file
-

For an example, see section C.3

3.1.2 Options and Settings

Orbital activation ordering: There are several choices available to order the orbitals along the lattice:

- `E_HF_1`: First all the α spin-orbitals ordered by molecular orbital energy, then all the β spin-orbitals ordered by molecular orbital energy.
- `E_HF_2`: Orbitals ordered by molecular spin-orbital energy, α and β spin-orbitals with same energy are placed one next to another \rightarrow alternation of α and β spin-orbitals.
- `RCM_TIJ_1`: Orbitals ordered to gather orbitals strongly interacting through t_{ij} using Reverse Cuthill-McKee algorithm [5, 4]. First all α are placed with his ordering, then all β .
- `RCM_TIJ_2`: Orbitals ordered to put together orbitals interacting through t_{ij} using a RCM algorithm [5, 4]. Alternation of α and β spin-orbitals.
- `RCM_KIJ_1`: Orbitals ordered to put together orbitals interacting through k_{ij} using a RCM algorithm [10, 4]. First all α are placed with this ordering, then all β .
- `RCM_KIJ_2`: Orbitals ordered to put together orbitals interacting through k_{ij} using a RCM algorithm [10, 4]. Alternation of α and β spin-orbitals.

Noise: Among the available orbital orderings in this code or presented in litterature, there is no way to know *a priori* which one is the best for a specific study. Given the cost of a DMRG computation, all options can not be tried. Thus, the ordering chosen may not be the best one and may lead to a local minima. To fixed this issue, Mitrushenkov and *al* [8] proposed to add a noise to the elements of the reduced density matrix to prevent DMRG algorithm to get stuck in such minima. Before diagonalization, the following operation is performed:

$$D_{A'}(i, j) = D_{A'}(i, j) + noise \times r(i, j) \quad \text{with } -1 < r(i, j) < 1$$

Davidson diagonalization: After a few sweeps, the super-Hamiltonian is too big to be diagonalized exactly. A diagonalization using Davidson algorithm [12] is available and recommended.

DBSS: In standard DMRG procedure, after D_A diagonalization, the M (fixed by user) eigenstates with the M largest eigenvalues (ω_i , $i = 1, M$) are selected for decimation. The error induced by this truncation is $TrE_M = 1 - \sum_{i=1,M} \omega_i$ [6]. If all states were kept, the error would be zero and the energy found would be FCI energy. As M grows larger, TrE_M decreases and E_M get closer to E_{FCI} : $\ln|TrE_M| \sim -\kappa(\ln M)^2$ [5]. Thus, $E_M(TrE_M)$ can be extrapolated to get FCI energies.

Another approach, proposed by Legeza and *al.* [7], is to fix the truncation error instead of M value. At each DMRG step, M is dynamically adapted to keep the error below a chosen threshold. Then, the parameter of the computation is the accuracy of the computation and $E_M(TrE_M)$ can directly be extrapolated to zero error.

Number of read operators: This operators are the nine initial operators (a_i^\dagger , a_{i+1}^\dagger , $a_i^\dagger a_i^\dagger$, $a_i^\dagger a_{i+1}^\dagger$, $a_i^\dagger a_{i+1}^\dagger$, $a_{i+1}^\dagger a_{i+1}^\dagger$, $a_{i+1}^\dagger a_{i+1}^\dagger a_i$, $a_i^\dagger a_i a_{i+1}^\dagger$) from which all the necessary ones to compute Hamiltonians will be derived (see p. 7).

Initial population: To initialize the environment state, the DMRG algorithm needs a initial way of filling spin-orbital. If a singlet state is wanted, the code can guess this population. The first $\frac{n_{electrons}}{2}$ orbitals issued from the *ad hoc* computation will be doubly occupied. If another filling is required (mandatory for non-singlet states), the user can decided which orbitals are filled and if they are simply or doubly filled. The DMRG code will then ‘translate’ this spatial orbital occupancy into the spin-orbital one.

3.2 integral.dat input file

`integral.dat` hold all integrals between molecular orbitals. This file must be generated by an external electronic structure code. The given values are resulting from integration of one and two electrons operators between spatial molecular orbitals in chemist notation. DMRG code will compute from these, spin-orbital molecular orbital integrals in physicist notation. The file is organized as follow:

- line 1:** If some orbitals are not activated by DMRG code, the energy contribution of these: E_{unact} must be written on the first line of `integral.dat` (see section 2.3.2).
- line 2:** Line of comments

- line 3 - line n^2+3** : $i j t_{ij}$, i and j both running on all orbitals, on line 3 the 1 1, on line 4 the 1 2 and so on until $n n$
- line n^2+4 - line n^2+5** : Two lines of comments
- line n^2+6 - line x** : $i j k l v_{ijkl}$, with $l \leq k$, $j \leq i$, i and k both running on all orbitals. For example with two orbitals: element 1 1 1 1, then 1 1 2 1, then 1 1 2 2, then 2 1 1 1, then 2 1 2 1, then 2 1 2 2, and finally 2 2 2 2.
- line $x+1$ - line $x+2$** : Two lines of comments
- line $x+3$ - end**: If some orbitals are not activated by DMRG code, $i j V_{unact_{ij}}$, i and j both running on all orbitals (see section 2.3.2).

3.3 overlap.dat input file

This file is only used if computation of Mulliken Population Analysis is demanded. It contains atomic orbital overlap integrals generated by an external electronic structure code. The file is organized as follow:

- line 1**: Line of comments
- line 2 - line $2+n^2$** : $i j \text{overlap_integral}(i,j)$, i and j both running on all orbitals.

3.4 atoms.dat input file

This file is only used if computation of Mulliken Population Analysis is demanded. List of atoms and of atomic orbitals belonging to these atoms that are contributing to molecular orbitals activated by DMRG code are provided. The file is organized as follow:

- line 1**: Line of comments
- line 2**: Number of atomic orbitals for the first atom.
- line 3**: Atomic orbitals belonging to the first atom
- line 4**: Line of comments
- line 5**: Number of atomic orbitals for the second atom.
- line 6**: Atomic orbitals belonging to the second atom

And so on.

3.5 cmohf.dat input file

This file is only used if computation of Mulliken Population Analysis is demanded. It contains the molecular orbital coefficients on atomic orbitals. The file is organized as follow:

–**line 1:** Line of comments

–**line 2 - end:** μ j $b_{\mu j}$ μ is the atomic orbital index, j the molecular orbital index

4 Output file

The code provides an output file, `dmrg.out` that allows the user to check inputs and to follow the computation. The print option gives much more infos but the file produced is then huge. DMRG energy and Mulliken Population Analysis are given, respectively, in `DIAGONALIZATION OF SUPERHAM` and `COMPUTATION OF MULLIKEN CHARGES` parts of the output file, they are written at each step of each sweep. Since for a DMRG computation there is no rigorous way to define a converged state, the user has to stop the job when he assesses the computation is converged enough. Energy for each step is given in `DIAGONALIZATION OF SUPERHAM` part.

5 About the code itself

The following table gives the files composing the DMRG code.

File name	
<code>davidson.f</code>	Subroutines for davidson diagonalization
<code>matrix.lib.f</code>	Subroutines to deal with large sparse matrix using minimum amount of memory
<code>mylapack.lib.f</code>	All necessary LAPACK subroutine [2]
<code>mrgrnk.lib.f</code>	Subroutine for Merge-sort ranking of an array
<code>rcm.f90</code>	Subroutines for RCM algorithm [4]
<code>dmrg_qc.f</code>	Main and DMRG specific routines

Some details about subroutines in `dmrg_qc.f`:

- `ReadVariables` : Read input files.
- `CheckErrors` : Input validity checked.
- `ORB_ORDERING` : Order orbitals as chosen by user. Integrals transformed from chemist notation on spatial orbitals to physicist notation on spin-orbitals.

- `InitializeBlk` : Construction of elementary blocks, Hamiltonian, states and operators.
- `InitializeEnv` : Construction of environment block, Hamiltonian, states and operators.
- `BuildEnl` : Calling `BuildEnlargedBlock` to enlarge blocks A and E.
- `Projectors` : Select states with proper spin and number of electrons to build super-Hamiltonian on needed restricted space.
- `BuildSuperHam` : Build super-Hamiltonian.
- `TraceGroundState` : Build projected density matrix $D_{A'}$.
- `DiagRho` : Diagonalize $D_{A'}$.
- `UpdateOperators` : Projection of block A' operators and Hamiltonian on the renormalized space.
- `AddRandom` (if `noise` \neq 0): Add noise to the elements of $D_{A'}$.
- `Compute_RDM1` (if MPA): Compute 1-particle reduced density matrix.
- `MPA` (if MPA): Compute Mulliken charges.

A Generate inputs with the Dalton electronic structure code

To generate needed integrals, Dalton electronic structure code can be used with the following modification:

- In Dalton, in `abacus/herdrv.F`, replace `HERINP` subroutine by the one provided in `DALTON/herinp_new.F`: to read the modified input file.
- In Dalton, in `include`, replace `print_twoint.h` file by the one provided in `DALTON`
- In Dalton, in `sirius/sirtra.F`, replace `TRAAB` subroutine by the one provided in `DALTON/traab_new.F`: to print the two-electron integral.
- In Dalton, in `sirius/sirdiis.F`, replace `DISCTL` subroutine by the one provided in `DALTON/disctl_new.F`: to print the one-electron integrals.

Dalton input file `input.dal` must contain the following keywords:

****INTEGRALS** part:

```
.TWOELC keyword,  
.true. keyword to print integrals  
nsym number of symmetries  
norb number of orbitals concerned by printing  
orb(:) list of orbitals
```

***TRANSFORMATION** part:

```
.FINAL  
10  
.LEVEL  
10
```

For more informations about Dalton input files see [1].

B To not modify input files

B.1 model.in file

```
aic_ai  
ai+1c_ai+1  
aic_ai+1  
aic_ai+1c_ai_ai+1  
aic  
ai+1c  
aic_ai+1c  
aic_ai_ai+1  
ai+1c_ai+1_ai
```

B.2 aic_ai.dat file

```
0.d0 0.d0 0.d0 0.d0  
0.d0 1.d0 0.d0 0.d0  
0.d0 0.d0 0.d0 0.d0  
0.d0 0.d0 0.d0 1.d0
```

B.3 ai+1c_ai+1.dat file

0.d0 0.d0 0.d0 0.d0
0.d0 0.d0 0.d0 0.d0
0.d0 0.d0 1.d0 0.d0
0.d0 0.d0 0.d0 1.d0

B.4 aic_ai+1.dat file

0.d0 0.d0 0.d0 0.d0
0.d0 0.d0 1.d0 0.d0
0.d0 0.d0 0.d0 0.d0
0.d0 0.d0 0.d0 0.d0

B.5 aic_ai+1c_ai_ai+1.dat file

0.d0 0.d0 0.d0 0.d0
0.d0 0.d0 0.d0 0.d0
0.d0 0.d0 0.d0 0.d0
0.d0 0.d0 0.d0 -1.d0

B.6 aic.dat file

0.d0 0.d0 0.d0 0.d0
1.d0 0.d0 0.d0 0.d0
0.d0 0.d0 0.d0 0.d0
0.d0 0.d0 1.d0 0.d0

B.7 ai+1c.dat file

0.d0 0.d0 0.d0 0.d0
0.d0 0.d0 0.d0 0.d0
1.d0 0.d0 0.d0 0.d0
0.d0 -1.d0 0.d0 0.d0

B.8 aic_ai+1c.dat file

0.d0 0.d0 0.d0 0.d0
0.d0 0.d0 0.d0 0.d0
0.d0 0.d0 0.d0 0.d0

```
1.d0  0.d0  0.d0  0.d0
```

B.9 aic_ai_ai+1.dat file

```
0.d0  0.d0  0.d0  0.d0  
0.d0  0.d0  0.d0 -1.d0  
0.d0  0.d0  0.d0  0.d0  
0.d0  0.d0  0.d0  0.d0
```

B.10 ai+1c_ai+1_ai.dat file

```
0.d0  0.d0  0.d0  0.d0  
0.d0  0.d0  0.d0  0.d0  
0.d0  0.d0  0.d0  1.d0  
0.d0  0.d0  0.d0  0.d0
```

C Example for system specific files

An example of `dmrg.in`, `integral.dat`, `atoms.dat`, `overlap.dat` and `cmohf.dat` inputs files are presented on 1A_1 Trimethyl methane (TMM) system. In this example, to get integral inputs, a Hartree Fock computation is ran with Dalton using cc-pVDZ basis set. π orbitals are selected to be activated by DMRG algorithm, σ orbitals are frozen. Dalton input files are given in section C.1. To be activated Hartree-Fock π Molecular Orbitals are: 14 15 16 19 29 34 35 37 43 45 48 50 51 56 61 62 66 67 68 72 74 75. Thus 22 orbitals, 44 spin-orbitals, are activated. The code will reordered the orbitals according to user choice (see 3.1.2). In table 2, the DMRG spin-orbital ordering resulting from E_HF_1 and E_HF_2 is given. Section C.2.1 shows the `integral.dat` file generated by dalton, and Section C.3 the `dmrg.in` file.

C.1 Dalton input files

C.1.1 TMM.mol

```
BASIS  
cc-pVDZ  
tmm_1a1_mcscf  
=====  
Atomtypes=2  Charge=0  Nosymmetry  Angstrom  
Charge=6.0  Atoms=4
```

DMRG SO HF MO		DMRG SO HF MO	
E_HF_1		E_HF_2	
1	14	1	14
2	15	2	14
3	16	3	15
4	19	4	15
5	29	5	16
6	34	6	16
7	35	7	19
8	37	8	19
9	43	9	29
10	45	10	29
11	48	11	34
12	50	12	34
13	51	13	35
14	56	14	35
15	61	15	37
16	62	16	37
17	66	17	43
18	67	18	43
19	68	19	45
20	72	20	45
21	74	21	48
22	75	22	48
23	14	23	50
24	15	24	50
25	16	25	51
26	19	26	51
27	29	27	56
28	34	28	56
29	35	29	61
30	37	30	61
31	43	31	62
32	45	32	62
33	48	33	66
34	50	34	66
35	51	35	67
36	56	36	67
37	61	37	68
38	62	38	68
39	66	39	72
40	67	40	72
41	68	41	74
42	72	42	74
43	74	43	75
44	75	44	75

Table 2: Examples of ordering on 1A_1 TMM system

C1	1.438000	0.000000	0.000000
C2	0.068000	0.000000	0.000000
C3	-0.704734	-1.280976	0.000000
C4	-0.704734	1.280976	0.000000
Charge=1.0 Atoms=6			
H5	-0.186092	-2.228293	0.000000
H6	-0.186092	2.228293	0.000000
H7	-1.784727	-1.277206	0.000000
H8	-1.784727	1.277205	0.000000
H9	1.998505	-0.925504	0.000000
H10	1.998505	0.925504	0.000000

C.1.2 TMM.dal

```

**DALTON INPUT
.RUN WAVE FUNCTIONS
**INTEGRALS
.TWOELC
.true.
1
30
14 15 16 19 29 34 35 37 43 45 48 50 51 56 61 62 66 67 68 72 74 75
**WAVE FUNCTION
.HF
*SCF INPUT
.MAX DIIS IT
20
*TRANSFORMATION
.FINAL
10
.LEVEL
10
*END OF INPUT

```

C.2 DMRG input files

C.2.1 integral.dat file

```
-151.72428747 !E_frozen+E_nuclear
```

ONE ELECTRON MOLECULAR INTEGRALS T_IJ

14	14	-9.97875932291710
14	15	-1.252555212793906E-007
14	16	7.222508335382696E-003
14	19	1.01690642048924
15	14	-1.252555218208412E-007
15	15	-8.89739962386853
15	16	3.308455531021368E-008
75	74	1.215377894020486E-007
75	75	-6.89184686987904

TWO ELECTRONS MOLECULAR INTEGRALS V_IJKL

14	14	14	14	0.381732986830152
14	14	15	14	3.877814149975088E-009
14	14	15	15	0.282813745242519
14	14	16	14	-2.193811998430499E-002
14	14	16	15	-1.829724877363550E-008
14	14	16	16	0.315370557837740
14	14	19	14	-6.938670124494889E-002
75	75	75	74	7.451006109847336E-009
75	75	75	75	0.314733749332523

V_UNACT_IJ INTEGRALS

14	14	8.64481559154142
14	15	8.831309886271962E-008
14	16	-4.638882084447826E-002

75	74	-1.065344080914602E-007
75	75	8.06025678646336

C.3 dmrg.in file for singlet state

```
.false. !lprint
.true. !lDBSS
E_HF_1 !check_order
.true. !ldavidson
1.d-15 0.000010 20 !rho_cut err_min,emme_min
0.d0 !noise
44 !ell1
4 !d
-----
10 !nsweep
9 !NumHamLc
-----
0 !2.spin_tot
4 !ne_tot
.true. !locc_auto
.false. !lmpa
-----
/home/DMRG/ !dirname
INPUTS !FolderIn
model.in !filemodel
integral.dat !fileintegral
-----
```

C.4 dmrg.in file for triplet state

```
.false. !lprint
.true. !lDBSS
E_HF_1 !check_order
.true. !ldavidson
1.d-15 0.000010 20 !rho_cut err_min,emme_min
0.d0 !noise
44 !ell1
```

```

4 !d
-----
10 !nsweep
9 !NumHamLc
-----
1 !2.spin_tot
4 !ne_tot
.false. !locc_auto
1 2 0 !n_doubly, n_simply_up, n_simply_down
1 !occ_doubly
2 3 !occ_simply_up
0 !occ_simply_down
.false. !lmpa
-----
/home/DMRG/ !dirname
INPUTS !FolderIn
model.in !filemodel
integral.dat !fileintegral
-----

```

C.5 dmrg.in file for singlet state with Mulliken population analysis

```

.false. !lprint
.true. !lDBSS
E_HF_1 !check_order
.true. !ldavidson
1.d-15 0.000010 20 !rho_cut err_min,emme_min
0.d0 !noise
44 !ell1
4 !d
-----
10 !nsweep
9 !NumHamLc
-----
0 !2.spin_tot
4 !ne_tot

```



```
.true. !locc_auto
.true. !lmpa
10 86 !natom,nbasis
-----
/home/DMRG/ !dirname
INPUTS !FolderIn
model.in !filemodel
integral.dat !fileintegral
atoms.dat !fileatoms
overlap.dat !fileoverlap
cmohf.dat !filecmohf
-----
```

C.6 atoms.dat file for singlet state with Mulliken population analysis

```
!C1
1 !Number of atomic orbitals on C1 atom for DMRG active space
1 !List of atomic orbitals
!C2
1
2
!C3
1
3
!C4
1
4
!H5
0
0
!H6
0
0
!H7
0
0
```

!H8
0
0

C.7 overlap.dat file for singlet state with Mulliken population analysis

OVERLAP INTEGRALS

1	1	0.99999999999999989
1	2	4.10320965567396459E-008
1	3	2.45095759781535430E-002
1	4	6.59171123497869762E-004

C.8 cmohf.dat file for singlet state with Mulliken population analysis

CMOHF

6	14	0.422392
6	15	0.000000
6	16	-0.580231
6	19	-0.249141
6	29	1.397619
6	34	-0.569579
6	35	-0.000001
6	37	-0.293284
6	43	0.000000
6	45	-0.202442
6	48	0.000000
6	50	-0.000004
6	51	-0.212626
6	56	-0.173837
6	61	0.000001
6	62	-0.428906
6	66	0.082673

6 67 0.000000
6 68 -0.028028
6 72 0.000000
6 74 -0.000001
6 75 0.258019
9 14 0.015600

|
|
|

References

- [1] dalton2011, a molecular electronic structure program (2011), <http://www.daltonprogram.org>.
- [2] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition, 1999.
- [3] N. O. Attila Szabo. *Modern quantum chemistry*, volume section 2.4. Dover publication, INC, 1996.
- [4] J. Burkardt, A. George, and J. Liu. Reverse cuthill mckee ordering http://people.sc.fsu.edu/~jburkardt/f_src/rcm/rcm.html.
- [5] G. K.-L. Chan and M. Head-Gordon. Highly correlated calculations with a polynomial cost algorithm: A study of the density matrix renormalization group. *J. Chem. Phys.*, 116:4462, 2001.
- [6] O. Legeza and G. Fath. Accuracy of the density-matrix renormalization-group method. *Phys. Rev. B*, 53(21):14349–14358, JUN 1 1996.
- [7] O. Legeza, J. Roder, and B. A. Hess. Controlling the accuracy of the dmrg method. the dynamical block state selection approach. *Phys. Rev. B*, 67:125114, 2003.
- [8] A. Mitrushenkov, R. Linguerri, P. Palmieri, and G. Fano. Quantum chemistry using the density matrix renormalization group II. *J. Chem. Phys.*, 119(8):4148–4158, AUG 22 2003.

- [9] G. Moritz. *On a new solution to the electron correlation problem in quantum chemistry: The density Matrix Renormalization Group algorithm*. PhD thesis, ETH Zurich, 2007.
- [10] G. Moritz, B. A. Hess, and M. Reiher. Convergence behavior of the density-matrix renormalization group algorithm for optimized orbital orderings. *J. Chem. Phys.*, 122:024107, 2005.
- [11] G. Moritz and M. Reiher. Decomposition of density matrix renormalization group states into a slater determinant basis. *J. Chem. Phys.*, 126:244109, 2007.
- [12] A. Stathopoulos and C. Fischer. A davidson program for finding a few selected extreme eigenpairs of a large, sparse, real, symmetric matrix. *Computer Physics Communications Program Library*, 79, 1994.
- [13] S. R. White and R. L. Martin. Ab initio quantum chemistry using the density matrix renormalization group. *J. Chem. Phys.*, 110:4127, 1999.