University of L’Aquila

Department of Information Engineering, Computer Science and Mathematics

Master’s Degree in

Mathematical Engineering

Master Thesis

An algorithm to calculate the ground state energy and force of multi-center transition metal complexes

Advisors:

Prof. Leonardo Guidoni
Dr. Daniele Bovi

Student:

Henry Martin

Matricola
228413

2014/2015

Contents

List of Figures

NOTATION

Symbols/terms used throughout the this work are defined as follows.

ab	Summation nuclei indices
B	Overlap Integral
χ_α	Nature of Basis function atomic orbital
$c_{\alpha i}$	Molecular Orbital expansion coefficients
C	Coulomb Integral
CG_{ij}	Clebsch-Gordan Coefficient of Spin Eigenstate
δ_{ij}	Kronecker delta ($\delta_{ij} = 1\ for\ i = j,\ \delta_{ij} = 0\ for\ i \neq j$)
Δ	A finite difference of a quantity
d_i	Coefficient from basis set
E	Energy, many particles or terms
ϕ	Molecular orbital
Φ	Slater determinant or similar approximate wave function
F	Fock operator or Fock matrix
$\hbar$	$h/2\pi,\ h - Plancks\ constant$
h_i	Kinetic and Potential energy of electrons moving in orbit
H, He	Hamiltonian operator or Hamiltonian matrix (general, electrons)
H_{ij}	Matrix element of a Hamiltonian operator
ijl	Indices representing number of spins
J_{ij}, J_{ex}	Exchange (Coupling Constant, Integral)
K_{ij}	Exchange Integral
k	Index representing number of Broken Symmetry (BS) calculation perform
m	Basis set
M	Total basis set
M_s	Broken Symmetry (BS) calculated Spin Momenta
Me	Metal of molecular system
n	Eigenstate
$n(r)$	Electron density
N	Total number of electrons
N_t	Total valence electrons of complexes
p	pair of indices ij
q	Total charge of the metal complexes
r,R	Position vector (electron, nuclear)
s	Electron spin operator
S	Total spin operator
t	Temporal coordinates of the electron
$u([n], r)$	Effective potential

$v(r)$	External potential
V_{nn}	Potential (Coulomb) energy operator nuclear - nuclear
ΔE	Energy difference of Spin Ladder
Ψ, Ψ_{tot}	Wave function (general, total)
$\mid n\rangle$	Ket, referring to a function characterized by quantum number n
$\langle n\mid$	Bra, referring to a function characterized by quantum number n
$\langle n\mid H\mid m\rangle$	Bracket (matrix element) of operator H between functions n and m
$*$	Complex conjugate

Abstract

An algorithm for calculating the ground state (low-spin) energy and force of a multi-center transition metal (MCTM) complexes using only single determinant broken symmetry states is formulated using Heisenberg-Dirac-van Vleck formulation of spin Hamiltonian. We considered several broken symmetry states of the Schrödinger equation using the Density Functional Theory. Then, we estimated the magnetic coupling constants, J, using Ising Hamiltonian with all the calculated DFT energies and spin momenta. Afterwards, using the Heisenberg-Dirac-van Vleck formulation of spin Hamiltonian we calculated the full spin ladder energy spectra by evaluating the appropriate Clebsch-Gordan coefficients. Once the lowest energy state with the correct spin symmetry is identified we can also calculate the forces acting on the nuclei by linear combination of broken symmetry forces. All the algorithms were implemented numerically by a FORTRAN code, given us the opportunity to calculate the energy and force of the ground state (low-spin) to properly describe the magnetic, electronic and geometrical properties of MCTM materials. In this present work we conducted an application for the tetra-nuclear iron-sulphur cluster (cubane) $[Fe_4S_4(SCH_3)_4]^{-1}$.

INTRODUCTION

The multi-center transition metal (MCTM) complexes play an important role in the day to day activities of our lives in areas such as biology and material science. MCTM are present as catalyst and redox centers in many metalloenzymes and electron-transport proteins such as ferredoxin (Fd) [13], hydrogenases, copper-based enzymes and photosynthetic complexes [16].

The theoretical calculations are valuable for evaluating geometrical and electronic structures. Despite the wide application, density functional theory calculations suffer from some limitations when applied to multi-center TM complexes. This is because, it cannot fully describe their electron state due to the average field of the Coulomb repulsion term between the electrons. This results in an overestimated calculation ("self-interaction error") that makes the orbitals localization more difficult [3]. In addition these functionals may also suffer from static correlation errors that discourage the formation of open-shell electron states usually possible by the high degeneracy of the d-orbitals. Other approaches, such as the Hartree-Fock (HF) approach suffers from the opposite drawback, preferring parallel spin configurations and therefore leading to an excessive disposition for localized high-spin (HS) state [5]. In both cases the results might provide an erroneous description of the magnetic and geometrical properties of the complexes [12]. The second problem affecting Kohn-Sham (KS)-DFT calculations on multi-center TM complexes is represented by the fact that the Kohn-Sham scheme is intrinsically a one-determinant theory which is not suitable to properly describe open-shell spin multiplets, which are often found at ground state of these complexes [12].

In the multi-center transition metal (MCTM) complexes the correct representation of the spin state is beyond the one-determinant picture of DFT because of the missing appropriate symmetrization of the spinor part. An approach to access the (LS) ground state with the correct spin symmetry is represented by the so-called extended broken-symmetry (EBS) method proposed by Marx and co-workers [11]. The approach is based on the reconstruction of low spin (LS) energy surface through an Heisenberg like Hamiltonian depending on the exchange coupling parameter J. The parameter can be estimated by performing two independent single determinant high spin (HS) and broken-symmetry (BS) calculations [11][22].

In the present work we investigate the capability of EBS in combination with hybrid functionals to describe the magnetic, electronic, geometry properties as well as determining the ground state energy and force of a tetra-nuclear iron-sulphur cluster (cubane) $[Fe_4S_4(SCH_3)_4]^{-1}$.

1 Transition Metal Complexes

The summary of the main concepts related to transition metal complexes is done here, for understanding the theory and the calculations presented in the next chapters.

Transition metal complexes (TMs) are molecular systems containing transition metal atoms. They can be composed by one single metal ion as shown in figure 1.1 or by two or more metal centres as shown in figure 1.2. These metal atoms are surrounded by molecules or ions called ligands.

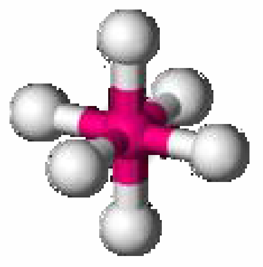


Figure 1.1: An Octahedral arrangement of TMs

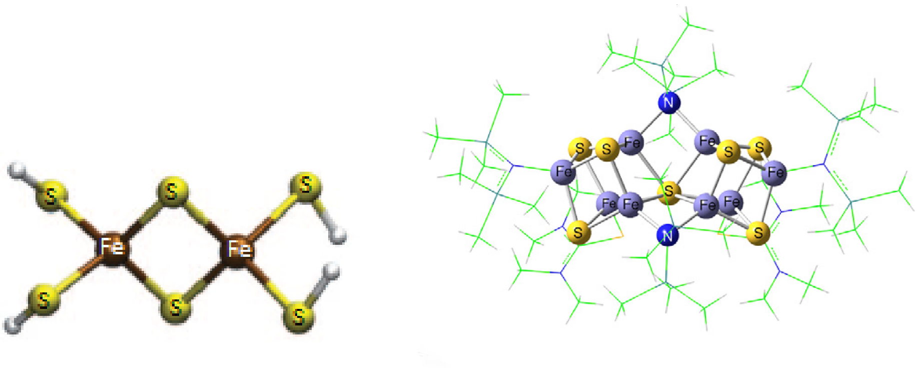


Figure 1.2: Multi-center TMs

1.1 OXIDATION STATES OF TRANSITION METALS

The electronic and geometrical structure are mutually correlated and provide us the basis for interpreting the thermodynamic, kinetic and spectroscopic data. The number of ligands attached to the TMs is called coordination number. Which depends on the electronic configuration (i.e. the number of empty d-orbitals) of the TMs.
The ligands can coordinate with a metal, Me, as [1]:

- Sharing the pair of electrons by doublet or by a π bond. This ligand is of type L_l.
- Sharing a single electron or more generally x single electron. This ligand is of type X_x.
- Sharing both the pairs of electrons and x electrons individually. This is ligand is of type $L_l X_x$.

The metal complexes can be indicated with the notation $[ML_l X_x]^q$.
where q is the total charge of the metal complexes.

The ligands therefore share the metal complexes as a total number of electrons equal to $\boldsymbol{2l+x}$. The core electrons of the transition metals are reasonably assumed to play a minor role as compared to the valence electrons[2]. The electron configuration of the transition metals assume the form $nd^a(n+1)s^b$, where n is equal to 3, 4 or 5 representing the first, second or third series of transition respectively. The number of valence electrons of the metal is equal to $m = a + b$. Therefore, the total valence electrons of the complexes is

$$N_t = m + 2l + x - q \tag{1.1}$$

Now, to determine the oxidation number of the metal complexe, we begin to dissociate the ligands from the metal as

$$[ML_l X_x]^q \rightarrow lL + xX^- + M^{x+q} \tag{1.2}$$

The oxidation number of the metal is not defined as the remaining charge on the metal, M, after the dissociation, $no =: x + q$ but formally, the number of electrons that remains on the metal is $m - no$.

There are various models for describing the electronic structure of the TMs complexes, but the Crystal Field theory (CFT) seems the simplest among them. CFT

[1] From Atkins book Page 54, 60, 62 [4]

[2] From Yves Jean book Page 54-59 [15]

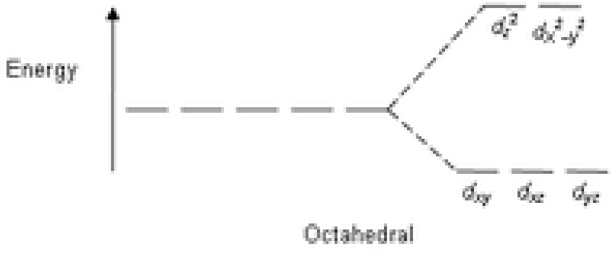


Figure 1.3: Octahedral Crystal field splitting of energy levels

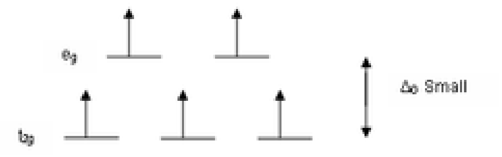


Figure 1.4: CFT High-Spin Splitting of $[FeBr_6]^{3-}$

shows the breaking of degeneracies of the electron orbital states (i.e. d-orbitals) due to a static electric field produced by the surrounding charge distribution (ligands) [6]. The interaction arises from the attraction between the positively charged metal, cation (TMs) and negative charge of the non-bonding electrons of the ligand. As a ligand approaches the metal ion, the electrons from the ligand will be closer to some of the d-orbitals and farther away from others causing a loss of degeneracy. This is possible due to the repulsion between the electrons of the d-orbitals and those of the ligand if their charges are like. Thus, the d-orbitals electrons closer to the ligands will have a higher energy than those further away, which results in the d-orbitals splitting in energy shown by figure 1.3. The size of the energy gap,Δ, between two or more sets of orbitals depends on several factors such as the ligands, the geometry of the complex and the oxidation state of TMs. The size of the splitting of the d-orbitals gives rise to a high spin or low spin state as shown by figure 1.4 and figure 1.5 respectively, base on whether the ligand is a weak or strong field.

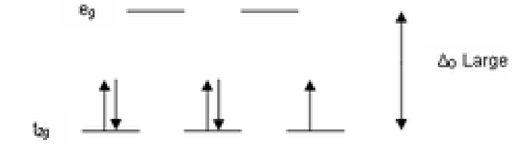


Figure 1.5: CFT Low-Spin Splitting of $[Fe(NO_2)_6]^{3-}$

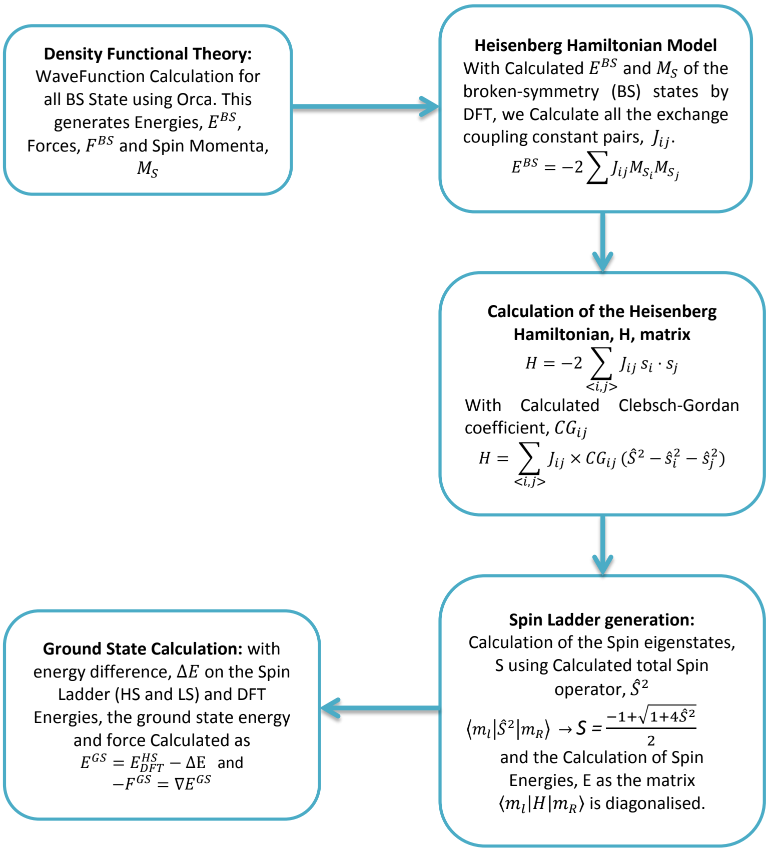
If the gap, Δ, is small (figure 1.4), the five orbitals are considered Degenerated and they are filled respectively. Also, figure 1.5 has half-filled d-orbital. The Hund's rule places unpaired electrons with parallel spin first, resulting in local high-spin configuration on the single metal center. Conversely, if the gap, Δ, is large the degeneracy is broken and the electrons orbitals are paired (figure 1.5) resulting in low spin configuration. These local spins may interact with each other either in ferromagnetic or in anti-ferromagnetic (or ferrimagnetic) way, leading to total high-spin (HS) or low-spin (LS), respectively. The LS state is often an open-shell spin singlet ($S = 0$) state, although formed by unpaired high-spin states on-sites[8].

1.2 Exchange Coupling Constant

The interaction of spins (parallel or anti-parallel) is known to give exchange energy. This interaction brings about ferromagnetism or anti-ferromagnetism. Since the spin state is beyond one determinant, due to the missing symmetric of the spin (parallel). One approach used to access the ground state (LS) with the correct spin symmetry is represented by extended broken-symmetry (EBS) method proposed by Marx and co-workers[11]. This approach elucidates the ground-state (LS) properties through the evaluation of the exchange coupling constant J using separated calculations performed on broken-symmetry (BS) and on high spin (HS) states. Therefore, it helps in calculating geometrical, energetic and magnetic properties such as magnetic susceptibility and magnetic moment of multi-nuclear TM complexes[19][20].

2 Electron Structure Theory

In this chapter all the mathematical models used are summarized. Figure 2.1 shows the diagram (flowchart) of the algorithm developed and computed.



Density Functional Theory: WaveFunction Calculation for all BS State using Orca. This generates Energies, E^{BS}, Forces, F^{BS} and Spin Momenta, M_S

Heisenberg Hamiltonian Model With Calculated E^{BS} and M_S of the broken-symmetry (BS) states by DFT, we Calculate all the exchange coupling constant pairs, J_{ij}.

$$E^{BS} = -2\sum J_{ij} M_{S_i} M_{S_j}$$

Calculation of the Heisenberg Hamiltonian, H, matrix

$$H = -2 \sum_{<i,j>} J_{ij}\, s_i \cdot s_j$$

With Calculated Clebsch-Gordan coefficient, CG_{ij}

$$H = \sum_{<i,j>} J_{ij} \times CG_{ij} \,(\hat{S}^2 - \hat{s}_i^2 - \hat{s}_j^2)$$

Spin Ladder generation: Calculation of the Spin eigenstates, S using Calculated total Spin operator, $\hat{S}^2$

$$\langle m_l|\hat{S}^2|m_R\rangle \rightarrow S = \frac{-1+\sqrt{1+4\hat{S}^2}}{2}$$

and the Calculation of Spin Energies, E as the matrix $\langle m_l|H|m_R\rangle$ is diagonalised.

Ground State Calculation: with energy difference, ΔE on the Spin Ladder (HS and LS) and DFT Energies, the ground state energy and force Calculated as $E^{GS} = E^{HS}_{DFT} - \Delta\mathrm{E}$ and $-F^{GS} = \nabla E^{GS}$

Figure 2.1: Flowchart to calculated the ground state energy of MCTM

2.1 Schrödinger Equation

Multi-center TMs complexes rely basically on the electronic structure, by means of the time-independent Schrödinger equation given below

$$H\Psi = E\Psi \tag{2.1}$$

where Ψ - wave function, H - Hamiltonian operator and E - Energy of the state.

This in effect means "When the Hamiltonian operator, H, acts on a certain wave function,Ψ, and the result is proportional to the same wave function,Ψ then, Ψ is a stationary state, and the proportionality constant, E, is the energy of the state Ψ.

These can be shown from the point of view that the wave function is not a function of time, therefore, it can be separated as a product of spatial and temporal parts given as

$$\Psi(r,t) = \psi(r)\tau(t) \tag{2.2}$$

where r is the spatial coordinates. Therefore, given a relevant number of particles in a relevant number of dimensions. Solving by separation of variables implies that the general solution of the time-dependent equation has the form[17]

$$\Psi(r,t) = \psi(r)\exp^{-iEt/\hbar} \tag{2.3}$$

Therefore, the energy operator Ê = $i\hbar\partial/\partial t$ can always be replaced by the energy eigenvalue, E, thus the time independent Schrödinger equation is an eigenvalue equation for the Hamiltonian operator[23]. These eigenvalue, E, is interpretable physically as the total energy of the system.

This can be calculated by introducing independent-particle models, where the motion of one electron is considered to be independent of the dynamics of all other electrons. Therefore, the interactions between the electrons are approximated by taking all interactions and finding the average (thus, the Hartree-Fock (HF) theory. This is done by multiplying both sides of eqn 2.1 with the complex conjugate of Ψ, Ψ^* and then integrate over all space:

$$E = \frac{\int_{\Re^3} \Psi^* H\Psi d\tau}{\int_{\Re^3} \Psi^*\Psi d\tau} \tag{2.4}$$

For a system composed of several atoms, the wave functions depend on the coordinates of all the particles of the system. Being R the coordinate of nuclei and r the coordinates of the electrons, we have a system with of total wave function $\Psi_{tot}(R,r)$ such that

$$H\Psi_{tot}(R,r) = E_{tot}\Psi_{tot}(R,r) \tag{2.5}$$

in which

$$H = \sum_{a=1}^{M} \frac{1}{2}\nabla_a + \sum_{i=1}^{N} \frac{1}{2}\nabla_i + \sum_{i=1}^{N}\sum_{a=1}^{M}\Big(-\frac{Z_a}{\mid R_a - r_i \mid}\Big) + \sum_{i=1}^{N}\sum_{j>1}^{N} \frac{1}{\mid r_i - r_j \mid} + \sum_{a=1}^{M}\sum_{b>a}^{M} \frac{Z_a Z_b}{\mid R_a - R_b \mid} \tag{2.6}$$

In eqn 2.6, the atomic units have been reabsorbed as constants which are used by the following notations:

- a and b represents the nuclei indices while i and j are for electron indices
- Z is the atomic number of the nucleus shown by the index
- N and M are the total number of electrons and nuclei respectively

We can rewrite eqn 2.6 in terms of the electron Hamiltonian

$$H = \sum_{a=1}^{M} \frac{1}{2}\nabla_a + H_e + \sum_{a=1}^{M}\sum_{b>a}^{M} \frac{Z_a Z_b}{\mid R_a - R_b \mid} \tag{2.7}$$

where

$$\begin{aligned} H_e = \sum_{i=1}^{N} \frac{1}{2}\nabla_i + \sum_{i=1}^{N}\sum_{a=1}^{M}\Big(-\frac{Z_a}{\mid R_a - r_i \mid}\Big) + \sum_{i=1}^{N}\sum_{j>1}^{N} \frac{1}{\mid r_i - r_j \mid} = \\ = \sum_{i=1}^{N} h_i + \sum_{i=1}^{N}\sum_{j>1}^{N} g_{ij} \end{aligned} \tag{2.8}$$

Note, from eqn 2.8 that H_e only depends on the nuclear positions but not on their momenta.
For a system of many particles, it is necessary to describe each wave function with different Schrödinger equation. Therefore, a system composed of more electrons, each i-th electron can be written as

$$H_e(R)\Psi_i(R,r) = E_i(R)\Psi_i(R,r) \tag{2.9}$$

where H_e is the Hamiltonian relative to electrons. Now we can consider two points on the wave function as:

- being the Hamiltonian self-adjoint operator ($\int \Psi_i^* H \Psi_j dr = \int \Psi_j H^* \Psi_i^* dr$) and introducing the bra - ket notation (see Appendix, A.1) ($\langle \Psi_i \mid H \mid \Psi_j \rangle = \langle \Psi_j \mid H \mid \Psi_i \rangle^*$) then the wave functions can be chosen orthonormal as:

$$\begin{aligned} \int \Psi_i^*(R,r)\Psi_j(R,r)dr = \delta_{ij} \leftrightarrow \langle \Psi_i \mid \Psi_j \rangle = \delta_{ij} \\ \delta_{ij} = 1,\ i = j \\ \delta_{ij} = 0,\ i \neq j \end{aligned} \tag{2.10}$$

- the total (exact) wave function of the total system can be written as an expansion, such that a complete set of electronic function has a coefficient Ψ_{ni} which depend on the positions of the nuclei. Then we can write:

$$\Psi_{tot}(R,r) = \sum_{i=1}^{\infty} \Psi_{ni}(R)\Psi_i(R,r) \tag{2.11}$$

Substituting eqn 2.11 and eqn 2.10 into eqn 2.5 and introducing the Dirac delta notation, we obtain

$$\nabla^2\Psi_{ni} + E_j\Psi_{ni} + \sum_{i=1}^{\infty}(2\langle\Psi_j \mid \nabla_n \mid \Psi_i\rangle(\nabla_n\Psi_{ni}) + \langle\Psi_j \mid \nabla_n^2 \mid \Psi_i\rangle) = E_{tot}\Psi_{nj} \tag{2.12}$$

Now, we introduce two important approximations:

- Adiabatic approximation: which the total wave function can be approximated using a single electronic surface (thus, all coupling elements in eqn 2.12 are neglected) which are the terms $i \neq j$ giving:

$$(\nabla_n^2 + E_j(R) + \langle\Psi_j(R) \mid \nabla_n^2 \mid \Psi_j(R)\rangle)\Psi_{ni}(R) = E_{tot}(R)\Psi_{nj}(R) \tag{2.13}$$

- Born-Oppenheimer approximation: the term $\langle\Psi_j \mid \nabla_n^2 \mid \Psi_j\rangle$, is said to be the correction diagonal and its negligible compared to the term $E_j(R)$, therefore, the correction angle can be neglected in eqn 2.13 to obtain[24]:

$$(\nabla_n^2 + E_j(R))\Psi_{nj}(R) = E_{tot}\Psi_{nj}(R) \tag{2.14}$$

Sometimes these two approximations listed above are considered as one hypothesis and called indiscriminately with one of the names. More concretely, these assumptions imply that the system will be considered to have wave functions of electrons over an adiabatic evolved potential energy surface (PES) generated by the nuclei; in other words, for each different arrangement the spatial nuclei in time, should be resolved and the Schrödinger equation for only electrons with configuration of the nuclei. The total energy, in addition should be calculated as the sum of a contribution linked to only the repulsion of the nuclei and another due to kinetic energy, potential and Coulomb interaction between different electrons and between electrons and nuclei: to change the nuclei location, the second contribution is obtained from the Schrödinger equation for the electrons, and its contribution related to the repulsion between the nuclei in the light of their new coordinates.

2.2 HARTREE-FOCK APPROXIMATION

At this point, the problem has been reduced to solve eqn 2.14 for electrons associated with a given geometry of nuclei: the problem, however, in spite of the approximations made, can be solved analytically only for a very small number of cases (thus, systems of single electron as the atomic or hydrogen ions $H_2^+ and He_2^+$). Apart from that, solutions must be sought approximately (numerically) and the standard by which one can do that is by the so called variational principle, which states that any approximate wave function has an energy above or equal to the exact energy (see Appendix A.2). The equality holds only if the wave function is the exact function. By constructing a trial wave function containing a number of parameters, we can generate the "best" trial function of the given form by minimizing the energy as a function of these parameters.

Normally the wave function that is considered is related to the entire molecule and is approximated using the Slater determinant (i.e. in terms of spin-orbitals). It is a function of the form:

$$\Phi = \frac{1}{\sqrt{N!}} \begin{vmatrix} \phi_1(1) & \phi_2(1) & ... & \phi_N(1) \\ \phi_1(2) & \phi_2(2) & ... & \phi_N(2) \\ ... & ... & ... & ... \\ \phi_1(N) & \phi_2(N) & ... & \phi_N(N) \end{vmatrix} \tag{2.15}$$

where the functions $\phi_i(n)$ represent the i-th atomic orbital spin (thus, the i-th possible combination of the four quantum numbers) and n is the nth electron in the system. The choice to represent molecular orbitals by single Slater determinant is linked to the fact that they meet some important properties that are required from the chemical point of view: However, the price of this approximation is that the repulsion between electrons is considered only as a mean value. Using eqn 2.4 we can calculate the energy of an orbital as:

$$E = \frac{\langle \Phi \mid H \mid \Phi \rangle}{\langle \Phi \mid \Phi \rangle} \tag{2.16}$$

By developing the Slater determinant eqn 2.15 in eqn 2.16, and replacing the Hamiltonian with eqn 2.8 the numerator takes the following form:

$$\begin{aligned} \langle \Phi \mid H \mid \Phi \rangle = \int ... \int dr_1 ... dr_N \Big\{ [\phi_1(1)\phi_2(2)...] \Big[\sum_{i=1}^{N} h_i + \sum_{i=1}^{N} \sum_{j>i}^{N} g_{ij} + V_{nn} \Big] [\phi_1(1)\phi_2(2)...] \Big\} \\ - \int ... \int dr_1 ... dr_N \Big\{ [\phi_2(1)\phi_1(2)...] \Big[\sum_{i=1}^{N} h_i + \sum_{i=1}^{N} \sum_{j>i}^{N} g_{ij} + V_{nn} \Big] [\phi_2(1)\phi_1(2)...] \Big\} + ... \end{aligned} \tag{2.17}$$

whereas the denominator is

$$\begin{aligned}\langle\Phi\mid\Phi\rangle = \int\ldots\int dr_1 \ldots dr_N\{[\phi_1(1)\phi_2(2)\ldots\phi_N(N) - \phi_2(1)\phi_1(2)\ldots\phi_N(N) + \ldots]\\ [\phi_1(1)\phi_2(2)\ldots\phi_N(N) - \phi_2(1)\phi_1(2)\ldots\phi_N(N) + \ldots]\}\\ = \int\ldots\int dr_1 \ldots dr_N\{[\phi_1(1)\phi_2(2)\ldots][\phi_1(1)\phi_2(2)\ldots]\}\\ -\int\ldots\int dr_1 \ldots dr_N\{[\phi_2(1)\phi_1(2)\ldots][\phi_2(1)\phi_1(2)\ldots]\} + \ldots\end{aligned} \tag{2.18}$$

but the spin orbitals are chosen orthonormal to each other (if they refer to the same particle), to give

$$\int dr_i d\phi_i(k) d\phi_j(k) = 0 \Rightarrow (\langle\phi_i(k)\mid\phi_j(k)\rangle = 0) \tag{2.19}$$

$$\int dr_i d\phi_i(i) d\phi_i(i) = 1 \Rightarrow (\langle\phi_i(i)\mid\phi_i(i)\rangle = 1) \tag{2.20}$$

hence eqn 2.18 implies

$$\begin{aligned}\langle\Phi\mid\Phi\rangle = \langle\phi_1(1)\phi_2(2)\phi_3(3)\ldots\mid\phi_1(1)\phi_2(2)\phi_3(3)\ldots\rangle+\\ -\langle\phi_1(1)\phi_2(2)\phi_3(3)\ldots\mid\phi_1(1)\phi_3(2)\phi_2(3)\ldots\rangle+\\ +\langle\phi_1(1)\phi_3(2)\phi_2(3)\ldots\mid\phi_1(1)\phi_3(2)\phi_2(3)\ldots\rangle - \ldots =\\ = \langle\phi_1(1)\phi_1(1)\rangle\langle\phi_2(2)\phi_2(2)\rangle\langle\phi_3(3)\phi_3(3)\rangle\ldots+\\ -\langle\phi_1(1)\phi_1(1)\rangle\langle\phi_2(2)\phi_3(2)\rangle\langle\phi_3(3)\phi_2(3)\rangle\ldots+\\ +\langle\phi_1(1)\phi_1(1)\rangle\langle\phi_3(2)\phi_3(2)\rangle\langle\phi_2(3)\phi_2(3)\rangle\ldots = N!\end{aligned} \tag{2.21}$$

but for normalized wave function $\langle\Phi\mid\Phi\rangle = 1$.

while eqn 2.17 becomes

$$\begin{aligned}\langle\Phi\mid H\mid\Phi\rangle = \left\langle\phi_1(1)\phi_2(2)\ldots\middle|\sum_{i=1}^{N} h_i\middle|\phi_1(1)\phi_2(2)\ldots\right\rangle+\\ +\left\langle\phi_1(1)\phi_2(2)\ldots\middle|\sum_{i=1}^{N}\sum_{j>i}^{N} g_{ij}\middle|\phi_1(1)\phi_2(2)\ldots\right\rangle+\\ +\left\langle\phi_1(1)\phi_2(2)\ldots\mid V_{nn}\mid\phi_1(1)\phi_2(2)\ldots\right\rangle =\\ = \sum_{i=1}^{N}\langle\phi_1(1)\phi_2(2)\ldots\mid h_i\mid\phi_1(1)\phi_2(2)\ldots\rangle+\\ +\sum_{i=1}^{N}\sum_{j>i}^{N}\langle\phi_1(1)\phi_2(2)\ldots\mid g_{ij}\mid\phi_1(1)\phi_2(2)\ldots\rangle + V_{nn}\langle\Phi\Phi\rangle\end{aligned} \tag{2.22}$$

The terms in the above equation can be rewritten as follows:

$$\begin{aligned}\sum_{i=1}^{N}\langle\phi_1(1)\phi_2(2)...\mid h_i\mid\phi_1(1)\phi_2(2)...\rangle =\\ =\langle\phi_1(1)\mid h_1\mid\phi_1(1)\rangle\langle\phi_2(2)\mid\phi_2(2)\rangle...\langle\phi_N(N)\mid\phi_N(N)\rangle+\\ +\langle\phi_1(1)\mid\phi_1(1)\rangle\langle\phi_2(2)\mid h_2\mid\phi_2(2)\rangle...\langle\phi_N(N)\mid\phi_N(N) =\\ =\sum_{i=1}^{N}\langle\phi_i\mid h_i\mid\phi_i\rangle\end{aligned}\tag{2.23}$$

$$\begin{aligned}\sum_{i=1}^{N}\sum_{j>i}^{N}\langle\phi_1(1)\phi_2(2)...\mid g_{ij}\mid\phi_1(1)\phi_2(2)...\rangle =\\ =\langle\phi_1(1)\phi_2(2)\mid g_{12}\mid\phi_1(1)\phi_2(2)\rangle...\langle\phi_N(N)\mid\phi_N(N)\rangle+\\ -\langle\phi_1(1)\phi_2(2)\mid g_{12}\mid\phi_2(1)\phi_1(2)\rangle...\langle\phi_N(N)\mid\phi_N(N)\rangle+\\ +\langle\phi_1(1)\phi_3(3)\mid g_{13}\mid\phi_1(1)\phi_3(3)\rangle...\langle\phi_N(N)\mid\phi_N(N)\rangle...+\\ -\langle\phi_1(1)\phi_3(3)\mid g_{13}\mid\phi_3(1)\phi_1(3)\rangle...\langle\phi_N(N)\mid\phi_N(N)\rangle... =\\ =\sum_{i=1}^{N}\sum_{j>i}^{N}\langle\phi_i(i)\phi_j(j)\mid g_{ij}\mid\phi_i(i)\phi_j(j)\rangle+\\ -\sum_{i=1}^{N}\sum_{j>i}^{N}\langle\phi_i(i)\phi_j(j)\mid g_{ij}\mid\phi_j(i)\phi_i(j)\rangle =\sum_{i=1}^{N}\sum_{j>i}^{N}(J_{nn}-K_{nn})\end{aligned}\tag{2.24}$$

In summary, substituting these results into eqn 2.16, the energy can be calculated as

$$E=\sum_{i=1}^{N}h_i+\sum_{i=1}^{N}\sum_{j>i}^{N}(J_{nn}-K_{nn})+V_{nn}\tag{2.25}$$

where

- The terms h_i represents the kinetic and potential energy of the electrons which move in orbit;
- The terms J_{ij} are called Coulomb integral and represents the energy generated by the repulsion between electron pairs;
- The terms K_{ij} are called exchange integrals and originate from the fact that the electrons with spin parallel movements are related to each other; therefore, these terms has no counterpart in classical physics.
- The term V_{nn} is the Coulomb repulsion between the nuclei.

At this point the problem becomes finding a set of atomic orbitals such that the energy obtained from eqn 2.25 is minimal, since from the principle of variation, the more the energy is low the more the wave functions approximation come close to the actual ones. Therefore, having to maintain the orthonormality condition between the orbitals (eqn 2.19 and eqn 2.20), it becomes necessary to find the minimum of the function bound which is used as the Lagrange multiplier method.

$$\delta\Big(E-\sum_{ij}^{N}\lambda_{ij}(\langle\phi_i\mid\phi_j\rangle-\delta_{ij})\Big)=0 \Rightarrow \delta E-\sum_{ij}^{N}\lambda_{ij}(\langle\delta\phi_i\mid\phi_j\rangle\langle\phi_i\mid\delta\phi_j\rangle-\delta_{ij})=0 \tag{2.26}$$

From eqn 2.25 the energy change is

$$\delta E=\sum_{i}^{N}\langle\delta\phi_i\mid h_i\mid\phi_i\rangle+\langle\phi_i\mid h_i\mid\delta\phi_i\rangle+\sum_{ij}^{N}(\langle\delta\phi_i\mid J_i-K_j\mid\phi_i\rangle+\langle\phi_i\mid J_i-K_j\mid\delta\phi_i\rangle) \tag{2.27}$$

and defines the so-called Fock operator as

$$F_i =: h_i+\sum_{j}^{N}(J_i-K_j) \tag{2.28}$$

then from eqn 2.26, we can have

$$F_i\phi_i=\sum_{j}^{N}\lambda_{ij}\phi_j \tag{2.29}$$

which can be diagonalized by transformation unitary to obtain:

$$F_i\phi'_i=\epsilon\phi'_i \tag{2.30}$$

where ϕ'_i are called canonical molecular orbitals or generally orbital SCF.

Equation 2.30 above is apparently the eigenvalue equation: in reality, the Fock operator depends on all the occupied molecular orbitals shown in eqn 2.29 then each orbital can be computed only by knowing all the others: Therefore, to solve the Hartree-Fock equation one requires an iterative method.

The first step that must be done, is to give an analytical shape to the molecular orbitals: what we normally do is to form a linear combination of some functions of a kind that is compatible in terms of physical properties with what is expected from the molecular orbitals and that leads to an expression which from the point of view, is computationally manageable within a reasonable time; to a good compromise between these two characteristics, the function is represented by

Gaussian functions, which are the type of function most used in this type of treatment. In general, one choose a set K functions of the same type, then its written as type

$$\phi_i' = \sum_{\alpha=1}^{K} c_{\alpha i}\chi_\alpha \tag{2.31}$$

Defining eqn 2.31, we must determine two things: K, number of functions and their nature χ_α.

As a general principle, whenever there is a function of linear combination of some kind, the greater the number of functions which are added, the higher the level of details obtained in representing the limit function. This also applies to the case of the representation of the canonical molecular orbitals, eqn 2.31: leading to infinity number of K functions in χ_α, resulting in the theory to give the exact function of ϕ_i' that we want. Obviously, in practice one cannot use bases set composed of an infinite number of elements, but it is clear that both the greater the number of basic functions used, the lower the degree of approximation of the canonical molecular orbital, ϕ_i', at the expense of the computational cost.

As regards to the choice of a type of basic function, in theory any form of χ_α chosen is okay, as long as the eqn 2.31 converge: we already know that, it is desirable to choose a type of basic function which from the point of computation, it is not particularly heavy to handle and simultaneously, replaced in eqn 2.31, which reproduces a quite good ϕ_i' without a need for high number of K terms.

2.3 DENSITY FUNCTIONAL THEORY

The basis for Density Functional Theory (DFT) is that the ground state electronic energy is determined completely by the electron density $n(r)$, which shows that the energy of a system is defined as a functional of the electron density. This makes the many-electron wavefunction $\Psi(r1\sigma1,...,rN\sigma N)$ contains a lot of information which are more than we usually want. Because it is a function of many variables, it is not easy to calculate and store. Mostly, we are after the total energy, E (and its changes), or perhaps also the spin densities $n\uparrow(r)$ and $n\downarrow(r)$, for the ground state. We can formally replace Ψ by the observables $n\uparrow$ and $n\downarrow$ as the basic variational objects. There are two (2) main density functionals mostly used, namely the local spin density (LSD) and generalized gradient approximation (GGA). The density functionals help to predict properties such the shapes and sizes of molecules, the crystal structures and the energy barriers of materials.

A function assigns a number, $f(x)$ to a number x, a functional is a rule which assigns a number $F[f]$ to a function, f. Given an example, $h[\Psi] = -\Psi|H|\Psi$is a functional of the trial wavefunction Ψ, given the Hamiltonian H . The local spin density approximation for the exchange energy is a functional of the density n(r). The functional derivative $\delta F/\delta n(r)$ tells us how the functional $F[n]$ changes under a small variation $\delta n(r)$.

Kohn-Sham Theory [3] : This mostly used for electronic structure calculation of materials under the DFT. Which mainly seeks the ground state energy and spin densities $n\uparrow(r)$, $n\downarrow(r)$ for a collection of N electrons interacting with one another, with an external potential v(r), the effective potential, u([n],r) which includes the classical Hartree-Fock and finally a spin-dependent exchange correlation potential,$v_{xc}([n\uparrow, n\downarrow], r)$ which is a functional of the spin densities. These are found by the self consistent solution of a one-electron Schrödinger equation.

[3] From the book A Primer in Density Functional Theory, Page 1-5 [9]

3 Heisenberg-Dirac-van Vleck Model

3.1 Two Spin System

The solution of the systems energies can be found by the eigenvalues in a symmetric or anti-symmetric spin as

$$E = E_0 + \frac{C \pm J_{ex}}{1 \pm B^2} \tag{3.1}$$

Where C is the Coulomb integral, B is the Overlap integral and J_{ex} is the exchange integral.
But then J. H. Van Vleck proposed a theory stating that *The potential energy of the interaction between the two electrons in orthogonal orbitals can be represented by a matrix, E as above; the characteristic values of this matrix are $C \pm J_{ex}$. The characteristic values of a matrix are its diagonal elements after it is converted to a diagonal matrix. Now, the characteristic values of the square of the magnitude of the resultant spin $(s_1 + s_2)^2$ is $S(S+1)$.*[26] then

$$E = C - \frac{1}{2}J_{ex} - 2J_{ex}(s_1) \bullet (s_2) \tag{3.2}$$

where s_1 and s_2 are the spin momenta.
Then, Dirac pointed out that *the critical features of the exchange interaction could be obtained in an elementary way by neglecting the first two terms on the right-hand side of the above equation, thereby considering the two electrons as simply having their spins coupled by a potential,V.*
Which follows that the exchange interaction Hamiltonian between two electrons in orbitals Φ_1 and Φ_2 can be written in terms of their spin momenta s_1 and s_2. This is the well-known Heisenberg-Dirac Hamiltonian[20].

$$H = -2J_{12}(s_1) \bullet (s_2) \tag{3.3}$$

where J_{12} is the exchange coupling constant. However, with orthogonal orbitals (thus, $B = 0$), for example with different orbitals in the same atom, $J_{12} = J_{ex}$.
Therefore, J_{12} is positive when the electrons are with parallel spins (ferromagnetism) and negative when electrons are anti-parallel spins (anti-ferromagnetism).
Therefore, the magnetic coupling between the two on-site high spin states in TM dimers can be described as phenomenological level by the Heisenberg-Dirac-van Vleck spin Hamiltonian [14][1][25]. Now, with the evaluation of the unpaired electron spins which are anti-ferromagnetic of the multi-center TMs complexes, we note that $s_i \bullet s_j$ do commute. Therefore, we introduction the total spin operator

$$\hat{S}^2 = (\hat{s}_1 + \hat{s}_2)^2 = \hat{s}_1^2 + \hat{s}_2^2 + 2\hat{s}_1\hat{s}_2 \tag{3.4}$$

Then

$$H = -J_{12}(\hat{S}^2 - \hat{s}_1^2 - \hat{s}_2^2) \tag{3.5}$$

But we also know that $\hat{s}_1^2 = s_1(s_1+1)$
Then the energy of a state of total spin, S, is estimated as an approximation [19][18][21] using the Clebsch-Gordan coefficient spin eigenstates employed for many-electron Hamiltonian.

$$H = -J_{12} \times CG_{12} \times [S(S+1) - s_1(s_1+1) - s_2(s_2+1)] \tag{3.6}$$

3.2 Generalized Case

In situation, when a particular eigenstate, n, has more than two spins then, the energy becomes:

$$E^n = \langle n \mid H \mid n \rangle \tag{3.7}$$

Where, n is given as:

$$n = s_1 m_1; s_2 m_2; s_3 m_3; ...; s_i m_i \tag{3.8}$$

thus the energy

$$E^n = \langle s_1 m_1; s_2 m_2; s_3 m_3; ...; s_i m_i \mid H \mid s_1 m_1; s_2 m_2; s_3 m_3; ...; s_i m_i \rangle \tag{3.9}$$

therefore, the Hamiltonian, H, can be decomposed by applying the two electron system technique as:

$$\begin{aligned} H &= H_{12} + H_{13} + ... + H_{ij} \\ &\qquad where \\ H_{12} &= -2J_{12}(s_1) \bullet (s_2) \\ &\vdots \\ H &= -2 \sum_{\langle i<j \rangle} J_{ij}(s_i) \bullet (s_j) \end{aligned} \tag{3.10}$$

hence, from the eqn 3.5 we can rewrite the above as

$$H = -\sum_{\langle i<j \rangle} J_{ij} \times CG_{ij} \times (\hat{S}_{tot}^2 - \hat{s}_i^2 - \hat{s}_j^2) \tag{3.11}$$

Where the summation of the Hamiltonian over the entire spin follows the orthogonality and orthonormal condition used in eqn 2.19 and eqn 2.20 as:

$$\sum_{i<j} \Big(\prod_{l \neq i \; l \neq j} \langle m_l^R \mid m_l^L \rangle \langle m_i m_j \mid -J_{ij} \times CG_{ij} \times (\hat{S}^2 - \hat{s}_i^2 - \hat{s}_j^2) \mid m_i m_j \rangle \Big) \tag{3.12}$$

Where m_l^R and m_l^L denotes all the right and left sided basis set excluded from the calculation of the Hamiltonian for each electron pair respectively. Therefore, for every

$$\begin{aligned} m_l^R = m_l^L \Rightarrow \langle m_l^R \mid m_l^L \rangle = 1 \\ else\ \langle m_l^R \mid m_l^L \rangle = 0 \end{aligned} \tag{3.13}$$

Also, the Clebsch-Gordan coefficient, CG, of spin eigenstates for many-electron Hamiltonian is of the form below (see appendix A.5 for formulation and code):

$$|s_i s_j; SM\rangle = \sum_{m_i} \sum_{m_j} |s_i m_i s_j m_j\rangle \langle s_i m_i s_j m_j | SM\rangle \tag{3.14}$$

Thus, the Hamiltonian generates Heisenbergs spin-ladder of eigenstates shown by figure 3.1 with total spins, S, in range $S \in [S_{min}, S_{max}]$, where $S_{min} = |s_i - s_j|$ (low- spin, LS) and $S_{max} = |s_i + s_j|$ (high-spin, HS).

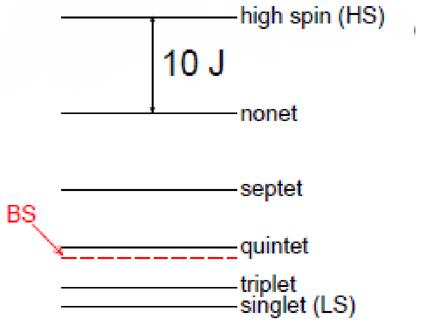


Figure 3.1: A spin-ladder generated by Heisenberg's spins Hamiltonian

3.3 Broken Symmetry State

In figure 3.1, we can notice that the broken symmetry state is an imaginary state, but one can formulate the kth energy using the same idea of the calculation in the above section as:

$$E_k^{BS} = \langle BS \mid H \mid BS \rangle_k \tag{3.15}$$

but, we generated the appropriate initial BS guess and convergence of the BS-DFT calculations by the "flipspin" feature implemented in ORCA. This relies on a high-spin wavefunction that is either readin after calculated as a first step, and exchanges the α and β blocks of the density on the specified centres (see section 4). Then, using the spin momenta,M, in the Ising Hamiltonian we can determine the exchange coupling constant, J, as:

$$H = -2 \sum_{i<j} J_{ij} M_{s_i}^k M_{s_j}^k \tag{3.16}$$

Then by considering the pairs (thus, the index $ij = p$), of the spin momenta we can rewrite the above as

$$H = -2 \sum_p^P J_p M_p^k \tag{3.17}$$

Then, its energy can be written as

$$\begin{pmatrix} E_1^{BS} \\ E_2^{BS} \\ E_3^{BS} \\ \vdots \\ E_k^{BS} \end{pmatrix} = -2 \begin{pmatrix} M_{12}^1 & M_{13}^1 & \cdots & M_{1N}^1 \\ M_{12}^2 & M_{13}^2 & \cdots & M_{1N}^2 \\ M_{12}^3 & M_{13}^3 & \cdots & M_{1N}^3 \\ \vdots & \vdots & \ddots & \vdots \\ M_{12}^k & M_{13}^k & \cdots & M_{1N}^k \end{pmatrix} \begin{pmatrix} J_1 \\ J_2 \\ \vdots \\ J_P \end{pmatrix} \tag{3.18}$$

which is simplified in a linear form as

$$E_k^{BS} = -2 J_p M_p^k \tag{3.19}$$

With the above equation, one can determine the exchange coupling constant, J, of the system by using singular value decomposition (SVD) (see appendix A.3)[27] as shown below since the kth broken symmetry energy can be found by using Density Functional Theory (DFT)[13].

$$J = -M^{-1} E_{DFT\ cal}^{BS} \tag{3.20}$$

3.4 Energy and Force on the Spin Ladder

From eqn 3.20, we noticed the possibility to access the spin ladder energy, E, (which eventually will give us the energies of all spin states in the Spin ladder shown in figure 3.1). Since J is now determined, then the basis set, m, can be determined as all spin combination (see appendix A.4)[11].

Now, we determine the spin ladder energy, E, and its corresponding states, n, by applying the generalized spin system introduced in the previous section to calculate the Hamiltonian matrix (see appendix A.7) then, we diagonalise it using additional libraries called LAPACK (see appendix A.6) after which we calculated the total spin matrix, $\hat{S}^2$ as:

$$\langle n \mid \hat{S}^2 \mid n \rangle \tag{3.21}$$

$$\langle s_1 m_1; s_2 m_2; s_3 m_3; ...; s_i m_i \mid \hat{S}^2 \mid s_1 m_1; s_2 m_2; s_3 m_3 ... s_i m_i \rangle \tag{3.22}$$

but $\hat{S}^2$ can be written below by using expansion and applying the Clebsch-Gordan Coefficient as:

$$\hat{S}^2 = (\sum_i \hat{S}_i)^2 = \sum_i \hat{S}_i^2 + 2\sum_{ij} \hat{s}_i \cdot \hat{s}_j = \sum_i S_i(S_i+1) + CG_{ij} \times \sum_{ij} (\hat{S}_{ij}^2 - \hat{s}_i^2 - \hat{s}_j^2) \tag{3.23}$$

Therefore, putting eqn 3.23 into eqn 3.22, we obtain:

$$\langle s_1 m_1; s_2 m_2; s_3 m_3; ...; s_i m_i \mid \sum_i S_i(S_i+1) + CG_{ij} \times \sum_{ij} (\hat{S}_{ij}^2 - \hat{s}_i^2 - \hat{s}_j^2) \mid s_1 m_1; s_2 m_2; s_3 m_3 ... s_i m_i \rangle \tag{3.24}$$

From eqn 3.24 above, the total spin matrix is calculated using the same process shown in eqn 3.12 - eqn 3.14 for the application of orthogonality and orthonormal conditions under the summation (see appendix A.8 for code). After, we then again use the libraries called LAPACK for the diagonalization then, we calculate the spins (see appendix A.9 for code) as:

$$S = \frac{-1 + \sqrt{1 + (4 \times \hat{S}^2)}}{2} \tag{3.25}$$

But, we know we can calculate the low spin state energy using the broken symmetry energy from DFT as:

$$E^{LS} = E^{HS}_{DFT\ cal} - \Delta E \tag{3.26}$$

then the force can also be calculated as

$$F^{LS} = -\nabla E^{LS} = -(-F^{HS}_{DFT\ cal} - \nabla(\Delta E)) \tag{3.27}$$

Therefore, one can notice from the spin ladder, the energy difference, ΔE between the high spin state, HS, and the low spin state(ground state), LS, is given as:

$$\Delta E = E^{HS} - E^{LS} \tag{3.28}$$

where the energy of the high spin states (HS) and the low spin state (LS) can be written as

$$E^{HS} = \langle \Psi^{HS} \mid H \mid \Psi^{HS} \rangle \; and \; E^{LS} = \langle \Psi^{LS} \mid H \mid \Psi^{LS} \rangle \tag{3.29}$$

but $\mid \Psi^{HS} \rangle$ and $\mid \Psi^{LS} \rangle$ being the low spin state can be given as

$$\mid \Psi^{HS} \rangle = \sum_i d_i^{HS} \mid m_i \rangle \; ; \mid \Psi^{LS} \rangle = \sum_i d_i^{LS} \mid m_i \rangle \tag{3.30}$$

where d_i^{LS} is the coefficient from the basis set, m_i. therefore, using eqn 3.29 and eqn 3.30 we can rewrite the energy difference, ΔE as

$$\begin{aligned}\Delta E = \sum_{ij} d_i^{HS} d_j^{HS} \langle m_i \mid H \mid m_j \rangle + \sum_{ij} d_i^{LS} d_j^{LS} \langle m_i \mid H \mid m_j \rangle = \sum_{ij} \left[d_i^{HS} d_j^{HS} H_{ij} - d_i^{LS} d_j^{LS} H_{ij} \right] = \\ = \sum_{ij} \left[(d_i^{HS} d_j^{HS} - d_i^{LS} d_j^{LS}) H_{ij} \right]\end{aligned} \tag{3.31}$$

then the gradient of the energy difference, $\nabla(\Delta E)$ is given as

$$\begin{aligned}\nabla(\Delta E) = \sum_{ij} \left[(d_i^{HS} d_j^{HS} - d_i^{LS} d_j^{LS}) \; \nabla(\langle m_i \mid H \mid m_j \rangle) \right] = \\ = \sum_{ij} \left[(d_i^{HS} d_j^{HS} - d_i^{LS} d_j^{LS})(\langle m_i \mid \; \nabla H \mid m_j \rangle) \right]\end{aligned} \tag{3.32}$$

knowing the Hamiltonian, H is

$$H = -\sum_p J_p (\hat{S}^2_{tot_p} - \hat{s}^2_{p1} - \hat{s}^2_{p2}) \tag{3.33}$$

the gradient of the Hamiltonian, ∇H is written

$$\nabla H = -\nabla \sum_p J_p (\hat{S}^2_{tot_p} - \hat{s}^2_{p1} - \hat{s}^2_{p2}) = -\sum_p C_p \nabla J_p \tag{3.34}$$

with a constant, $C_p = \hat{S}^2_{tot_p} - \hat{s}^2_{p1} - \hat{s}^2_{p2}$. From eqn 3.20 we can rewrite the above equation as

$$\nabla H = -\sum_p C_p \sum_k \frac{\partial J_p}{\partial E_k} \nabla E^{BS}_{DFT} \tag{3.35}$$

where $\frac{\partial J_p}{\partial E_k} = -A_{pk}$. Given $M^{-1} = A$

4 Application Results

In this chapter we show numerical results using the FORTRAN Code as while as ORCA by applying the developed algorithm in figure 2.1 on an iron-sulphur cluster $[Fe_4S_4(SCH_3)_4]^{-1}$. Also, comparison to some numerical results from other published papers with different approaches, but have similar results in some aspects of our work are shown.

4.1 Density Functional Theory Optimization

Density functional theory (DFT) geometry optimization for iron-sulphur cluster $(4Fe-4S)$ with oxidation state of -1 was performed. Iron-sulphur clusters in nature play an important role in electron transfer and catalysis in biological systems. These clusters are of a 'cubane' structure figure 4.1.
From electronic structure, the electron open-shell of $[Fe_4S_4(SCH_3)_4]^{-1}$ cluster is composed by one Fe^{2+} and three Fe^{3+}. The optimized structure were performed using ORCA with a total of one high-spin state and seven BS states determinants for $Fe^{3+}Fe^{2+}$ with different multiplicities using the unrestricted B3LYP density functional with the TZVP and TZV/J basis set as shown in table 1 below.

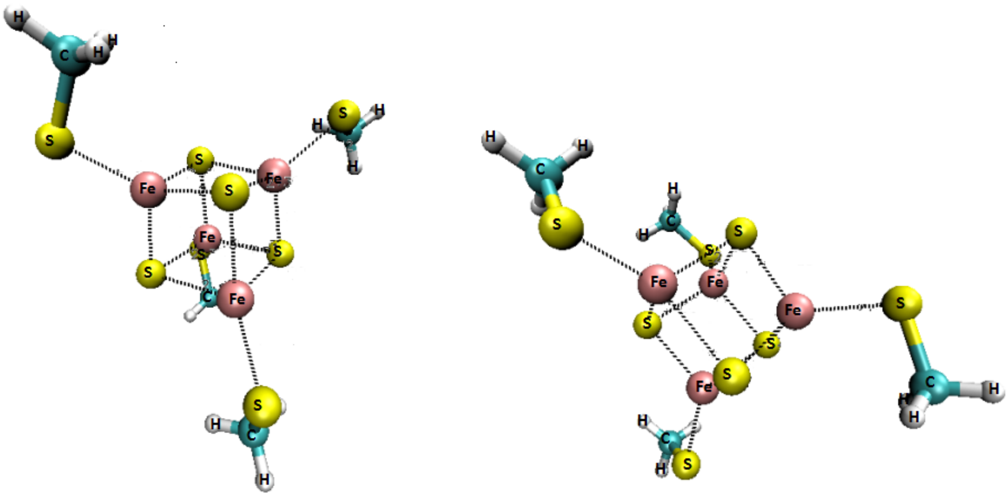


Figure 4.1: Iron-Sulphur Cluster $[Fe_4S_4(SCH_3)_4]^{-1}$ (Cubane)

Table 1: A description of all the possible spin-electron states of $[Fe_4S_4(SCH_3)_4]^{-1}$ with its local spin projection determinants.

	spin-electron states	$(2S+1)^*$	$\mid m_1, m_2, m_3, m_4\rangle$	M_S
HS: $\alpha\alpha\alpha\alpha$	Ψ_1	20	$\mid +5/2, +5/2, +5/2, +2\rangle$	19/2
BS: $\alpha\alpha\beta\beta$	Ψ_2	2	$\mid +5/2, +5/2, -5/2, -2\rangle$	1/2
BS: $\alpha\beta\alpha\beta$	Ψ_3	2	$\mid +5/2, -5/2, +5/2, -2\rangle$	1/2
BS: $\beta\alpha\alpha\beta$	Ψ_4	2	$\mid -5/2, +5/2, +5/2, -2\rangle$	1/2
BS: $\alpha\alpha\alpha\beta$	Ψ_5	12	$\mid +5/2, +5/2, +5/2, -2\rangle$	11/2
BS: $\alpha\alpha\beta\alpha$	Ψ_6	10	$\mid +5/2, +5/2, -5/2, +2\rangle$	9/2
BS: $\alpha\beta\alpha\alpha$	Ψ_7	10	$\mid +5/2, -5/2, +5/2, +2\rangle$	9/2
BS: $\beta\alpha\alpha\alpha$	Ψ_8	10	$\mid -5/2, +5/2, +5/2, +2\rangle$	9/2

*Multiplicities of spin-electron states.
M_S is the total local spin projection.

Table 2: Some important output features of all the possible spin-electron states of $[Fe_4S_4(SCH_3)_4]^{-1}$ from DFT Optimization.

	spin-electron states	$\langle S^2\rangle$	$S(S+1)$	Mulliken spin population
HS: $\alpha\alpha\alpha\alpha$	Ψ_1	99.8014	99.7500	19.0000
BS: $\alpha\alpha\beta\beta$	Ψ_2	9.4517	0.7500	1.0000
BS: $\alpha\beta\alpha\beta$	Ψ_3	9.4287	0.7500	1.0000
BS: $\beta\alpha\alpha\beta$	Ψ_4	9.4463	0.7500	1.0000
BS: $\alpha\alpha\alpha\beta$	Ψ_5	39.6508	35.7500	11.0000
BS: $\alpha\alpha\beta\alpha$	Ψ_6	29.4937	24.7500	9.0000
BS: $\alpha\beta\alpha\alpha$	Ψ_7	29.4852	24.7500	9.0000
BS: $\beta\alpha\alpha\alpha$	Ψ_8	29.4884	24.7500	9.0000

From table 2 above, the DFT optimization simulation shows that $\langle S^2\rangle$ is almost equal to $S(S+1)$ for the high-Spin (HS) state and greater for all the broken-symmetry (BS) states, which is expected.
Also, the calculated Mulliken spin population analysis from the DFT spin density converged in each spin state Ψ_k was in agreement with the ideal once.

4.2 Broken Symmetry DFT Energies and Exchange Coupling Constant

TABLE 3. Exchange coupling constant, J, Broken symmetry DFT Energies and the local spin projection (m_i) for $Fe^{3+}Fe^{2+}$ from Mulliken spin population using the extended broken symmetry approach.

J (cm^{-1})	BS^{DFT} Energies (Hartree)	$\vert m_1, m_2, m_3, m_4\rangle$
-102.4847	-8399.9227	$\vert 3.9593, 3.9579, 3.9529, 3.7917\rangle$
-87.9080	-8399.9403	$\vert 3.9287, 3.9385, -3.9013, -3.7451\rangle$
-45.1037	-8399.9425	$\vert 3.9230, -3.9116, 3.9228, -3.7374\rangle$
-92.6838	-8399.9409	$\vert -3.8992, 3.9320, 3.9294, -3.7453\rangle$
-27.1861	-8399.9307	$\vert 3.9458, 3.9565, 3.9445, -3.7064\rangle$
-44.0406	-8399.9385	$\vert 3.9300, 3.9407, -3.9007, 3.7655\rangle$
	-8399.9384	$\vert 3.9328, -3.9074, 3.9179, 3.7754\rangle$
	-8399.9392	$\vert -3.8986, 3.9338, 3.9314, 3.7643\rangle$

We can observe from table 3 above that there are NO positive exchange coupling constant, J, which favours electrons with parallel spins owing to be the main cause of ferromagnetism in material but ONLY negative exchange coupling constant, J, showing interaction favouring electrons with anti-parallel spins, causing anti-ferromagnetism in the biological system.

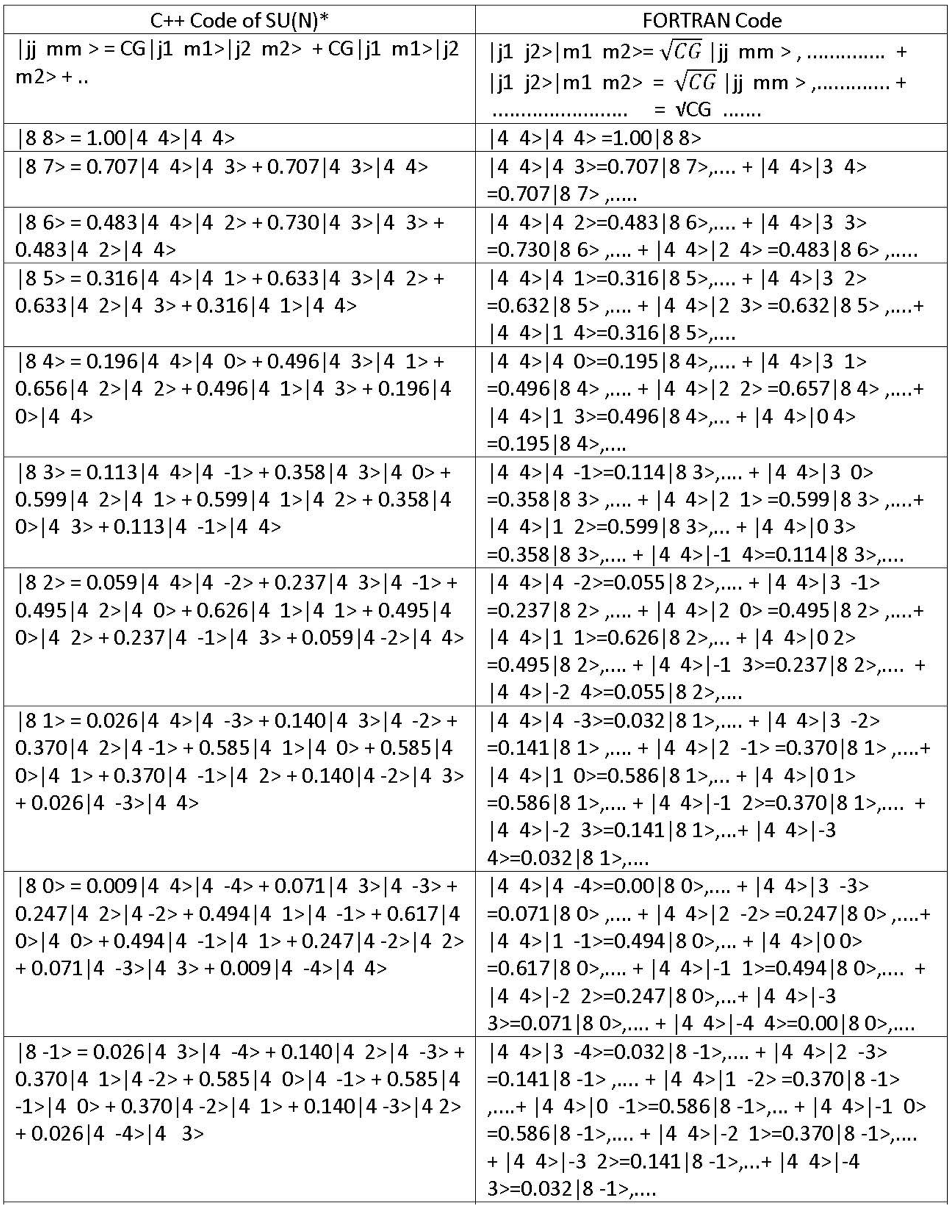
4.3 Clebsch-Gordan Coefficient of Spin

TABLE 1. A Comparison of the Clebsch-Gordan coefficient spin eigenstates employed for two (2) spin Hamiltonian

$J1, J2, m1, m2$	$\mid JM\rangle CGCoeff^1$	$\mid JM\rangle CGCoeff^2$
0.5 0.5 0.5 0.5	$\mid 1.0\ 1.0\rangle 1.000$	$\mid 1.0\ 1.0\rangle 1.000$
0.5 0.5 0.5 -0.5	$\mid 0.0\ 0.0\rangle 0.500$ $\mid 1.0\ 0.0\rangle 0.500$	$\mid 0.0\ 0.0\rangle 0.500$ $\mid 1.0\ 0.0\rangle 0.500$
0.5 0.5 -0.5 0.5	$\mid 0.0\ 0.0\rangle - 0.500$ $\mid 1.0\ 0.0\rangle 0.500$	$\mid 0.0\ 0.0\rangle - 0.500$ $\mid 1.0\ 0.0\rangle 0.500$
0.5 0.5 -0.5 -0.5	$\mid 1.0\ -1.0\rangle 1.000$	$\mid 1.0\ -1.0\rangle 1.000$
1.0 1.0 1.0 1.0	$\mid 2.0\ 2.0\rangle 1.000$	$\mid 2.0\ 2.0\rangle 1.000$
1.0 1.0 1.0 0.0	$\mid 1.0\ 1.0\rangle 0.500$ $\mid 2.0\ 1.0\rangle 0.500$	$\mid 1.0\ 1.0\rangle 0.500$ $\mid 2.0\ 1.0\rangle 0.500$
1.0 1.0 0.0 1.0	$\mid 1.0\ 1.0\rangle - 0.500$ $\mid 2.0\ 1.0\rangle 0.500$	$\mid 1.0\ 1.0\rangle - 0.500$ $\mid 2.0\ 1.0\rangle 0.500$
1.0 1.0 1.0 -1.0	$\mid 0.0\ 0.0\rangle 0.333$ $\mid 1.0\ 0.0\rangle 0.500$ $\mid 2.0\ 0.0\rangle 0.167$	$\mid 0.0\ 0.0\rangle 0.333$ $\mid 1.0\ 0.0\rangle 0.500$ $\mid 2.0\ 0.0\rangle 0.167$
1.0 1.0 0.0 0.0	$\mid 0.0\ 0.0\rangle - 0.333$ $\mid 1.0\ 0.0\rangle 0.000$ $\mid 2.0\ 0.0\rangle 0.667$	$\mid 0.0\ 0.0\rangle - 0.333$ $\mid 1.0\ 0.0\rangle 0.000$ $\mid 2.0\ 0.0\rangle 0.667$
1.0 1.0 -1.0 1.0	$\mid 0.0\ 0.0\rangle 0.333$ $\mid 1.0\ 0.0\rangle - 0.500$ $\mid 2.0\ 0.0\rangle 0.167$	$\mid 0.0\ 0.0\rangle 0.333$ $\mid 1.0\ 0.0\rangle - 0.500$ $\mid 2.0\ 0.0\rangle 0.167$
1.0 1.0 0.0 -1.0	$\mid 1.0\ -1.0\rangle 0.500$ $\mid 2.0\ -1.0\rangle 0.500$	$\mid 1.0\ -1.0\rangle 0.500$ $\mid 2.0\ -1.0\rangle 0.500$
1.0 1.0 -1.0 0.0	$\mid 1.0\ -1.0\rangle - 0.500$ $\mid 2.0\ -1.0\rangle 0.500$	$\mid 1.0\ -1.0\rangle - 0.500$ $\mid 2.0\ -1.0\rangle 0.500$
1.0 1.0 -1.0 -1.0	$\mid 2.0\ -2.0\rangle 1.000$	$\mid 2.0\ -2.0\rangle 1.000$
2.0 1.5 2.0 1.5	$\mid 3.5\ 3.5\rangle 1.000$	$\mid 3.5\ 3.5\rangle 1.000$
2.0 1.5 2.0 0.5	$\mid 2.5\ 2.5\rangle 0.571$ $\mid 3.5\ 2.5\rangle 0.429$	$\mid 2.5\ 2.5\rangle 0.571$ $\mid 3.5\ 2.5\rangle 0.429$

1. Our Fortran Clebsch-Gordan coefficient code
2. The coefficient here have been calculated using computer programs written independently by Cohen and at LBNL (Tables of the Clebsch-Gordan Coefficients, North American Rockwell Science Center, Thousand Oaks, Calif., 1974).

TABLE 2. A Comparison of the Clebsch-Gordan coefficient spin eigenstates employed for three (3) or more spin Hamiltonian



<table>
<tr><th>C++ Code of SU(N)*</th><th>FORTRAN Code</th></tr>
<tr><td>|jj mm > = CG|j1 m1>|j2 m2> + CG|j1 m1>|j2 m2> + ..</td><td>|j1 j2>|m1 m2>= $\sqrt{CG}$ |jj mm > , ... + |j1 j2>|m1 m2> = $\sqrt{CG}$ |jj mm > ,............ + ... = √CG ....</td></tr>
<tr><td>|8 8> = 1.00|4 4>|4 4></td><td>|4 4>|4 4> =1.00|8 8></td></tr>
<tr><td>|8 7> = 0.707|4 4>|4 3> + 0.707|4 3>|4 4></td><td>|4 4>|4 3>=0.707|8 7>,.... + |4 4>|3 4> =0.707|8 7> ,.....</td></tr>
<tr><td>|8 6> = 0.483|4 4>|4 2> + 0.730|4 3>|4 3> + 0.483|4 2>|4 4></td><td>|4 4>|4 2>=0.483|8 6>,.... + |4 4>|3 3> =0.730|8 6> ,.... + |4 4>|2 4> =0.483|8 6> ,.....</td></tr>
<tr><td>|8 5> = 0.316|4 4>|4 1> + 0.633|4 3>|4 2> + 0.633|4 2>|4 3> + 0.316|4 1>|4 4></td><td>|4 4>|4 1>=0.316|8 5>,.... + |4 4>|3 2> =0.632|8 5> ,.... + |4 4>|2 3> =0.632|8 5> ,....+ |4 4>|1 4>=0.316|8 5>,....</td></tr>
<tr><td>|8 4> = 0.196|4 4>|4 0> + 0.496|4 3>|4 1> + 0.656|4 2>|4 2> + 0.496|4 1>|4 3> + 0.196|4 0>|4 4></td><td>|4 4>|4 0>=0.195|8 4>,.... + |4 4>|3 1> =0.496|8 4> ,.... + |4 4>|2 2> =0.657|8 4> ,....+ |4 4>|1 3>=0.496|8 4>,... + |4 4>|0 4> =0.195|8 4>,....</td></tr>
<tr><td>|8 3> = 0.113|4 4>|4 -1> + 0.358|4 3>|4 0> + 0.599|4 2>|4 1> + 0.599|4 1>|4 2> + 0.358|4 0>|4 3> + 0.113|4 -1>|4 4></td><td>|4 4>|4 -1>=0.114|8 3>,.... + |4 4>|3 0> =0.358|8 3> ,.... + |4 4>|2 1> =0.599|8 3> ,....+ |4 4>|1 2>=0.599|8 3>,... + |4 4>|0 3> =0.358|8 3>,.... + |4 4>|-1 4>=0.114|8 3>,....</td></tr>
<tr><td>|8 2> = 0.059|4 4>|4 -2> + 0.237|4 3>|4 -1> + 0.495|4 2>|4 0> + 0.626|4 1>|4 1> + 0.495|4 0>|4 2> + 0.237|4 -1>|4 3> + 0.059|4 -2>|4 4></td><td>|4 4>|4 -2>=0.055|8 2>,.... + |4 4>|3 -1> =0.237|8 2> ,.... + |4 4>|2 0> =0.495|8 2> ,....+ |4 4>|1 1>=0.626|8 2>,... + |4 4>|0 2> =0.495|8 2>,.... + |4 4>|-1 3>=0.237|8 2>,.... + |4 4>|-2 4>=0.055|8 2>,....</td></tr>
<tr><td>|8 1> = 0.026|4 4>|4 -3> + 0.140|4 3>|4 -2> + 0.370|4 2>|4 -1> + 0.585|4 1>|4 0> + 0.585|4 0>|4 1> + 0.370|4 -1>|4 2> + 0.140|4 -2>|4 3> + 0.026|4 -3>|4 4></td><td>|4 4>|4 -3>=0.032|8 1>,.... + |4 4>|3 -2> =0.141|8 1> ,.... + |4 4>|2 -1> =0.370|8 1> ,....+ |4 4>|1 0>=0.586|8 1>,... + |4 4>|0 1> =0.586|8 1>,.... + |4 4>|-1 2>=0.370|8 1>,.... + |4 4>|-2 3>=0.141|8 1>,...+ |4 4>|-3 4>=0.032|8 1>,....</td></tr>
<tr><td>|8 0> = 0.009|4 4>|4 -4> + 0.071|4 3>|4 -3> + 0.247|4 2>|4 -2> + 0.494|4 1>|4 -1> + 0.617|4 0>|4 0> + 0.494|4 -1>|4 1> + 0.247|4 -2>|4 2> + 0.071|4 -3>|4 3> + 0.009|4 -4>|4 4></td><td>|4 4>|4 -4>=0.00|8 0>,.... + |4 4>|3 -3> =0.071|8 0> ,.... + |4 4>|2 -2> =0.247|8 0> ,....+ |4 4>|1 -1>=0.494|8 0>,... + |4 4>|0 0> =0.617|8 0>,.... + |4 4>|-1 1>=0.494|8 0>,.... + |4 4>|-2 2>=0.247|8 0>,...+ |4 4>|-3 3>=0.071|8 0>,.... + |4 4>|-4 4>=0.00|8 0>,....</td></tr>
<tr><td>|8 -1> = 0.026|4 3>|4 -4> + 0.140|4 2>|4 -3> + 0.370|4 1>|4 -2> + 0.585|4 0>|4 -1> + 0.585|4 -1>|4 0> + 0.370|4 -2>|4 1> + 0.140|4 -3>|4 2> + 0.026|4 -4>|4 3></td><td>|4 4>|3 -4>=0.032|8 -1>,.... + |4 4>|2 -3> =0.141|8 -1> ,.... + |4 4>|1 -2> =0.370|8 -1> ,....+ |4 4>|0 -1>=0.586|8 -1>,... + |4 4>|-1 0> =0.586|8 -1>,.... + |4 4>|-2 1>=0.370|8 -1>,.... + |4 4>|-3 2>=0.141|8 -1>,...+ |4 4>|-4 3>=0.032|8 -1>,....</td></tr>
</table>

∗A numerical algorithm for the explicit calculation of SU(N) and SL(N;C) Clebsch-Gordan coefficients[2].

The C++ code has an input of $j1$, $j2$ and jj which gives an output of all the combinations of $j1$, $j2$, $m1$, $m2$ and mm with their Clebsch-Gordan Coefficient whiles the FORTRAN code has an input of $j1$, $j2$, $m1$ and $m2$ to give an output of all combinations of jj and mm with their Clebsch-Gordan Coefficient. Therefore, apart from these they both give the same Clebsch-Gordan Coefficient as required and shown by the table above. The dots in FORTRAN code are other combinations of jj, mm with the Clebsch-Gordan Coefficient resulting from $j1$, $j2$, $m1$, $m2$.

One may notice some slight errors in the comparison of especially table (2) which are due to rounding off errors (thus, the truncation of the decimal values during the calculation.

4.4 Spin Ladder Energy Spectra and Spin

TABLE 4. The Spin Ladder of iron-sulphur cluster $[Fe_4S_4(SCH_3)_4]^{-1}$ showing the Energy, E, Spins, S and its states using the Heisenberg-Dirac-van Vleck Model.

States	Energy	Spin	States	Energy	Spin
1:	-1949.007006	3.00	2:	-1949.007006	3.00
3:	-1949.007006	3.00	4:	-1924.200877	1.00
5:	-1818.087766	3.00	6:	-1818.087766	3.00
7:	-1818.087766	3.00	8:	-1784.181555	5.00
9:	-1784.181555	5.00	10:	-1784.181555	5.00
11:	-1784.181555	5.00	12:	-1784.181555	5.00
13:	-1737.407749	3.00	14:	-1737.407749	3.00
15:	-1737.407749	3.00	16:	-1720.164752	5.00
17:	-1720.164752	5.00	18:	-1720.164752	5.00
19:	-1720.164752	5.00	20:	-1720.164752	5.00
21:	-1711.868588	1.00	22:	-1617.139528	3.00
23:	-1617.139528	3.00	24:	-1617.139528	3.00
25:	-1604.530999	5.00	26:	-1604.530999	5.00
27:	-1604.530999	5.00	28:	-1604.530999	5.00
29:	-1604.530999	5.00	30:	-1578.071043	3.00
31:	-1578.071043	3.00	32:	-1578.071043	3.00
33:	-1557.189036	1.00	34:	-1556.422470	5.00
35:	-1556.422470	5.00	36:	-1556.422470	5.00
37:	-1556.422470	5.00	38:	-1556.422470	5.00
39:	-1491.701749	5.00	40:	-1491.701749	5.00
41:	-1491.701749	5.00	42:	-1491.701749	5.00
43:	-1491.701749	5.00	44:	-1478.926766	3.00
45:	-1478.926766	3.00	46:	-1478.926766	3.00

47:	-1456.492895	1.00	48:	-1447.773675	7.00
49:	-1447.773675	7.00	50:	-1447.773675	7.00
51:	-1447.773675	7.00	52:	-1447.773675	7.00
53:	-1447.773675	7.00	54:	-1447.773675	7.00
55:	-1429.247804	5.00	56:	-1429.247804	5.00
57:	-1429.247804	5.00	58:	-1429.247804	5.00
59:	-1429.247804	5.00	60:	-1397.848736	3.00
61:	-1397.848736	3.00	62:	-1397.848736	3.00
63:	-1391.783006	3.00	64:	-1391.783006	3.00
65:	-1391.783006	3.00	66:	-1388.898224	7.00
67:	-1388.898224	7.00	68:	-1388.898224	7.00
69:	-1388.898224	7.00	70:	-1388.898224	7.00
71:	-1388.898224	7.00	72:	-1388.898224	7.00
73:	-1344.106001	7.00	74:	-1344.106001	7.00
75:	-1344.106001	7.00	76:	-1344.106001	7.00
77:	-1344.106001	7.00	78:	-1344.106001	7.00
79:	-1344.106001	7.00	80:	-1338.386641	1.00
81:	-1333.877055	5.00	82:	-1333.877055	5.00
83:	-1333.877055	5.00	84:	-1333.877055	5.00
85:	-1333.877055	5.00	86:	-1279.913366	7.00
87:	-1279.913366	7.00	88:	-1279.913366	7.00
89:	-1279.913366	7.00	90:	-1279.913366	7.00
91:	-1279.913366	7.00	92:	-1279.913366	7.00
93:	-1278.506614	3.00	94:	-1278.506614	3.00
95:	-1278.506614	3.00	96:	-1263.170669	5.00
97:	-1263.170669	5.00	98:	-1263.170669	5.00
99:	-1263.170669	5.00	100:	-1263.170669	5.00
101:	-1232.489224	3.00	102:	-1232.489224	3.00
103:	-1232.489224	3.00	104:	-1211.529865	5.00
105:	-1211.529865	5.00	106:	-1211.529865	5.00
107:	-1211.529865	5.00	108:	-1211.529865	5.00
109:	-1174.269469	7.00	110:	-1174.269469	7.00
111:	-1174.269469	7.00	112:	-1174.269469	7.00
113:	-1174.269469	7.00	114:	-1174.269469	7.00
115:	-1174.269469	7.00	116:	-1159.099335	5.00
117:	-1159.099335	5.00	118:	-1159.099335	5.00
119:	-1159.099335	5.00	120:	-1159.099335	5.00
121:	-1133.811359	7.00	122:	-1133.811359	7.00
123:	-1133.811359	7.00	124:	-1133.811359	7.00
125:	-1133.811359	7.00	126:	-1133.811359	7.00
127:	-1133.811359	7.00	128:	-1107.692260	3.00

129:	-1107.692260	3.00	130:	-1107.692260	3.00
131:	-1067.859232	7.00	132:	-1067.859232	7.00
133:	-1067.859232	7.00	134:	-1067.859232	7.00
135:	-1067.859232	7.00	136:	-1067.859232	7.00
137:	-1067.859232	7.00	138:	-1067.838722	5.00
139:	-1067.838722	5.00	140:	-1067.838722	5.00
141:	-1067.838722	5.00	142:	-1067.838722	5.00
143:	-986.943984	3.00	144:	-986.943984	3.00
145:	-986.943984	3.00	146:	-980.604061	7.00
147:	-980.604061	7.00	148:	-980.604061	7.00
149:	-980.604061	7.00	150:	-980.604061	7.00
151:	-980.604061	7.00	152:	-980.604061	7.00
153:	-960.147980	5.00	154:	-960.147980	5.00
155:	-960.147980	5.00	156:	-960.147980	5.00
157:	-960.147980	5.00	158:	-938.562037	9.00
159:	-938.562037	9.00	160:	-938.562037	9.00
161:	-938.562037	9.00	162:	-938.562037	9.00
163:	-938.562037	9.00	164:	-938.562037	9.00
165:	-938.562037	9.00	166:	-938.562037	9.00
167:	-934.202647	7.00	168:	-934.202647	7.00
169:	-934.202647	7.00	170:	-934.202647	7.00
171:	-934.202647	7.00	172:	-934.202647	7.00
173:	-934.202647	7.00	174:	-883.328780	9.00
175:	-883.328780	9.00	176:	-883.328780	9.00
177:	-883.328780	9.00	178:	-883.328780	9.00
179:	-883.328780	9.00	180:	-883.328780	9.00
181:	-883.328780	9.00	182:	-883.328780	9.00
183:	-847.589660	5.00	184:	-847.589660	5.00
185:	-847.589660	5.00	186:	-847.589660	5.00
187:	-847.589660	5.00	188:	-838.992899	9.00
189:	-838.992899	9.00	190:	-838.992899	9.00
191:	-838.992899	9.00	192:	-838.992899	9.00
193:	-838.992899	9.00	194:	-838.992899	9.00
195:	-838.992899	9.00	196:	-838.992899	9.00
197:	-827.133554	7.00	198:	-827.133554	7.00
199:	-827.133554	7.00	200:	-827.133554	7.00
201:	-827.133554	7.00	202:	-827.133554	7.00
203:	-827.133554	7.00	204:	-801.274893	9.00
205:	-801.274893	9.00	206:	-801.274893	9.00
207:	-801.274893	9.00	208:	-801.274893	9.00
209:	-801.274893	9.00	210:	-801.274893	9.00

211:	-801.274893	9.00	212:	-801.274893	9.00
213:	-742.837644	9.00	214:	-742.837644	9.00
215:	-742.837644	9.00	216:	-742.837644	9.00
217:	-742.837644	9.00	218:	-742.837644	9.00
219:	-742.837644	9.00	220:	-742.837644	9.00
221:	-742.837644	9.00	222:	-741.913180	7.00
223:	-741.913180	7.00	224:	-741.913180	7.00
225:	-741.913180	7.00	226:	-741.913180	7.00
227:	-741.913180	7.00	228:	-741.913180	7.00
229:	-721.066847	5.00	230:	-721.066847	5.00
231:	-721.066847	5.00	232:	-721.066847	5.00
233:	-721.066847	5.00	234:	-639.739205	7.00
235:	-639.739205	7.00	236:	-639.739205	7.00
237:	-639.739205	7.00	238:	-639.739205	7.00
239:	-639.739205	7.00	240:	-639.739205	7.00
241:	-576.163812	5.00	242:	-576.163812	5.00
243:	-576.163812	5.00	244:	-576.163812	5.00
245:	-576.163812	5.00	246:	-575.230353	9.00
247:	-575.230353	9.00	248:	-575.230353	9.00
249:	-575.230353	9.00	250:	-575.230353	9.00
251:	-575.230353	9.00	252:	-575.230353	9.00
253:	-575.230353	9.00	254:	-575.230353	9.00
255:	-520.877976	9.00	256:	-520.877976	9.00
257:	-520.877976	9.00	258:	-520.877976	9.00
259:	-520.877976	9.00	260:	-520.877976	9.00
261:	-520.877976	9.00	262:	-520.877976	9.00
263:	-520.877976	9.00	264:	-467.345576	7.00
265:	-467.345576	7.00	266:	-467.345576	7.00
267:	-467.345576	7.00	268:	-467.345576	7.00
269:	-467.345576	7.00	270:	-467.345576	7.00
271:	-453.583400	9.00	272:	-453.583400	9.00
273:	-453.583400	9.00	274:	-453.583400	9.00
275:	-453.583400	9.00	276:	-453.583400	9.00
277:	-453.583400	9.00	278:	-453.583400	9.00
279:	-453.583400	9.00	280:	-444.798017	5.00
281:	-444.798017	5.00	282:	-444.798017	5.00
283:	-444.798017	5.00	284:	-444.798017	5.00
285:	-364.369595	9.00	286:	-364.369595	9.00
287:	-364.369595	9.00	288:	-364.369595	9.00
289:	-364.369595	9.00	290:	-364.369595	9.00
291:	-364.369595	9.00	292:	-364.369595	9.00

293:	-364.369595	9.00	294:	-342.513759	7.00
295:	-342.513759	7.00	296:	-342.513759	7.00
297:	-342.513759	7.00	298:	-342.513759	7.00
299:	-342.513759	7.00	300:	-342.513759	7.00
301:	-223.308894	7.00	302:	-223.308894	7.00
303:	-223.308894	7.00	304:	-223.308894	7.00
305:	-223.308894	7.00	306:	-223.308894	7.00
307:	-223.308894	7.00	308:	-192.200965	11.00
309:	-192.200965	11.00	310:	-192.200965	11.00
311:	-192.200965	11.00	312:	-192.200965	11.00
313:	-192.200965	11.00	314:	-192.200965	11.00
315:	-192.200965	11.00	316:	-192.200965	11.00
317:	-192.200965	11.00	318:	-192.200965	11.00
319:	-144.970855	11.00	320:	-144.970855	11.00
321:	-144.970855	11.00	322:	-144.970855	11.00
323:	-144.970855	11.00	324:	-144.970855	11.00
325:	-144.970855	11.00	326:	-144.970855	11.00
327:	-144.970855	11.00	328:	-144.970855	11.00
329:	-144.970855	11.00	330:	-133.598334	9.00
331:	-133.598334	9.00	332:	-133.598334	9.00
333:	-133.598334	9.00	334:	-133.598334	9.00
335:	-133.598334	9.00	336:	-133.598334	9.00
337:	-133.598334	9.00	338:	-133.598334	9.00
339:	-91.964235	11.00	340:	-91.964235	11.00
341:	-91.964235	11.00	342:	-91.964235	11.00
343:	-91.964235	11.00	344:	-91.964235	11.00
345:	-91.964235	11.00	346:	-91.964235	11.00
347:	-91.964235	11.00	348:	-91.964235	11.00
349:	-91.964235	11.00	350:	-34.531285	9.00
351:	-34.531285	9.00	352:	-34.531285	9.00
353:	-34.531285	9.00	354:	-34.531285	9.00
355:	-34.531285	9.00	356:	-34.531285	9.00
357:	-34.531285	9.00	358:	-34.531285	9.00
359:	-22.053677	11.00	360:	-22.053677	11.00
361:	-22.053677	11.00	362:	-22.053677	11.00
363:	-22.053677	11.00	364:	-22.053677	11.00
365:	-22.053677	11.00	366:	-22.053677	11.00
367:	-22.053677	11.00	368:	-22.053677	11.00
369:	-22.053677	11.00	370:	70.581118	9.00
371:	70.581118	9.00	372:	70.581118	9.00
373:	70.581118	9.00	374:	70.581118	9.00

375:	70.581118	9.00	376:	70.581118	9.00
377:	70.581118	9.00	378:	70.581118	9.00
379:	134.275850	7.00	380:	134.275850	7.00
381:	134.275850	7.00	382:	134.275850	7.00
383:	134.275850	7.00	384:	134.275850	7.00
385:	134.275850	7.00	386:	272.743711	11.00
387:	272.743711	11.00	388:	272.743711	11.00
389:	272.743711	11.00	390:	272.743711	11.00
391:	272.743711	11.00	392:	272.743711	11.00
393:	272.743711	11.00	394:	272.743711	11.00
395:	272.743711	11.00	396:	272.743711	11.00
397:	279.281690	7.00	398:	279.281690	7.00
399:	279.281690	7.00	400:	279.281690	7.00
401:	279.281690	7.00	402:	279.281690	7.00
403:	279.281690	7.00	404:	345.292525	11.00
405:	345.292525	11.00	406:	345.292525	11.00
407:	345.292525	11.00	408:	345.292525	11.00
409:	345.292525	11.00	410:	345.292525	11.00
411:	345.292525	11.00	412:	345.292525	11.00
413:	345.292525	11.00	414:	345.292525	11.00
415:	436.070428	11.00	416:	436.070428	11.00
417:	436.070428	11.00	418:	436.070428	11.00
419:	436.070428	11.00	420:	436.070428	11.00
421:	436.070428	11.00	422:	436.070428	11.00
423:	436.070428	11.00	424:	436.070428	11.00
425:	436.070428	11.00	426:	455.378786	9.00
427:	455.378786	9.00	428:	455.378786	9.00
429:	455.378786	9.00	430:	455.378786	9.00
431:	455.378786	9.00	432:	455.378786	9.00
433:	455.378786	9.00	434:	455.378786	9.00
435:	584.776145	9.00	436:	584.776145	9.00
437:	584.776145	9.00	438:	584.776145	9.00
439:	584.776145	9.00	440:	584.776145	9.00
441:	584.776145	9.00	442:	584.776145	9.00
443:	584.776145	9.00	444:	737.372250	13.00
445:	737.372250	13.00	446:	737.372250	13.00
447:	737.372250	13.00	448:	737.372250	13.00
449:	737.372250	13.00	450:	737.372250	13.00
451:	737.372250	13.00	452:	737.372250	13.00
453:	737.372250	13.00	454:	737.372250	13.00
455:	737.372250	13.00	456:	737.372250	13.00

457:	796.505612	13.00	458:	796.505612	13.00
459:	796.505612	13.00	460:	796.505612	13.00
461:	796.505612	13.00	462:	796.505612	13.00
463:	796.505612	13.00	464:	796.505612	13.00
465:	796.505612	13.00	466:	796.505612	13.00
467:	796.505612	13.00	468:	796.505612	13.00
469:	796.505612	13.00	470:	853.210946	11.00
471:	853.210946	11.00	472:	853.210946	11.00
473:	853.210946	11.00	474:	853.210946	11.00
475:	853.210946	11.00	476:	853.210946	11.00
477:	853.210946	11.00	478:	853.210946	11.00
479:	853.210946	11.00	480:	853.210946	11.00
481:	872.693244	13.00	482:	872.693244	13.00
483:	872.693244	13.00	484:	872.693244	13.00
485:	872.693244	13.00	486:	872.693244	13.00
487:	872.693244	13.00	488:	872.693244	13.00
489:	872.693244	13.00	490:	872.693244	13.00
491:	872.693244	13.00	492:	872.693244	13.00
493:	872.693244	13.00	494:	964.823255	11.00
495:	964.823255	11.00	496:	964.823255	11.00
497:	964.823255	11.00	498:	964.823255	11.00
499:	964.823255	11.00	500:	964.823255	11.00
501:	964.823255	11.00	502:	964.823255	11.00
503:	964.823255	11.00	504:	964.823255	11.00
505:	1182.382129	9.00	506:	1182.382129	9.00
507:	1182.382129	9.00	508:	1182.382129	9.00
509:	1182.382129	9.00	510:	1182.382129	9.00
511:	1182.382129	9.00	512:	1182.382129	9.00
513:	1182.382129	9.00	514:	1324.541813	13.00
515:	1324.541813	13.00	516:	1324.541813	13.00
517:	1324.541813	13.00	518:	1324.541813	13.00
519:	1324.541813	13.00	520:	1324.541813	13.00
521:	1324.541813	13.00	522:	1324.541813	13.00
523:	1324.541813	13.00	524:	1324.541813	13.00
525:	1324.541813	13.00	526:	1324.541813	13.00
527:	1418.412388	13.00	528:	1418.412388	13.00
529:	1418.412388	13.00	530:	1418.412388	13.00
531:	1418.412388	13.00	532:	1418.412388	13.00
533:	1418.412388	13.00	534:	1418.412388	13.00
535:	1418.412388	13.00	536:	1418.412388	13.00
537:	1418.412388	13.00	538:	1418.412388	13.00

539:	1418.412388	13.00	540:	1573.117887	11.00
541:	1573.117887	11.00	542:	1573.117887	11.00
543:	1573.117887	11.00	544:	1573.117887	11.00
545:	1573.117887	11.00	546:	1573.117887	11.00
547:	1573.117887	11.00	548:	1573.117887	11.00
549:	1573.117887	11.00	550:	1573.117887	11.00
551:	1864.756090	15.00	552:	1864.756090	15.00
553:	1864.756090	15.00	554:	1864.756090	15.00
555:	1864.756090	15.00	556:	1864.756090	15.00
557:	1864.756090	15.00	558:	1864.756090	15.00
559:	1864.756090	15.00	560:	1864.756090	15.00
561:	1864.756090	15.00	562:	1864.756090	15.00
563:	1864.756090	15.00	564:	1864.756090	15.00
565:	1864.756090	15.00	566:	1944.685853	15.00
567:	1944.685853	15.00	568:	1944.685853	15.00
569:	1944.685853	15.00	570:	1944.685853	15.00
571:	1944.685853	15.00	572:	1944.685853	15.00
573:	1944.685853	15.00	574:	1944.685853	15.00
575:	1944.685853	15.00	576:	1944.685853	15.00
577:	1944.685853	15.00	578:	1944.685853	15.00
579:	1944.685853	15.00	580:	1944.685853	15.00
581:	2039.798926	13.00	582:	2039.798926	13.00
583:	2039.798926	13.00	584:	2039.798926	13.00
585:	2039.798926	13.00	586:	2039.798926	13.00
587:	2039.798926	13.00	588:	2039.798926	13.00
589:	2039.798926	13.00	590:	2039.798926	13.00
591:	2039.798926	13.00	592:	2039.798926	13.00
593:	2039.798926	13.00	594:	2581.068486	15.00
595:	2581.068486	15.00	596:	2581.068486	15.00
597:	2581.068486	15.00	598:	2581.068486	15.00
599:	2581.068486	15.00	600:	2581.068486	15.00
601:	2581.068486	15.00	602:	2581.068486	15.00
603:	2581.068486	15.00	604:	2581.068486	15.00
605:	2581.068486	15.00	606:	2581.068486	15.00
607:	2581.068486	15.00	608:	2581.068486	15.00
609:	3195.255215	17.00	610:	3195.255215	17.00
611:	3195.255215	17.00	612:	3195.255215	17.00
613:	3195.255215	17.00	614:	3195.255215	17.00
615:	3195.255215	17.00	616:	3195.255215	17.00
617:	3195.255215	17.00	618:	3195.255215	17.00
619:	3195.255215	17.00	620:	3195.255215	17.00

621:	3195.255215	17.00	622:	3195.255215	17.00
623:	3195.255215	17.00	624:	3195.255215	17.00
625:	3195.255215	17.00			

A Appendix

A.1 Bra - Ket Notation

It is conventional to use bra-ket notation for wave functions and multi-dimensional integrals in electronic structure theory in order to simplify the notation. The equivalences are defined as:

$$| \Psi\rangle = \Psi; \ \langle\Psi |= \Psi^*$$
$$\int \Psi^* \Psi dr = \langle\Psi | \Psi\rangle$$
$$\int \Psi^* H \Psi dr = \langle\Psi | H | \Psi\rangle$$

The bra $\langle n|$ denotes a complex conjugate wave function with quantum number, n, standing to the left of the operator, while the ket $|m\rangle$ denotes a wave function with quantum number, m, standing to the right of the operator, and the combined bracket denotes that the whole expression should be integrated over all coordinates. Such a bracket is often referred to as a matrix element, or as an overlap element when there is no operator involved.

A.2 The Variational Principle

The Variational Principle states that an approximate wave function has an energy that is above or equal to the exact energy. The equality holds only if the wave function is exact. The proof is as follows.
Assume that we know the exact solutions to the Schrödinger equation.

$$H\Psi_i = E_i \Psi_i \ i = 0, 1, 2, ...\infty \tag{A.1}$$

There are infinitely many solutions and we assume that they are labelled according to their energies, E_0 being the lowest. Since the H operator is Hermitian, the solutions form a complete basis. We may furthermore choose the solutions to be orthogonal and normalized.

$$\langle\Psi_i | \Psi_j\rangle = \delta_{ij} \tag{A.2}$$

An approximate wave function can be expanded in the exact solutions, since they form a complete set.

$$\Phi = \sum_{ij}^{\infty} a_i \Psi_i \tag{A.3}$$

The energy of an approximate wave function is calculated as

$$E_{app} = \frac{\langle \Phi \mid H \mid \Phi \rangle}{\langle \Phi \mid \Phi \rangle} \tag{A.4}$$

Inserting eqn A.3, we obtain an expansion as

$$E_{app} = \frac{\sum_{i=0}^{\infty} \sum_{j=0}^{\infty} a_i a_j \langle \Psi_i \mid H \mid \Psi_j \rangle}{\sum_{i=0}^{\infty} \sum_{j=0}^{\infty} a_i a_j \langle \Psi_i \mid \Psi_j \rangle} \tag{A.5}$$

Using the fact that $H\Phi_i = E_i\Phi_i$ and the orthonormality of the Φ_i we obtain

$$E_{app} = \frac{\sum_{i=0}^{\infty} a_i^2 E_i}{\sum_{i=0}^{\infty} a_i^2} \tag{A.6}$$

Then, the variational principle states that $E_{app} \geq E_0$ or, equivalently, $E_{app} - E_0 \geq 0$.

$$E_{app} - E_0 = \frac{\sum_{i=0}^{\infty} a_i^2 E_i}{\sum_{i=0}^{\infty} a_i^2} - E_0 = \frac{\sum_{i=0}^{\infty} a_i^2 (E_i - E_0)}{\sum_{i=0}^{\infty} a_i^2} \geq 0 \tag{A.7}$$

Since a_i^2 is always positive or zero, and $E_i - E_0$ is always positive or zero (E_0 is by definition the lowest energy). Furthermore, in order for the equal sign to hold, all $a_{i\neq 0} = 0$ since $E_{i\neq 0} - E_0$ is non-zero (neglecting degenerate ground states). This in turns means that $a_0 = 1$, owing to the normalization of Ψ, and consequently the wave function is the exact solution.

This proof shows that any approximate wave function will have an energy above or equal to the exact ground state energy.

A.3 Exchange Coupling Constant

```
!/*******************************************************
PROGRAM SVD
  USE PARAMETERS

        IMPLICIT NONE
!********************************************************
        real(kind=dp), ALLOCATABLE :: s(:), energies(:)
        real(kind=dp), ALLOCATABLE :: A(:,:), U(:,:), Vt
           (:,:), Sigma(:,:)
        real(kind=dp), ALLOCATABLE :: M_matrix(:,:),M_ave
           (:), J_values(:), Mul_Matrix(:)
        real(kind=dp), ALLOCATABLE :: tmp(:),tmp_ava(:),
           temp_ave(:,:)
        real(kind=dp), ALLOCATABLE :: work(:)
        INTEGER, ALLOCATABLE :: desca(:),descu(:),descvt
           (:)
        INTEGER :: M,N,P, ifail, i,j,k,r, info, lwork
        real(kind=dp) :: fatt_au2cm1, trace,det
        INTEGER   ::    iostat
        INTEGER, PARAMETER     :: unit_in=27,unit_out=29
        CHARACTER(len=20)  :: FMT
        fatt_au2cm1=219474.63_dp


!OPEN the M_values.dat and read first line with N&M
!!!!!!!!!! READ INPUT VALUES !!!!!!!!!!!
!write(*,'(a)',advance='no')'Reading Inputs...
                  '
        iostat=0
        open(unit=unit_in,file='M_values.dat',status='old
           ',iostat=iostat)
        if(iostat>0)STOP "Error Opening File M_values.dat
           "
        read(unit_in,*)M,N
        allocate(M_matrix(M,N)) ; M_matrix=0.0_dp
        allocate(M_ave(N))      ; M_ave=0.0_dp
!Read and make averages for all M_i values
!Make averages along columns (absolute value)
        do i=1,M
          read(unit_in,*)(M_matrix(i,j),j=1,N)
          do j=1,N
            M_ave(j)=M_ave(j)+abs(M_matrix(i,j))
          enddo
        enddo
        M_ave(:)=M_ave(:)/M
        close(unit_in)
```

```
!Make the proper multiplications to build the M matrix M_i*M_j/4
!Store this M matrix in the matrix A below.
        P=N*(N-1)/2
        allocate (A(M,P))      ;A=0.0_dp
        do k=1,M
          r=1
          do i=1,N
            do j=i+1,N
              A(k,r)=sign(M_ave(i),M_matrix(k,i))*sign(M_ave(j),M_matrix(k,j))/4.0_dp
              r=r+1
            enddo
          enddo
        enddo
        write(stdout,*)'Done!'


        allocate (U(M,P))      ;U=0.0_dp
        allocate(Vt(P,P))      ;Vt=0.0_dp
        allocate(s(P))         ;s=0.0_dp
        lwork=6*P
        allocate(work(lwork))
!        allocate (tmp_ava(dimA))     ;tmp_ava=0
!        allocate (tmp_ave(dimA,dimA));tmp_ave=0
!        allocate (avalori(dimA))      ;avalori=0
!        allocate (avettori(dimA,dimA)) ;avettori=0

        lwork=(M)**2 ; ifail=0

        do i=1,M
          write (stdout,'(12f10.5)') (A(i,j),j=1,P)
        enddo
        A=A*2
!        E=-(2M)*J

!call dgesvd(jobu, jobvt, m, n, a, lda, s, u, ldu, vt, ldvt, work, lwork, info)
call dgesvd('S','S', M, P, A, M, s, U, M, Vt, P, work, lwork, info)

!Write the results for the procedure SVD
        write (stdout,*) 'Matrix U:'
        do i=1,M
          write (stdout,'(12f10.5)') (U(i,j),j=1,P)
        enddo


        write (stdout,*) 'Matrix Vt:'
        do i=1,P
          write (stdout,'(12f10.5)') (Vt(i,j),j=1,P)
        enddo

        write (stdout,*) 'diag Matrix S:'
        write (stdout,'(12f10.5)') (s(j),j=1,P)

!Decomposition of A=U*Sigma*Vt
        allocate(Sigma(P,P))      ;Sigma=0.0_dp
        do i=1,P
          Sigma(i,i)=s(i)
        enddo

       allocate(temp_ave(P,P)) ;temp_ave=0.0_dp
       do r=1,P
        do i=1,P
           do j=1,P
            temp_ave(i,j)=temp_ave(i,j)+Sigma(i,r)*Vt(r,j)
           enddo
        enddo
       enddo
       A=0
       do r=1,P
        do i=1,M
           do j=1,P
            A(i,j)=A(i,j)+U(i,r)*temp_ave(r,j)
           enddo
        enddo
       enddo
!Divide A by 2 to reduce its Doubling
       A=A/2
       write (stdout,*) 'Matrix A: '
       do i=1,M
          write (stdout,'(12f10.5)') (A(i,j),j=1,P)
       enddo

        write (stdout,*) 'Starting J_ij calculations.'

        allocate(energies(M))         ;energies=0.0_dp
        allocate(J_values(P))         ;J_values=0.0_dp

!write (*,*) ' Energies of the ',M,' states will be read from a file '
!Read (*,*) Energies
        open (unit_in,file='Energies.dat',status='old')
        do i=1,M
          read (unit_in,*) (energies(i))
        enddo
        close(unit_in)

        do i=1,M
           write (stdout,'(1f15.5)') energies(i)
        enddo

!Invert the Diagonal Matrix
        Sigma=0.0_dp
        do i=1,P
          Sigma(i,i)=1/s(i)
        enddo
!Calculate the inverse Matrix A: (A^-1)^T=U*Sigma^-1*Vt
       temp_ave=0.0_dp
       do r=1,P
        do i=1,P
           do j=1,P
            temp_ave(i,j)=temp_ave(i,j)+Sigma(i,r)*Vt(r,j
               )
           enddo
        enddo
       enddo
       A=0.0_dp
       do r=1,P
        do i=1,M
           do j=1,P
            A(i,j)=A(i,j)+U(i,r)*temp_ave(r,j)
           enddo
        enddo
        enddo


        do i=1,P
          do j=1,M
              J_values(i)=J_values(i)-A(j,i)*energies(j)
          enddo
        enddo
        J_values=J_values*fatt_au2cm1
!Write J
        write (stdout,*) ' The J values: '
        do i=1,P
           write (stdout,'(1f15.5)') J_values(i)
        enddo
!Output of number of Spin centers(N), Spin Values and J
   values as Input to Spin Ladder
        open (unit=unit_out,file="SpinLadder.inp",action=
           "write",status="replace")
        write (unit_out,'(I3)')N
        write(FMT,'(a,i,a)')'(',N,'F5.1)'
        write (unit_out,FMT)(real(int(M_ave(i)+0.5_dp))
           /2.0_dp, i=1,N)
        write(FMT,'(a,i,a)')'(',P,'F10.4)'
        write (unit_out,FMT) ( J_values(i), i=1,P )
        close(unit_out)

end
```

Listing 1: Exchange Coupling Constant Pseudocode.

A.4 Basis Set

```
SUBROUTINE get_Basis_Set(spin_array,basis_set)
real(kind=dp),dimension(:), intent(in)      :: spin_array
real(kind=dp),dimension(:,:),intent(out)    :: basis_set

integer   :: num_spin
integer    :: num_state
integer    :: s
real(kind=dp),dimension(size(spin_array))   :: m_array

num_state=size(basis_set(1,:))
num_spin=size(spin_array)

m_array(:)=-spin_array(:)

!Creating an entire Matrix to contain both the Spins
   repeatition and the basis set Matrix
do s=1,num_state
!    basis_set(1:num_spin,s)=spin_array
!    basis_set(num_spin+1:2*num_spin,s)=m_array
    basis_set(:,s)=m_array
    call next(m_array,spin_array)
end do


CONTAINS


recursive subroutine next(m_array,spin_array)
  real(kind=dp),dimension(:), intent(inout) :: m_array
  real(kind=dp),dimension(:), intent(in) :: spin_array
  integer :: num_spin
  num_spin=size(spin_array)
  m_array(num_spin)=m_array(num_spin)+1.0_dp
  if((num_spin.gt.1).and.(m_array(num_spin).gt.(
     spin_array(num_spin)+EPS)))then
     m_array(num_spin)=-1.0_dp*spin_array(num_spin)
     call next(m_array(1:num_spin-1),spin_array(1:
        num_spin-1))
  endif
end subroutine next

End Subroutine get_Basis_Set
```

Listing 2: Basis Set Pseudocode.

A.5 Clebsch-Gordan Coefficient

The overall sign of the coefficients for each set of constant s_1, s_2, S is arbitrary to some degree and has been fixed according to the Condon-Shortley and Wigner sign convention[10]. These are solutions to

$$\mid s_i s_j; m_i m_j\rangle = \sum_{m_1=-s_1}^{s_1} \sum_{m_2=-s_2}^{s_2} \mid s_1 m_1 s_2 m_2\rangle\langle s_1 s_2; m_1 m_2 \mid s_1 s_2; SM\rangle \tag{A.8}$$

Explicitly:

$$\begin{aligned}\langle s_1 s_2; m_1 m_2 \mid s_1 s_2; SM\rangle &= \\ = \delta_{M,m_1+m_2}\sqrt{\frac{(2S+1)(S+s_1-s_2)!(S-s_1+s_2)!(s_1+s_2-S)!}{(s_1+s_2+S+1)!}}&\times \\ \times\sqrt{(S+M)!(S-M)!(s_1-m_1)!(s_1+m_1)!(s_2-m_2)!(s_2+m_2)!}&\times \\ \times\sum_k \frac{(-1)^k}{k!(s_1+s_2-S-k)!(s_1-m_1-k)!(s_2+m_2-k)!(S-s_2+m_1+k)!(S-s_1-m_2+k)!}\end{aligned} \tag{A.9}$$

Condition: The summation is extended over all integer k for which the argument of every factorial is non-negative[7]. These mathematically is given as

- $k \geq 0$
- $k \leq s_1 + s_2 - S$
- $k \leq s_1 - m_1$
- $k \leq s_2 + m_2$
- $k \geq -S + s_2 - m_1$
- $k \geq -S + s_1 + m_2$

For brevity, solution with $M < 0$ and $s_1 < s_2$ are omitted. Then, the calculation is given as

$$\langle s_1 s_2; m_1 m_2 \mid s_1 s_2; SM\rangle = (-1)^{S-s_1-s_2}\langle s_1 s_2; -m_1, -m_2 \mid s_1 s_2; S, -M\rangle \tag{A.10}$$

and

$$\langle s_1 s_2; m_1 m_2 \mid s_1 s_2; SM\rangle = (-1)^{S-s_1-s_2}\langle s_2 s_1; m_2 m_1 \mid s_2 s_1; SM\rangle \tag{A.11}$$

```
MODULE TOOLS

USE PARAMETERS

IMPLICIT NONE

PRIVATE

real(kind=dp), PARAMETER, PRIVATE :: EPS = 0.0001_dp

PUBLIC :: Calculate_CG

CONTAINS





SUBROUTINE Calculate_CG(state,jj_list,CG_list,size_list)
real(kind=dp),dimension(:),intent(in)    ::   state
real(kind=dp), dimension(:),intent(out)  :: jj_list
real(kind=dp), dimension(:),intent(out)  :: CG_list
INTEGER, intent(out)                 :: size_list

real(kind=dp) :: j1,j2,m1,m2
real(kind=dp) :: mm,jj,mm_max,mm_min,jj_min,jj_max
real(kind=dp) :: a,b,c,d,z,Spin_SQR
INTEGER :: i,k,n_cg,N_jj,N_mm

!! -------------------------------------------------- !!
!             s1,s2,s3,...,sn;m1,m2,m3,....mn
!           ASSIGNING VALUES FROM THE BASIS SET
!! -------------------------------------------------- !!
  j1=state(1)
  j2=state(2)
  m1=state(3)
  m2=state(4)

  a=0.0_dp;b=0.0_dp;c=0.0_dp;d=0.0_dp;z=0.0_dp;Spin_SQR
     =0.0_dp
  CG_list(:)=0.0_dp
  jj_list(:)=0.0_dp
  n_cg=0

!! -------------------------------------------------- !!
!     CONDITION FOR RANGE OF SPINS AND BASIS SET
!         m = |s1 - s2| , s1 + s2
!! -------------------------------------------------- !!
```

```
  jj_min = abs(j1-j2)
  jj_max = j1+j2
  N_jj= int((jj_max - jj_min)+ 1)
  do i=0,N_jj-1
    jj=i+jj_min
    mm_min=(-jj)
    mm_max=(+jj)
    N_mm=int(mm_max-mm_min+1)
    do k=0,N_mm-1
      mm=k+mm_min
      d=0.0_dp

!! -------------------------------------------------- !!
!   OBTAINING THE LISTS OF RESULTS FOR CLEBSCH-GORDAN
!        COEFFIECENT AND TOTAL SPIN, S
!! -------------------------------------------------- !!

      if(CHECK(m1,m2,j1,j2,mm,jj))then
          CALL Calculate_one_CG(m1,m2,j1,j2,mm,jj,d)
          n_cg=n_cg+1
          CG_list(n_cg)=d
          jj_list(n_cg)=real(jj,kind=dp)
       end if
    end do
  end do
  size_list=n_cg


END SUBROUTINE Calculate_CG



SUBROUTINE Calculate_one_CG(v_m1,v_m2,v_j1,v_j2,v_mm,v_jj,d)
real(kind=dp),INTENT(IN)   :: v_m1,v_m2,v_j1,v_j2
real(kind=dp),INTENT(IN)  :: v_jj,v_mm
real(kind=dp)              :: m1,m2,mm,j1,j2,jj
real(kind=dp)              :: segno,tmp,a,b,c
real(kind=dp),INTENT(OUT)  :: d
INTEGER :: k_min,k_max,z
     m1=v_m1
     m2=v_m2
     mm=v_mm
     j1=v_j1
     j2=v_j2
     jj=v_jj
     segno=1.0_dp

```

```
!! --------------------------------------------------- !!
!  CONDITIONS FOR CALCULATING CLEBSCH-GORDAN COEFFIECENT
!     FOR SOLUTIONS with m < 0 and j1 < j2
!! --------------------------------------------------- !!

     if (abs(m1+m2-mm)<eps) then
       if(mm.lt.-eps)then
         m1=-v_m1
         m2=-v_m2
         mm=-v_mm
         if(abs(mod(v_jj-v_j1-v_j2,2.0_dp)).gt.eps)then
           segno=-1.0_dp
         endif
       endif
       if(j2-j1.gt.eps)then
          j1=v_j2
          j2=v_j1
          m1=v_m2
          m2=v_m1
          if(abs(mod(v_jj-v_j1-v_j2,2.0_dp)).gt.eps)then
             segno=-1.0_dp
           endif
       endif

!! --------------------------------------------------- !!
! CONDITIONS FOR CALCULATING CLEBSCH-GORDAN COEFFIECENT
!    UNDER THE SUMMATION SIGN
!! --------------------------------------------------- !!

       k_min=0
       if((-jj+j2-m1).gt.k_min)then
         k_min=int(-jj+j2-m1)
       endif
       if((-jj+j1+m2).gt.k_min)then
         k_min=int(-jj+j1+m2)
       endif

       k_max=int(j1+j2-jj)
       if((j1-m1).lt.k_max)then
          k_max=int(j1-m1)
       endif
       if((j2+m2).lt.k_max)then
          k_max=int(j2+m2)
       endif

!! --------------------------------------------------- !!
!     CLEBSCH-GORDAN COEFFIECENT FORMULATION
!! --------------------------------------------------- !!
```

```
       a=sqrt(  ( (2.0_dp*jj+1.0_dp)*R_Fact(jj+j1-j2)*
          R_Fact(jj-j1+j2)*R_Fact(j1+j2-jj) ) / (R_Fact(
          j1+j2+jj+1.0_dp) ) )
       b=sqrt(R_Fact(jj+mm)*R_Fact(jj-mm)*R_Fact(j1-m1)*
          R_Fact(j1+m1)*R_Fact(j2-m2)*R_Fact(j2+m2))
       c=0.0_dp
       do z=k_min,k_max
          tmp=(1.0_dp/(R_Fact(real(z,kind=dp))*R_Fact(j1+
             j2-jj-z)*R_Fact(j1-m1-z)*&
            R_Fact(j2+m2-z)*R_Fact(jj-j2+m1+z)*R_Fact(jj-
               j1-m2+z)))
          if(mod(z,2)>eps)then
            tmp=-tmp
          endif
          c=c+tmp
        enddo
        d = a*b*c

     endif
        d=segno*d

END SUBROUTINE Calculate_one_CG


!! --------------------------------------------------- !!
! CONDITIONS FOR THE NON-OCCURANCE OF CLEBSCH-GORDAN
!    COEFFIECENT
!! --------------------------------------------------- !!

FUNCTION CHECK(m1,m2,j1,j2,mm,jj) RESULT(test)
real(kind=dp)    :: m1,m2,mm,j1,j2,jj
LOGICAL :: test

   test=.TRUE.
   if(abs(mm)>jj) then
        test=.FALSE.
   elseif (MOD((j1-j2-mm),1.0_dp) >eps) then
        test=.FALSE.
   elseif (MOD((j1-j2+mm),1.0_dp) >eps) then
        test=.FALSE.
   elseif (abs(m1+m2-mm) > eps) then
        test=.FALSE.
   endif

END FUNCTION CHECK


!! --------------------Factorial--------------------- !!
!    Function to calculate factorials resursively
```

```fortran
!! --------------------------------------------------- !!

RECURSIVE FUNCTION R_Fact(R)  RESULT(Fact)
real(kind=dp), INTENT(IN) :: R
real(kind=dp) :: Fact


IF (R < 1.0_dp) THEN
   Fact = 1.0_dp
ELSE
   Fact = R * R_Fact(R-1.0_dp)
END IF

END FUNCTION R_Fact




END MODULE TOOLS
```

Listing 3: Clebsch-Gordan Pseudocode.

A.6 LAPACK

LAPACK: Is an additional library for performing Singular value Decomposition (SVD).

```
!------------------------------------------------------!
!Calls the LAPACK diagonalization subroutine DSYEV    !
!input:  H(n,n) = real(kind=dp) symmetric matrix to be
   diagonalized!
!             n  = size of a                           !
!output: states(n,n) = orthonormal eigenvectors of a  !
!   energies(n) = eigenvalues of a in ascending order !
!------------------------------------------------------!

subroutine get_spin_ladder(H,energies,states,n)
    real(kind=dp), dimension(n,n), intent(in) :: H
    real(kind=dp), dimension(n,n), intent(out) :: states
    real(kind=dp), dimension(n), intent(out) :: energies
    integer, intent(in) :: n

    integer l,inf
    real(kind=dp)  work(n*(3+n/2))

 l=n*(3+n/2)
 states=H
 call dsyev('V','U',n,states,n,energies,work,l,inf)
```

Listing 4: SVD (LAPACK) Pseudocode .

A.7 Spin Hamiltonian Matrix

```
MODULE SPIN_H

USE PARAMETERS
USE TOOLS

IMPLICIT NONE

PUBLIC

real(kind=dp), PARAMETER, PRIVATE :: EPS = 0.0001_dp

CONTAINS

SUBROUTINE H_calculate_one_element(state_l,state_r,
   spin_array,J_list,H_value)
real(kind=dp), DIMENSION(:), intent(in) :: state_l,
   state_r
```

```
real(kind=dp), DIMENSION(:), intent(in) :: spin_array
real(kind=dp), DIMENSION(:), intent(in) :: J_list
real(kind=dp),intent(out)               :: H_value

real(kind=dp),dimension(size(J_list))    :: H_pairs
real(kind=dp), dimension(4)             :: pair_l,pair_r
real(kind=dp)                           :: Delta_Dirac
INTEGER :: num_spin,num_pair
INTEGER :: i,j,k,k_pair

num_spin=size(spin_array)
num_pair=size(J_list)
H_pairs(:)=0.0_dp

!! ---- Decompose Hamiltonian in Spin Pair terms ---- !!
!  < L |- J*CG*[(Stot^2 - s1^2 -s2^2)]| R >
!  < L |- J*CG*[(Stot^2 - s1^2 -s3^2)]| R > ...
!  < L |- J*CG*[(Stot^2 - sn^2 -sn+1^2)]| R >          !
!           Sum each term to obtain the result
!! -------------------------------------------------- !!

!! -------------------------------------------------- !!
!         CONDITION FOR SUMMATION OF THE HALMITONIAN
! <m3_l|m3_r>*...*<mn_l|mn_r>*<m1_l m2_l|- J*CG*[(Stot^2
   - s1^2 -s2^2)]|m1_r m2_r>
! <m2_l|m2_r>*...*<mn_l|mn_r>*<m1_l m3_l|- J*CG*[(Stot^2
   - s1^2 -s3^2)]|m1_r m3_r> ...
! <m1_l|m1_r>*...*<mn-1_l|mn-1_r>*<m_nl mn+1_l|- J*CG*[(
   Stot^2 - sn^2 -sn+1^2)]|mn_r mn+1_r>
! Therefore  for the <L| or |R> if m3_l = m3_r implies <
   m3_l|m3_r> = 1 and 0 otherwise (Dirac delta)
!! -------------------------------------------------  !!


k_pair=0
do i=1,num_spin
 do j=i+1,num_spin
  k_pair=k_pair+1
  !Apply Orthogonality to the states not involved in this
      pair
  Delta_Dirac=1.0_dp
  do k=1,num_spin
   if((k.ne.i).and.(k.ne.j))then
     if( .not.( abs( state_l(k)-state_r(k) ).lt.EPS ) )
        Delta_Dirac=0.0_dp
    endif
  enddo
  if(Delta_Dirac.gt.EPS)then
    !S1 and S2 for a specific pair
    pair_l(1)=spin_array(i)
    pair_r(1)=spin_array(i)
    pair_l(2)=spin_array(j)
    pair_r(2)=spin_array(j)
    !M1 and M2 for that specif pair
    pair_l(3)=state_l(i)
    pair_r(3)=state_r(i)
    pair_l(4)=state_l(j)
    pair_r(4)=state_r(j)


    Call calculate_spinpair_coupling(pair_l,pair_r,J_list
       (k_pair),H_pairs(k_pair))
  endif
 enddo
enddo

H_value=0.0_dp
do k_pair=1,num_pair
 H_value = H_value + H_pairs(k_pair)
enddo



CONTAINS

SUBROUTINE calculate_spinpair_coupling(pair_l,pair_r,
   J_coupling,val_out)
real(kind=dp), DIMENSION(:), intent(in) :: pair_l,pair_r
real(kind=dp), intent(in)    :: J_coupling
real(kind=dp), intent(out)   :: val_out

real(kind=dp), dimension(1000)  :: jj_pair_l,CG_pair_l,
   jj_pair_r,CG_pair_r
Integer :: size_pair_l,size_pair_r
Integer :: i,j
real(kind=dp)    :: JSpin_Square,S1_Square,S2_Square
real(kind=dp)    :: M_l,M_r

val_out=0.0_dp

!! ------------------------------------------------- !!
!                  < state_l| H |state_r>               !
!  <s1 m1,s2 m2, ... ,sn mn| H |s1 m1,s2 m2, ... ,sn mn>
! H = -2 J*[(s1s2)+(s1s3)+ ... +(s1sn)+(s2s3)+ ... +(snsn
   +1)] !
! H = - J*CG*[(Stot^2 - s1^2 -s2^2)+ ... +(Stot^2 - sn^2
   -sn+1^2)] !
```

```
!  <s1 m1,s2 m2, ... ,sn mn| - J*CG*[(Stot^2 - s1^2 -s2
   ^2)+ ... +(Stot^2 - sn^2 -sn+1^2)] |s1 m1,s2 m2, ... ,
   sn mn>  !
!! -------------------------------------------------- !!


!! -------------------------------------------------- !!
!CONDITIONS FOR THE HAMILTONIAN WITH THE COMBINATION OF!
!      THE STATES (<L|R>) = <S_l,M_l|S_r,M_r>         !
!IF  M_l=M_r and S_l=S_r  then <S_l,M_l|S_r,M_r> = 1 !
!! -------------------------------------------------- !!


M_l=pair_l(3)+pair_l(4)
M_r=pair_r(3)+pair_r(4)
if(abs(M_l-M_r).lt.EPS)then
 Call Calculate_CG(pair_l,jj_pair_l,CG_pair_l,size_pair_l
    )
 Call Calculate_CG(pair_r,jj_pair_r,CG_pair_r,size_pair_r
    )
 do i=1,size_pair_l
   do j=1,size_pair_r
     if(abs(jj_pair_l(i)-jj_pair_r(j)).lt.EPS)then
        JSpin_Square=jj_pair_l(i)*(jj_pair_l(i)+1.0_dp)
        S1_Square=pair_l(1)*(pair_l(1)+1.0_dp)
        S2_Square=pair_l(2)*(pair_l(2)+1.0_dp)

        val_out=val_out - J_coupling * CG_pair_l(i)*
           CG_pair_r(j) * (JSpin_Square-S1_Square-
           S2_Square)
     endif
   enddo
 enddo

endif

END SUBROUTINE calculate_spinpair_coupling

END SUBROUTINE H_calculate_one_element


END MODULE SPIN_H
```

Listing 5: Spin Hamiltonian Pseudocode.

A.8 TOTAL SPIN MATRIX

```
SUBROUTINE S2_calculate_one_element(state_l,state_r,
   spin_array,S2_tot)
real(kind=dp), DIMENSION(:), intent(in) :: state_l,
   state_r
real(kind=dp), DIMENSION(:), intent(in) :: spin_array
real(kind=dp),intent(out)               :: S2_tot

real(kind=dp)                          :: S2_tot_pair
real(kind=dp), dimension(4)            :: pair_l,pair_r
real(kind=dp)                          :: Delta_Dirac

INTEGER :: num_spin
INTEGER :: i,j,k

num_spin=size(spin_array)

!! TOT_S2 = SUM S2_i  + 2 * SUM (S_i*S_j)
!! TOT_S2 = SUM S2_i  +  SUM (S2_tot - S2_i - S2_j)
!! First Term Only for diagonal elements
!! Second Term Similar to the Hamiltonian
!!
!! ------ Decompose in Spin Pair terms ----- !!
!     < L | Stot^2 | R >
!  < L | Stot(Stot+1) | R >
!! ----------------------------------------------- !!

!! ----------------------------------------------- !!
!        CONDITION FOR SUMMATION OF THE SECOND TERM
! <m3_l|m3_r>*...*<mn_l|mn_r>*<m1_l m2_l| Stot^2 |m1_r
   m2_r>
! <m2_l|m2_r>*...*<mn_l|mn_r>*<m1_l m3_l| Stot(Stot+1) |
   m1_r m3_r> ...
! <m1_l|m1_r>*...*<mn-1_l|mn-1_r>*<m_nl mn+1_l| Stot(Stot
   +1) |mn_r mn+1_r>
! Therefore  for the <L| or |R> if m3_l = m3_r implies <
   m3_l|m3_r> = 1 and 0 otherwise (Dirac delta)
!! ----------------------------------------------- !!

S2_tot=0.0_dp

do i=1,num_spin
 do j=i+1,num_spin
  !Apply Orthogonality to the states not involved in this
      pair
  Delta_Dirac=1.0_dp
  do k=1,num_spin
   if((k.ne.i).and.(k.ne.j))then
```

```
     if( .not.( abs( state_l(k)-state_r(k) ).lt.EPS ) )
        Delta_Dirac=0.0_dp
    endif
  enddo
  if(Delta_Dirac.gt.EPS)then
    !S1 and S2 for a specific pair
    pair_l(1)=spin_array(i)
    pair_r(1)=spin_array(i)
    pair_l(2)=spin_array(j)
    pair_r(2)=spin_array(j)
    !M1 and M2 for that specific pair
    pair_l(3)=state_l(i)
    pair_r(3)=state_r(i)
    pair_l(4)=state_l(j)
    pair_r(4)=state_r(j)
    S2_tot_pair=0.0_dp
    call calculate_spinpair_coupling(pair_l,pair_r,
       S2_tot_pair)
    S2_tot = S2_tot + S2_tot_pair
  endif
 enddo
enddo

! IF it is a diagonal term
Delta_Dirac=1.0_dp
do i=1,num_spin
     if( .not.( abs( state_l(i)-state_r(i) ).lt.EPS ) )
        Delta_Dirac=0.0_dp
enddo
if(Delta_Dirac.gt.EPS)then
  do i=1,num_spin
    S2_tot=S2_tot+(spin_array(i)*(spin_array(i)+1))
  enddo
endif



CONTAINS

SUBROUTINE calculate_spinpair_coupling(pair_l,pair_r,
   S2_tot)
real(kind=dp), DIMENSION(:), intent(in) :: pair_l,pair_r
real(kind=dp), intent(out)   :: S2_tot

real(kind=dp), dimension(1000)  :: jj_pair_l,CG_pair_l,
   jj_pair_r,CG_pair_r
Integer :: size_pair_l,size_pair_r
Integer :: i,j
real(kind=dp)    :: M_l,M_r
real(kind=dp)     :: S1_Square,S2_Square,JJ_Square

S2_tot=0.0_dp


!! ----------------------------------------------- !!
!                  < state_l| Stot^2 |state_r>        !
!  <s1 m1,s2 m2, ... ,sn mn| Stot^2 |s1 m1,s2 m2, ... ,sn mn>
!               Second Term
! Stot^2 = 2*[(s1s2)+(s1s3)+ ... +(s1sn)+(s2s3)+ ... +(snsn+1)] !
! Stot^2 = CG*[(Stot^2 - s1^2 -s2^2)+ ... +(Stot^2 - sn^2 -sn+1^2)] !
!  <s1 m1,s2 m2, ... ,sn mn|  CG*[(Stot^2 - s1^2 -s2^2)+ ... +(Stot^2 - sn^2 -sn+1^2)] |s1 m1,s2 m2, ... ,sn mn>  !
!! ----------------------------------------------- !!


!! ----------------------------------------------- !!
!CONDITIONS FOR THE SPIN TOTAL WITH THE COMBINATION!
!  OF THE STATES (<L|R>) = <S_l,M_l|S_r,M_r>        !
!IF  M_l=M_r and S_l=S_r then <S_l,M_l|S_r,M_r> = 1
!! ----------------------------------------------- !!

M_l=pair_l(3)+pair_l(4)
M_r=pair_r(3)+pair_r(4)
if(abs(M_l-M_r).lt.EPS)then
 call Calculate_CG(pair_l,jj_pair_l,CG_pair_l,size_pair_l)
 call Calculate_CG(pair_r,jj_pair_r,CG_pair_r,size_pair_r)
 do i=1,size_pair_l
   do j=1,size_pair_r
     if(abs(jj_pair_l(i)-jj_pair_r(j)).lt.EPS)then
!         write(*,*)'Pair: ',i,j
!         write(*,fmt='(3X,A3,F3.1,F5.1,A3,F6.3)') " | ",jj_pair_l(i),M_l," > ",CG_pair_l(i)
!         write(*,fmt='(3X,A3,F3.1,F5.1,A3,F6.3)') " | ",jj_pair_r(j),M_r," > ",CG_pair_r(j)

        S1_Square=pair_l(1)*(pair_l(1)+1.0_dp)
        S2_Square=pair_l(2)*(pair_l(2)+1.0_dp)
        JJ_Square=jj_pair_l(i)*(jj_pair_l(i)+1.0_dp)
        S2_tot= S2_tot + CG_pair_l(i)*CG_pair_r(j) * (JJ_Square-S1_Square-S2_Square)

     endif
```

```
    enddo
 enddo
endif

END SUBROUTINE calculate_spinpair_coupling


END SUBROUTINE S2_calculate_one_element
```

Listing 6: Total Spin Pseudocode.

A.9 Spin

```
Program Spin_Ladder

USE PARAMETERS
USE TOOLS
USE SPIN_H

IMPLICIT NONE


INTEGER    :: num_spins,num_states,num_pairs
INTEGER    :: n,i,j
INTEGER    :: counter
INTEGER    ::    iostat
INTEGER, PARAMETER      :: unit_in=27,unit_out=29
real(kind=dp), PARAMETER    :: EPS=0.0001_dp
real(kind=dp),dimension(:,:), allocatable :: basis_set,
   H_matrix,S2_matrix
real(kind=dp),dimension(:),   allocatable :: spin_array
real(kind=dp),dimension(:),   allocatable :: J_coupling
real(kind=dp), dimension(:), allocatable :: state_l,
   state_r
real(kind=dp)  :: H_value,S2
character(len=100) :: FMT1


real(kind=dp),dimension(:,:), allocatable :: H_diag,
   states,S2_diag
real(kind=dp),dimension(:), allocatable :: energies,
   S2_values,Spin



!!!!!!!!!!!! DEFINE BASIS SET !!!!!!!!!!!!!!!!!!!!!!!!!!!!
write(*,'(a)',advance='no')'Defining BasisSet...
                  '
```

```
num_states=1
do i=1,num_spins
    num_states = num_states * int(2.0_dp * spin_array(i)
       + 1.0_dp)
end do


!allocate(basis_set(2*num_spins,num_states))
allocate(basis_set(num_spins,num_states))
basis_set(:,:)=0.0_dp

Call get_Basis_Set(spin_array,basis_set)

open(unit=unit_out, file="Basis_Set.out", action="write",
    status="replace")

write(unit_out,'(a)')'#       Basis Set'
write(unit_out,*)
write(FMT1,"(a,I2,a)") '(I3,a,',(2*num_spins),'F6.2)'
do i=1,num_states
    write(unit_out,FMT1) i,': ',spin_array(:),basis_set
       (:,i)
end do
close(unit_out)
write(*,*)'Done!'
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!!!!!!!!!!!!!! CALCULATE HAMILTONIAN MATRIX !!!!!!!!!!!
write(*,'(a)',advance='no')'Calculating Spin Hamiltoninan
   ...   '
allocate(state_l(num_spins),state_r(num_spins))
state_l(:)=0.0_dp
state_r(:)=0.0_dp
Allocate(H_matrix(num_states,num_states))
H_matrix(:,:)=0.0_dp

do i=1,num_states
    do j=1,num_states
        state_l(:)=basis_set(:,i)
        state_r(:)=basis_set(:,j)
        call H_calculate_one_element(state_l,state_r,
           spin_array,J_coupling,H_value)
        H_matrix(j,i)=H_value
    end do
end do

open(unit=unit_out, file="SpinHamiltonian.out", action="
   write", status="replace")
write(unit_out,'(a)')'#     HAMILTONIAN MATRIX'
write(unit_out,*)
call write_matrix(H_matrix,6,unit=unit_out)
close(unit_out)
write(*,*)'Done!'
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
write(*,'(a)',advance='no')'Calculating Spin Ladder...
          '
allocate(H_diag(num_states,num_states))
allocate(states(num_states,num_states))
allocate(energies(num_states))
H_diag(:,:)=0.0_dp
energies(:)=0.0_dp
states(:,:)=0.0_dp

call get_spin_ladder(H_matrix,energies,states,num_states)
states=TRANSPOSE(states)

!!! TEST RESULTS
!H_diag=MATMUL(states,MATMUL(H_matrix,TRANSPOSE(states)))
!call write_matrix(H_diag,5)

open(unit=unit_out, file="EigenStates.out", action="write
   ", status="replace")
!!! PRINT EACH EIGENVECTOR as EACH COLUMN
write(unit_out,'(a)')'#          EIGENSTATES'
write(unit_out,*)
call write_matrix(states,6,unit=unit_out)
close(unit_out)

!!! PRINT EIGENVALUES
!write(*,*)'          #ENERGIES'
!do i=1,num_states
!    write(*,'(I,a,F10.6)')i,': ',energies(i)
!enddo
write(*,*)'Done!'

!!!!!!!!!!!!! CALCULATE S2 TOTAL !!!!!!!!!!!!!!
write(*,'(a)',advance='no')'Calculating Spin...
                 '
Allocate(S2_matrix(num_states,num_states))
S2_matrix(:,:)=0.0_dp

do i=1,num_states
    do j=1,num_states
        state_l(:)=basis_set(:,i)
        state_r(:)=basis_set(:,j)
        call S2_calculate_one_element(state_l,state_r,
           spin_array,S2)
        S2_matrix(i,j)=S2
    end do
end do

!write(*,*)'Spin Matrix:'
!call write_matrix(S2_matrix,5)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
allocate(S2_diag(num_states,num_states))
!allocate(states(num_states,num_states))
allocate(S2_values(num_states))
S2_diag(:,:)=0.0_dp
S2_values(:)=0.0_dp


!!! TEST RESULTS
S2_diag=MATMUL(states,MATMUL(S2_Matrix,TRANSPOSE(states))
   )
!call write_matrix(S2_diag,5)

do i=1,num_states
 S2_values(i)=S2_diag(i,i)
enddo

!!! PRINT EACH EIGENVECTOR as EACH COLUMN
!states=TRANSPOSE(states)
!call write_matrix(states,5)
!!! PRINT EIGENVALUES
!write(*,*)'           S^2'
!do i=1,num_states
!   write(*,'(I3,a,F10.6)')i,': ',Spin_values(i)
!enddo
!write(*,*)
!write(*,*)
!write(*,*)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!


!!!!!!!!!!!!  CALCULATE SPIN MULTIPLICITY  !!!!!!!!!!!!

allocate(Spin(num_states))
Spin(:)=0.0_dp


Spin = (- 1.0_dp + sqrt(1.0_dp + 4.0_dp*S2_values))/2.0
   _dp

open(unit=unit_out, file="SpinLadder.out", action="write"
   , status="replace")
```

```
write(unit_out,'(a)')'#STATE  ENERGY  SPIN'
write(unit_out,*)
do i=1,num_states
   write(unit_out,'(I3,a,F10.6,2X,F10.6)')i,': ',energies(i),Spin(i)
enddo
close(unit_out)
write(*,*)'Done!'

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
write(*,'(a)')'############## Code Ended #############'


End Program Spin_Ladder
```

Listing 7: Total Spin Pseudocode.

ACKNOWLEDGEMENTS

I first of all give thanks to the Almighty GOD for granting me this opportunity and strength to undertake this wonderful programme successfully.

I would also like to thank my thesis advisor Prof. Leonardo Guidoni and co-advisor Dr. Daniele Bovi for their guidance, corrections, patience and advice. For making more clear the path of research, grazie mille.

To my professors Rodolfo Repetto, Corrado Lattanzio, Pasquale Palumbo, Debora Amadori, Angelo Luongo, Alessandro D'Innocenzo, Jens Struckmeier, Winnifried Wollner, SergioGonzález-Andrade, Roberto Verzicco, for everything that I learned from them, Pierangelo Marcati at Gran Sasso Science Institute (GSSI) for allowing me to undertake several lessons with their students and Donato Pera for his technical support to use the Caliban-HPC Laboratory. Not forgetting the MathMods team (both in L'Aquila - Italy and Hamburg - Germany) coordinated by Prof. Bruno Rubino who made all this possible.

To all my friends in MathMods for their help and friendship. Thanks for these two years.

Lastly and foremost, I say a special thanks to my family, especially my mother, sister and brother back home in Ghana for their support in Prayers and all other aspect to aid me achieve my aims and also my friends.

References

[1] *Quantum Mechanics of Many-Electron Systems*, volume 123 of *A*. The Royal Society, 1929.

[2] A. H. A. Alex, M. Kalus and J. von Delft. *Mathematical Physics*, 82, 2011.

[3] P. M.-S. A. Cohen and W. Yang. *Science*, 321(792), 2008.

[4] P. Atkins. *Shriver and Atkins' Inorganic Chemistry*. Oxford University Press, 2010.

[5] A. Bencini. *Inorganic Chemistry*, 361:3820 – 3831, 2008.

[6] H. A. Bethe and Ann. *Physics*, (3):33, 1929.

[7] A. Bohm and M. Loewe. *Quantum Mechanics: Foundations and Applications,*. ISBN 0-387-95330-2. New York: Springer-Verlag, 3rd. edition, 1993.

[8] D. Bovi and L. Guidoni. *Chemical Physics*, 114107(137), 2012.

[9] F. N. C. Fiolhais and M. Marques. *A Primer in Density Functional Theory*. ISBN 3-540-03082-2. Springer-Verlag Berlin Heidelberg New York, 2003.

[10] B. C. E and L. C. Biedenharn. *Mathematical Physics*, 5(12):1723 – 1730, 1964.

[11] R. P. V. S. E. Schreiner, N. N. Nair and D. Marx. *PNAS*, 104(52):20725 – 20730, 2007.

[12] D. G. F. Neese, T. Petrenko and G. Olbrich. *Chemistry Review*, 251:288 – 327, 2007.

[13] R. H. H. H. Beinert and E. Munck. *Science*, 277(5326):653 – 659, 1997.

[14] W. Heisenberg. *Zeitschrift für Physik*, 49:619–636, 1928.

[15] Y. Jean. *Molecular orbitals of transition metal orbitals*. Oxford University Press, 2005.

[16] K. M. J. B. K. N. Ferreira, T. M. Iverson and S. Iwata. *Science*, 303(5665):1831 – 1838, 2004.

[17] R. G. Lerner and G. L. Trigg. ISBN 0-89573-752-3. VHC publishers (VHC Inc.), 1991.

[18] P. Löwdin. *Physical Review*, 97(1474), 1955.

[19] R. P. V. S. N. N. Nair, E. Schreiner and D. Marx. *Chemical Theory and Computation*, 4:1174 – 1188, 2008.

[20] V. S. N. N. Nair, J. Ribas-Arino and D. Marx. *Chemical Theory and Computation*, 6(2):569 – 575, 2010.

[21] F. Neese. *Physics and Chemistry of Solids*, 65:781 – 785, 2004.

[22] N. N. N. J. R.-A. E. S. S. A. Fiethen, V. Staemmler and D. Marx. *Physical Chemistry*, 114:11612 – 11619, 2010.

[23] R. Shankar. *Principles of Quantum Mechanics.* ISBN 978-0-306-44790-7. Kluwer Academic/Plenum Publishers., 2nd edition, 1994.

[24] B. T. Sutcliffe. *Advanced Quantum Chemistry*, 28:65, 1997.

[25] J. H. V. Vleck. *Science*, 76(1971):326 – 328, 1932.

[26] V. Vleck and J. Hasbrouck. *The Theory of Electric and Magnetic Susceptibilities.* Clarendon Press, Oxford, 1932.

[27] V. K. N. C. J. M. W. L. W. Ames, D. A. Pantazis and F. Neese. *The American Chemical Society*, 133(49):19743 – 19757, 2011.