User's Guide: Crystallographic Coupled Cluster Method

D. J. J. Farnell[†]

Department of Physics, University of Manchester
Institute of Science and Technology (UMIST),
P. O. Box 88, Manchester M60 1QD, United Kingdom

(Dated: April 4, 2003)

[†] `mccmmdf@dirac.phy.umist.ac.uk`

CONTENTS

I. ABOUT THIS GUIDE

The coupled cluster method (CCM) is a well-known and powerful method of quantum many-body theory [1–9]. It has been extensively utilised in order to predict the zero-temperature properties of lattice quantum spin systems [10–31]. This document outlines how to obtain, set up, and use a high-order CCM code for spin systems and related problems on a general "crystallographic" lattice.

This section gives an overview of the CCM and explains some of the key points of the CCM formalism. It also deals with how to download the code from the internet and how to set up the codes so that they will run on your computer for the **UNIX** operating system. In particular, the graphical user interface uses `TCL/TK` and so the "windowing shell" `wish` must be installed on your computer. The CCM codes are written in ANSI `C` and are compiled using an appropriate makefile, also provided with the code.

The rest of the document explains how to use the GUI to define and run specific problems and how to plot the results of these simulations using a tailor-made graphics utility. Advanced topics which require the user to edit the CCM `C` code and recompile (such as parallel implementations of the code, user-defined constraints on "fundamental" CCM clusters, and expectation values other than the lattice magnetisation) are also dealt with. The final 'trouble-shooting' section deals with typical errors and possible solutions to them.

A Overview of the CCM

The CCM is a well known technique of many-body theory in which the many-body ket-state wave function is written as the exponential of a ket-state correlation operator which contains purely creation operators which acts on some model or reference state. The many-body

Schrödinger equation may be solved and ground- and excited-state properties determined. A characteristic of the method is that the bra and ket states are parametrised independently, and so no strict variational upper bound on the ground-state energy is afforded. However, the Goldstone linked cluster theorem is satisfied from the outset of any calculation, and the Hellmann-Feynman theorem is also satisfied. Note that a full explanation of the CCM formalism is given in Appendix A or see for example, R.F. Bishop, *Theor. Chim. Acta* **80**, 95 (1991). However, some of the key concepts are now introduced in order to give an idea of how one goes about implementing the CCM.

1 The Model State and the Ground Ket State

The reference or model state $|\Phi\rangle$ is a state with respect to which the quantum correlations are defined. As mentioned above, the ground-state wavefunction is defined as,

$$|\Psi\rangle \;=\; e^{S}\;|\Phi\rangle \tag{1}$$

where S is the ket-state correlation operator define by

$$S = \sum_{I} \mathcal{S}_I C_I^{+} \tag{2}$$

and where the $\{C_I^{+}\}$ is a set of multi-spin creation operators and $\{\mathcal{S}_I\}$ are their corresponding CCM correlation coefficients.

For the systems under consideration here we use a *Néel-type* model state, namely, one in which the classical ground state is used as the model state and one is able to perform a unitary transformation of the spin coordinates such that the model state now consists of spin pointing purely in the downwards *z*-direction. The creation operators C_I^{+} are thus products of spin-raising operators only.

The user must do this by hand before they start the CCM and an example of how one performs this transformation for the *XXZ* model on the linear chain and for the triangular lattice is given in the 'Tutorials' below.

Future developments of the CCM will include: the use of "non-classical" model states (such as valence-bond solid model states); crystallographic SUB2; estimation of excitation spectra using high-order CCM techniques; extension of high-order CCM to lattice boson and fermion models; and, the application of the CCM to non-zero temperature.

2 *Approximation Schemes*

We note that the solution of the CCM problem in which all possible correlations are retained in S (which thus provides an exact solution) is often difficult, if not impossible, to achieve and so an approximation must be made. The three main approximation schemes (in S) are given by:

1. The SUBn scheme, in which all correlations involving only n or fewer spins are retained, but no further restriction is made concerning their spatial separation on the lattice.
2. The SUBn-m sub-approximation, in which all SUBn correlations spanning a range of no more than m contiguous lattice sites are retained.
3. The localised LSUBm scheme, in which all multi-spin correlations over distinct locales on the lattice defined by m or fewer contiguous sites are retained.

We note that the maximum number of spin-flips at any particular site with respect to our "Néel model state" (i.e., all spin pointing downwards) is given by $2 \times s$.
Furthermore, we note that the task of determining all of the "fundamental" which lie in the physically relevant spin subspace and are independent under the symmetries of both the lattice

and Hamiltonian may be determined computationally for large values of m and n. Appendix B explains how this may be achieved.

3 CCM Equations and Expectation Values

The time-independent Schrödinger equation tells use that

$$H|\Psi\rangle = E_g|\Psi\rangle \quad \Rightarrow \quad He^S|\Phi\rangle = E_g e^S|\Phi\rangle \tag{3}$$

and thus

$$E_g|\Phi\rangle = e^{-S}He^S|\Phi\rangle \tag{4}$$

The application of $\langle\Phi|$ to this equation then leads to:

$$E_g = \langle\Phi|e^{-S}He^S|\Phi\rangle \quad . \tag{5}$$

Note E_g may be found analytically, although the high-order CCM program does this automatically and so the user need not do this "by hand." We now need to determine all of the CCM correlation coefficients, within a given approximation, in order to obtain a value, for example, for E_g. We do this by applying $\langle\Phi|C_I^-$ to the equation for $E_g|\Phi\rangle$ above where C_I^- is formed purely from spin-lowering operators in the $I^{\rm th}$ "fundamental" clusters in our approximation. Thus our ket-state equations are given by:

$$\langle\Phi|C_I^- e^{-S}He^S|\Phi\rangle = 0 \quad . \tag{6}$$

This is determined and solved computationally here and Appendix B explains how this is achieved. Appendix B also explains how one may determine the bra state once the ket

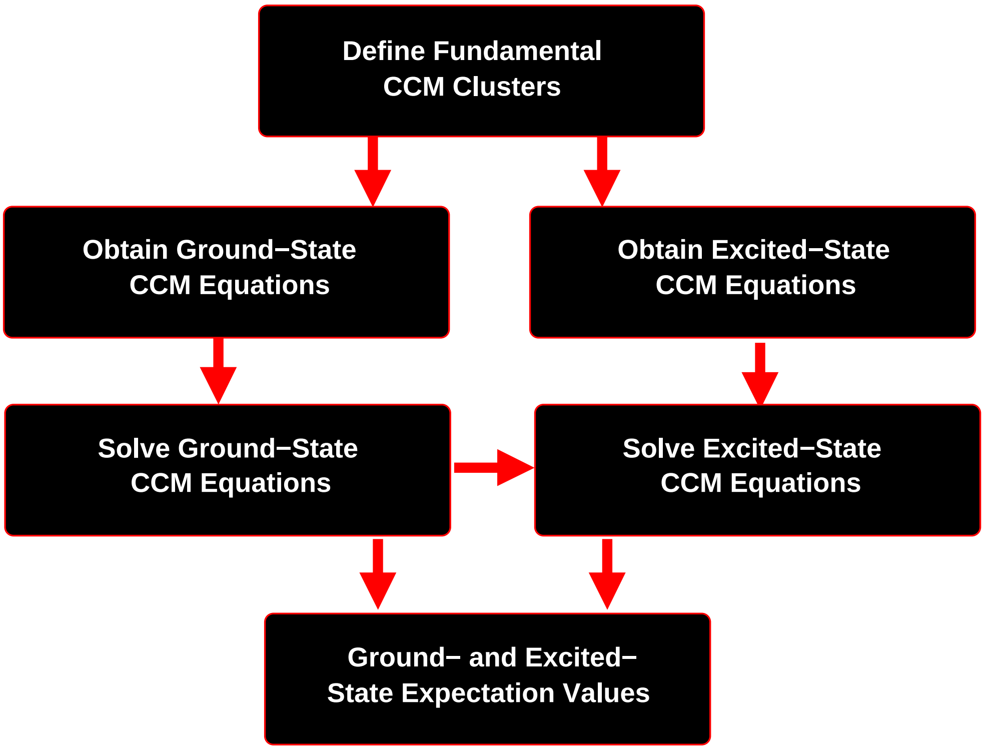

Define Fundamental CCM Clusters
Obtain Ground−State CCM Equations
Obtain Excited−State CCM Equations
Solve Ground−State CCM Equations
Solve Excited−State CCM Equations
Ground− and Excited− State Expectation Values

FIG. 1: The CCM algorithm.

state has been determined and one is therefore able to also obtain other expectation values also.

We note that the results for the ground-state energy, the amount of lattice or sublattice magnetisation, low-lying excitation energies, positions of quantum phase transition point (and in some cases the order of the phase transition), and (more recently) the spin-stiffness have previously been found to be among some of the best results as yet determined for many different lattice quantum spin models. The CCM results for the ground and excited states and for the order parameters of these models allow us to make strong statements regarding the nature of the quantum ordering and the type and position of phase transitions with respect to the phase diagram.

4 The CCM Algorithm

The CCM algorithm is split into distinct parts, as shown in the Fig. 1. Firstly, we must determine the 'fundamental' CCM clusters for a given approximation. These are fundamental because they are not equivalent under the symmetries of the lattice and have the correct spin symmetry (e.g., $s_T^z = \sum_i s_i^z = 0$). The CCM equations for each of these fundamental clusters is then determined (for the ground-state and/or excited states) and then the resulting coupled non-linear polynomial ground-state equations are solved. The excited-state equations form an eigenvalue problem [26] which may only be solved once the ground-state equations have been solved, although note that this implementation does not include excited states at this time.

A full account of how the CCM fundamental clusters are determine and how the CCM equations are obtained for the high-order CCM approach is given in Appendix B. All further details of the computational algorithms used in high-order CCM approach is thus also deferred until Appendix B.

B Definition of Terms

A number of ideas must now be introduced which are crucial to an implementation of the CCM using this code.

1 The Hamiltonian and Model State

To re-iterate: The model state in this implementation is made up of spins which all point in the downwards z-direction. The user must cast their problem for a given Hamiltonian with a suitable choice of untransformed model state into one in which all of the spins points downwards via a simple unitary transformation of the spin coordinates (perhaps on given sublattices).

The Tutorials shows how this may be done although all of the References [10–31] also give details of how one achieves this for a variety of cases.

2 The Script File

A crucial point is that this implementation uses CCM script files which are interpreted by the CCM executable, in order to determine and solve the CCM equations, and which fully defines the particular physical problem. The graphical user interface in large party deals with the manipulation of these script files. Note however that a particular CCM run will probably have a different name to the CCM script file name. Indeed, this CCM 'simulation' or 'run' name is actually defined *within* a given CCM script file. Thus, in order to change the approximation level and all output files associated with this approximation one can change a few parameters within the script file *without* having to write a whole new script from scratch.

3 The Simulation Name

A particular simulation (e.g., for a given level of approximation) is denoted by a file name which does need to be (and probably shouldn't be) the same as the CCM script name. In fact, the simulation or run name is defined *within* the CCM script file. Different simulations as thus easily run by minimal changes to the CCM script.

4 The Hamiltonian

Note that we split the Hamiltonian into two parts, H_1 and H_2 and that the first such part, H_1 is multiplied by a factor x which may be varied from some predefined starting value to a finishing value in constant increments. The factor for the second such part is unity.

$$H \; = \; xH_1 \; + \; H_2 \tag{7}$$

Thus, we choose an initial value for x for which we know that our initial values for the CCM correlation coefficients (*all initially taken to be zero!*) will converge using the Newton-Raphson technique or direct iteration of CCM equations. Often, we take the second such part H_2 to be one for which the model state is an exact eigenstate such that all of the CCM correlation coefficients must be zero if we set out starting value for x is also zero. More of this is said for the *XXZ* and triangular models etc. in the tutorials below.

Note that each part in H may have many different terms within it and indeed each "term" may have its own unique coefficient such that, for example, H_1 might be given by something like $H_1 = -0.5 \sum_i (s_i^+ s_{i+1}^+ + s_i^- s_{i+1}^-) + 0.125 \sum_i \sum_i (s_i^- s_{i+1}^+ + s_i^+ s_{i+1}^-)$. In this case there are four such terms and two have a factor of -0.5 and two have a factor of $+0.125$.

Thus, the ideas of splitting up the Hamiltonian in the way serves only to track our solution from some known point (often for which all of the CCM correlation coefficients are zero anyway) into the region of interest. Often the solution to the CCM equations will "break down" at some point (often evinced, for example, by the second derivative of the ground-state energy and the lattice magnetisation diverging) as one increases the parameter x. For the sake of simplicity, only two such parts are allowed here although the present implementation of the `C` code allows arbitrary numbers of such "parts".

5 *Crystallographic Point- and Space Groups*

The CCM code needs to know about the nature of the Hamiltonian and the underlying lattice in order to work. The information about the underlying lattice is used in order to define and identify multi-spin clusters which are formed from products of spin operators (which of course act on the model state). The information about the Hamiltonian is used in order to obtain the CCM equations and to obtain expectation values.

Thus, it is crucial to be able to define the primitive unit cell of an arbitrary lattice as well as the underlying Bravais lattice. The CCM code and GUI allows the user to input the relevant crystallographic lattice with corresponding symmorphic or non-symmorphic point-group symmetries.

The lattice nearest-neighbours, which define all of the points on the lattice, are thus defined with respect to each site within the unit cell respectively as a translation along the Bravais lattice vectors to a given site within the new unit cell.

Similarly, terms within the Hamiltonian "parts" are defined in a similar manner. Each site within the unit cell connects to Hamiltonian neighbours (i.e., other sites for a given interaction) by multiples of the Bravais lattice vector and a flag saying which site is to be identified in the new unit cell.

C Installation of the CCM Code

1 *Downloading the Code*

The full codes for the alpha version of the `TCL/TK` GUI and the high-order CCCM code can be obtained from:

`http://www.phy.umist.ac.uk/~mccmmdf/Programs/beta_CCCM/ccm.tar.gz`

Enter this as the "location" in your web browser and save to an appropriate directory which is empty. The choice of this directory is important because it becomes the *root directory* for the CCM suite of programs.

Now type in:

```
tar xvfz ccm.tar.gz
```

or (if this doesn't work on your machine)

```
gunzip -c ccm.tar.gz | tar -xvf -
```

or alternatively:

`gunzip ccm.tar.gz` and then `tar xvf ccm.tar`

You should now see five new directories, namely,

- *./include/* This directory contains all header files.
- *./documentation/* This directory contains all relevant documentation, including this user guide.
- *./Data/* The directory stores of the input and output data files.
- *./bin/* This directory stores the `C` code of the high-order CCCM program.
- *./GUI/* This directory stores the `TCL/TK` code of the GUI. *Always run the ccm graphical user interface from this directory!*

2 *Compiling the* `C` *Code*

The `C` code in directory *./bin/* from the root directory is compiled by typing in `make`. The `Makefile` uses the Gnu ANSI `C` compiler, namely, gcc. If you do not have this then change `gcc` in the Makefile for `cc` or your usual `C` compiler. If errors in compilation still occur then contact your system administrator to check that all libraries (the code uses the stdio, string, and math etc. libraries) are correctly installed. If the compilation procedure is successful you should now see the file `ccm.x` in this directory.

3 Configuring the User Interface

In order configure the GUI now change directory to *./GUI/* from the root directory. Now type in:

```
which wish
```

If the response is *no wish in ...* then contact your system administrator in order to install `wish`. If the response is something like */usr/bin/wish* then check that this is specified in the first lines of the programs `plot.tcl`, `ccm`, and `edit.tcl`, namely:

```
#!/usr/bin/wish -f
```

If the path for `wish` at the start of these scripts is different to that supplied by the "`which wish`" command then change it in each file accordingly. Now type in `ccm` at the command line and the main CCM panel should appear.

You are now ready to use the **Crystallographic Coupled Cluster Method**! *Always run the ccm graphical user interface from the ./GUI/ directory!*

D The Output File

The output file is stored in a file with `.output` identifier in the directory specified in the CCM script file. The format of the output is the following (enumerated by column number):

1. Value of x, the control parameter.
2. Ground-state energy per spin, E_g/N.
3. Magnetisation (in transformed, 'rotated', local spin coordinates) over ALL lattice sites.
4. Second derivative of E_g/N with respect to x determined computationally.
5. The value of the first ket-state correlations coefficient.
6. Number of iterations for the ket-state.
7. Number of iterations for the bra-state (for direct iteration only – it is set to zero for Newton-Raphson...)
8. The next N_{UC} columns contain the sublattice magetisation evaluated on N_{UC} individual sublattices which correspond to those N_{UC} sites in the unit cell. If the number of sites in the unit cell is one ($N_{UC} = 1$) then this is, of course, equal to the sublattice magnetisation evaluated on all sites and so this column(s) is omitted.

II. TUTORIALS

A The Transverse Ising Model for the Linear Chain

The Hamiltonian for the transverse Ising model is given by

$$H = -4 \sum_{\langle i,j \rangle} s_i^z s_j^z - 2x \sum_i s_i^x \quad = -4 \sum_{\langle i,j \rangle} s_i^z s_j^z - x \sum_i (s_i^+ + s_i^-) \quad , \tag{8}$$

where the usual Pauli spin σ-operators are replaced by fully equivalent spin operators for $s = 1/2$ and $\langle i, j \rangle$ indicates that each of the N nearest-neighbour bonds on the linear chain is counted once only. We note that the classical ground-state of the Ising model for negative factor outside the sum is one in which all spins point in the same direction along the z-axis and we use this state directly without having to perform any rotations of the local spin coordinates. Thus we may go ahead and use the CCM GUI.

Type in `ccm isingLinearChain.ccm` at the command line in the *./GUI/* subdirectory. The main CCM panel should now appear and the script should read as `isingLinearChain.ccm`. Click on `Edit Script` so that this panel now appears and again the script field should say `isingLinearChain.ccm`. *Always run the ccm graphical user interface from the ./GUI/ directory!*

The path should indicate the root directory for the CCM suite and the name of the simulation or run should be indicated in the field after `Run Name:` as `ising_1D_sub8-8`. Change this field so that it now reads `ising_1D_sub2-2`. Set the `output level` to be `1` and ensure that the spatial dimension and spin quantum number are correct, namely, `1` and `0.5` respectively, in the fields marked `Dimension (1-3)` and `Spin (0.5,1.0,...)`.

Now change the values of n and m in the fields marked `SUBn-m n=` and `SUBn-m m=` to 2 and 2, respectively. Ensure that the blue pull-down menu marked `Restrict Clusters` is set to `No Restriction`. Ensure that the minimum and maximum values for x are `0.0` and `2.0` respectively and that `Increment x` is a positive small number. Now ensure that the minimum and maximum values for y are again `-2.0` and `-1.0` respectively.

Check that the Bravais lattice vectors are given by

$$\vec{a} = (1.0, 0.0, 0.0) \ ; \vec{b} = (0.0, 1.0, 0.0) \text{ and } \vec{c} = (0.0, 0.0, 1.0) \quad , \tag{9}$$

that the `# of Sites in Unit Cell` in blue is set to one, and that the position of this site in the unit cell is at `x=0.0, y=0.0, z=0.0` and that the spin quantum number of this site is set at `s=0.5`.

Define the lattice nearest-neighbours to the single site in the unit cell in the line directly beneath the one with `# of Sites in Unit Cell` in blue. Thus, on the left of this new line you should see `UC#1: #NNs` which should also be in blue. Make sure that the pull-down is set to the option #nn=2. The fields after this blue pull-down menu button marked `UC#1: #NNs` define the lattice neighbours to the single site in the unit cell.

The first such neighbour is denoted by `1` and the you should see that the four fields following it holds the numbers `1 0 0 1`. This means that its first nearest neighbour is located at in the unit cell at `1` $\times\vec{a}$ `+ 0` $\times\vec{b}$ `+ 0` $\times\vec{c}$ Bravais lattice vectors from the current unit cell and that it connects to the 1-st site (of a total of one in this case!) in this unit cell.

The second such neighbour is denoted by `2` and the you should see that the four fields following it holds the numbers `-1 0 0 1`. This means that its first nearest neighbour is located at in the unit cell at `-1` $\times\vec{a}$ `+ 0` $\times\vec{b}$ `+ 0` $\times\vec{c}$ Bravais lattice vectors from the current unit cell and that it connects to the 1-st site (again of a total of one) in this unit cell.

We now set the terms in the Hamiltonian by selecting the `Site in UC`, `Term in Hamiltonian`, and `H_1 or H_2?` appropriately. Firstly, ensure that `H_1` is selected in the blue pull-down menu `H_1 or H_2?`, that the `Site in UC` is set to #`UC=1`, and that the `Term in Hamiltonian` is set to s^+. Click on `Refresh` or pressing ⟨`Return`⟩ anyway on the panel in order to "refresh" it. The entry in the fields `Coeff` should read `-1`, as defined by the Hamiltonian of Eq. (8). Note that this is a single-site term as so doesn't connect to any other lattices sites and so the pull-down menu #Ham. Neighs. should be set to zero.

By contrast, when we select `H_2` and the `Term in Hamiltonian` is set to $s^z s^z$ and again "refresh" the panel by `Refresh` or by pressing ⟨`Return`⟩ we see that there is one "Hamiltonian Neighbour." This indicates that there is a term in the Hamiltonian for which the first site (of a total one anyway) in the unit cell connects to the first site s in a unit cell which is `1` $\times \vec{a}$ (Bravais) vectors away from it. The coefficient must be given by `-4.0` for the Hamiltonian of Eq. (8).

Finally, we set the lattice symmetries by choosing the `Total Number of Symmetries` to be 2 and then selecting each symmetry in turn. Check both symmetries by typing into the `This Symmetry Operation` field firstly `1` and then 'refresh' the panel by clicking on `Refresh` or pressing ⟨`Return`⟩ anywhere in the panel. The "rotation" matrix should be the identity matrix and the non-primitive lattice translation should be `T = 0.0 0.0 0.0`. This is simply the identity symmetry operation. Now type `2` into the `This Symmetry Operation` field firstly `1` and then 'refresh' the panel by clicking on `Refresh` or pressing ⟨`Return`⟩ anywhere in the panel. The rotation matrix should now have an element of `-1.0` in the top left-hand element of the rotation matrix. This is simply the symmetry operation which corresponds to a reflection about the point $x = 0$.

Now click on the button marked `Save` and exit by pressing `Exit` if you want to. Note that this GUI does not automatically save on exit! If you want to edit a different (pre-existing only!) CCM script file at any time then enter its name into the `script` fields at the top of the panel and click on `Load`. To create a new script then change the relevant field in the main CCM panel and click on `Edit Script`.

To run a simulation now select `Define, Obtain, and Solve!` in the pull-down menu in blue menu marked `Options for Run CCM Script` and then click on the `Run Script` button. An terminal window now appears and the particular (SUB2-2 here!) run begins. The CCM equations are determined and these equations are solved over the range defined in the script file. Once the CCM program has finished successfully the user will be asked to press a key (while this terminal window) is selected in order to destroy the terminal window. The results may be viewed by now clicking on `View Results`.

B The *XXZ* Model for the Simple Cubic Lattice

The Hamiltonian for the *XXZ* Model is given by

$$H = \sum_{\langle i,j \rangle} \{ s_i^x s_j^x + s_i^y s_j^y + \Delta s_i^z s_j^z \} \tag{10}$$

where the sum on $\langle i,j \rangle$ runs over each of the $3N$ nearest-neighbour bonds on the cubic lattice, such that each bond is counted once only.

In this case we choose the Néel state to be the model state. The Néel state contains one sublattice populated entirely by spins which points upwards along the z-axis and another populated entirely by spins which points downwards along the z-axis. Each nearest-neighbour to a given sublattice site is a site on the other sublattice. In order to cast our problem into one which we may treat using the CCM code we now perform a rotation of 180° about the y-axis for the spins on the sublattice which contain only the 'up-pointing' spins. That is:

$$s^x \rightarrow -s^x \;\; ; \;\; s^y \rightarrow s^y \;\; s^z \rightarrow -s^z \tag{11}$$

The model state now appears to only contain spins which all point in the same direction along the z-axis. The Hamiltonian is now given by

$$H = - \sum_{\langle i,j \rangle} \{ \frac{1}{2} (s_i^+ s_j^+ + s_i^- s_j^-) + \Delta s_i^z s_j^z \} \tag{12}$$

where $s^{\pm} = s^x \pm \mathrm{i} s^y$. We may now use the the CCM program.

Type in `ccm xxzCubic.ccm` at the command line in the *./GUI/* subdirectory. The main CCM panel should now appear and the script should read as `xxzCubic.ccm`. Click on `Edit Script` so that this panel now appears and again the script field should say `xxzCubic.ccm`.

The path should indicate the root directory for the CCM suite and the name of the simulation or run should be indicated in the field after `Run Name:` as `ccm_cubic_sub2-2`. Set the `output level` to be `1` and ensure that the spatial dimension and spin quantum number are correct, namely, `3` and `0.5` respectively, in the fields marked `Dimension (1-3)` and `Spin (0.5,1.0,...)`.

Now change the values of n and m in the fields marked `SUBn-m n=` and `SUBn-m m=` to 2 and 2, respectively. Ensure that the blue pull-down menu marked `Restrict Clusters` is set to `No Restriction`. Ensure that the minimum and maximum values for x are `0.0` and `10.0` respectively and that `Increment x` is a negative small number. Now ensure that the minimum and maximum values for y are again `-30.0` and `0.0` respectively.

Check that the Bravais lattice vectors are given by

$$\vec{a} = (1.0, 0.0, 0.0)\ ; \vec{b} = (0.0, 1.0, 0.0) \text{ and } \vec{c} = (0.0, 0.0, 1.0) \quad , \tag{13}$$

that the `# of Sites in Unit Cell` in blue is set to one, and that the position of this site in the unit cell is at `x=0.0, y=0.0, z=0.0` and that the spin quantum number of this site is set at `s=0.5`.

Define the lattice nearest-neighbours to the single site in the unit cell in the line directly beneath the one with `# of Sites in Unit Cell` in blue. Thus, on the left of this new line you should see `UC#1: #NNs` which should also be in blue. Make sure that the pull-down is set to the option `#nn=6`. The fields after this blue pull-down menu button marked `UC#1: #NNs` define the lattice neighbours to the single site in the unit cell.

The first such neighbour is denoted by `1` and the you should see that the four fields following it holds the numbers `1 0 0 1`. This means that its first nearest neighbour is located at in the unit cell at `1` $\times\vec{a}$ `+ 0` $\times\vec{b}$ `+ 0` $\times\vec{c}$ Bravais lattice vectors from the current unit cell and that it connects to the 1-st site (of a total of one in this case!) in this unit cell. The second such neighbour is denoted by 2 and the you should see that the four fields following it holds the numbers `-1 0 0 1`. This means that its first nearest neighbour is located at in the unit cell at `-1` $\times\vec{a}$ `+ 0` $\times\vec{b}$ `+ 0` $\times\vec{c}$ Bravais lattice vectors from the current unit cell and that it connects to the 1-st site (again of a total of one) in this unit cell. The third such neighbour is denoted by 3 and you be able to should see that the four fields following it holds the numbers `0 1 0 1`. This means that its first nearest neighbour is located at in the unit cell at `0` $\times\vec{a}$ `+ 1` $\times\vec{b}$ `+ 0` $\times\vec{c}$ Bravais lattice vectors from the current unit cell and that it connects to the 1-st site (again of a total of one) in this unit cell. The rest of the neighbours follow on accordingly.

We now set the terms in the Hamiltonian by selecting the `Site in UC`, `Term in Hamiltonian`, and `H_1 or H_2?` appropriately. Firstly, ensure that `H_1` is selected in the blue pull-down menu `H_1 or H_2?`, that the `Site in UC` is set to `#UC=1`, and that the `Term in Hamiltonian` is set to $s^z s^z$. Click on `Refresh` or pressing ⟨`Return`⟩ anyway on the panel in order to "refresh" it. The entry in the all of the fields `Coeffs` should read `-1` for each 'Hamiltonian Neighbour, as defined by the Hamiltonian of Eq. (12). Check that this term connects to 6 "Hamiltonian neighbours" which are the nearest-neighbours to site 1 (of 1 in the unit cell).

When we select `H_2` and the `Term in Hamiltonian` is set to s^+s^+ or s^-s^- and again "refresh" the panel by `Refresh` or by pressing ⟨`Return`⟩ we see that there are again 6 "Hamiltonian neighbours." The coefficients in the relevant fields for each "Hamiltonian Neighbour" must be given by `-0.5` for the Hamiltonian of Eq. (12).

Finally, we set the lattice symmetries by choosing the `Total Number of Symmetries` to be `48` and then selecting each symmetry in turn. The CCM GUI should read in all of the symmetries correctly and these all correspond to the 48 different reflections and rotations of the point-group for the simple cubic lattice.

Now click on the button marked `Save` and exit by pressing `Exit` if you want to. Note that this GUI does not automatically save on exit! If you want to edit a different (pre-existing only!) CCM script file at any time then enter its name into the `script` fields at the top of the panel and click on `Load`. To create a new script then change the relevant field in the main CCM panel and click on `Edit Script`.

1 Advanced Topic: Setting the User Defined "Restriction" for CCM clusters (Example for cubic XXZ)

We note that the ground-state of this model is known to lie in the $s_T^z = \sum_i s_i^z = 0$ subspace. We start from a model for which this is true and so all CCM correlation coefficients must have an equal number of 'up' spin flips as 'down' spin flips *with respect to the original untransformed model state*. We note that after the rotation of the local spin axes all of the spins points downwards and, importantly, all spin operators are now spin-raising operators (these all mutually commute – as required by the CCM).

We restrict the clusters by choosing only those clusters which preserve $s_T^z = \sum_i s_i^z = 0$. However, we only have one site per unit cell and we have effectively "thrown away" our two sublattice structure by the transformation of the local spin axes (although it's still actually there, but hidden.) Thus, we must choose which lattice we are on by looking at the values of x-, y-, and z-components of each of the sites in the fundamental CCM clusters. We simply check if the $x + y + z$ is odd or even in order to do this for the implementation for the cubic *XXZ* model above. Thus, we may sum the $+1/2$ or $-1/2$ contributions of each site in every clusters to s_T^z in order to determine if it is a 'valid' cluster or not.

This is therefore a highly implementation-driven constraint and thus it may vary (e.g., doubling the size of the Bravais lattice vectors) drastically. There is therefore a `user-defined option` for the `Restrict Clusters` in the `Edit Script` panel which allows the user to define the own restriction. This causes the option marked `USER DEFINED OPTION!!` in the function:
`int subsetClusters(int *descriptor)`
in the program **define.C**. Now take a look at the code in this function using `emacs` or `vi` (or whatever text editor you prefer). The option for the *XXZ* model for the cubic lattice is already set up.

```
if (restrictionClusters==3) {

  // USER DEFINED OPTION!!
  // ADVANCED: USER MUST ALTER CODE IN THIS CASE!!!

  // EXAMPLE: XXZ model or Heisenberg linear-chain, square or cubic
  // lattices arbitrary ‘s’ though.

  // EXAMPLE: Ca V_4 O_9 Neel model state. Take out comment markers
  // on ‘x+=0.5;’ and ‘y+=0.5;’ and recompile

  if (flag==1) {
    stz=0.0;
    for(i=1;i<=length;i++) {
      findCartesiansFromGeneralisedPosition(x,y,z,config[i]);
      // x+=0.5;
      // y+=0.5;
      if (((int)((x+y+z)/2.0)) == ((int)((x+y+z+1.0)/2.0)))
        stz++;
      else
        stz--;
    }
    if (fabs(stz)>0.1)
      flag=0;
  }
}
```

This routine assumes that the Bravais lattice vectors are simply unit vectors along the Cartesian axes and calculates s_T^z for each cluster by looping over each site for a given cluster (indicated by the index i) and determining the coordinates of the site (stored in x, y, and z). If the sum of the components is odd then the variable stz is decreased by one and if it's even then stz is decreased by one. Once all sites have been considered, if the final value of stz doesn't equal zero then the cluster is rejected and a value of $flag$ equal to zero is returned to the main part of the `define clusters` program.

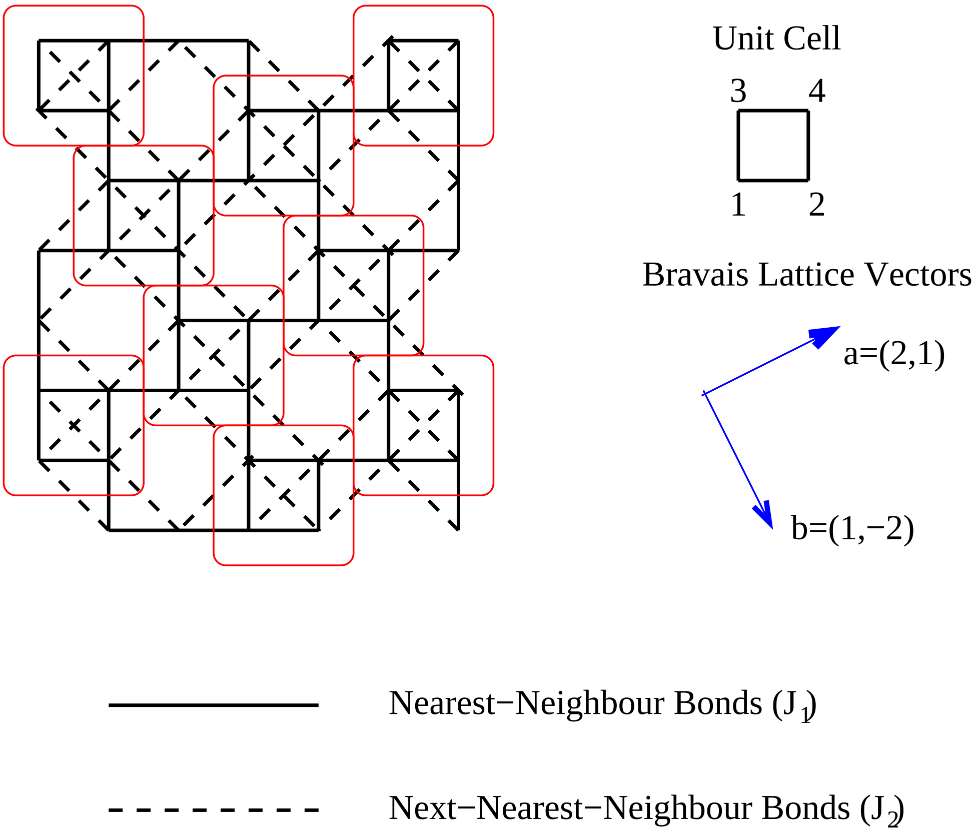

Unit Cell
3
4
1
2
Bravais Lattice Vectors
a=(2,1)
b=(1,−2)
Nearest−Neighbour Bonds (J1)
Next−Nearest−Neighbour Bonds (J2)

FIG. 2: The CaV$_4$O$_9$ Lattice.

C The J_1–J_2 Antiferromagnet for the CaV$_4$O$_9$ Lattice

By contrast to the previous examples, we must use a unit cell of four sites for the J_1–J_2 Antiferromagnet for the CaV$_4$O$_9$ Lattice. The lattice is shown in Fig. 2.

$$H = J_1 \sum_{\langle i,j \rangle} \vec{s}_i \cdot \vec{s}_j \; + \; J_2 \sum_{\langle\langle i,k \rangle\rangle} \vec{s}_i \cdot \vec{s}_k \tag{14}$$

where the sum on $\langle i,j \rangle$ runs over each of the $3N/2$ nearest-neighbour bonds on the CAVO lattice, such that each bond is counted once only, and $\langle\langle i,k \rangle\rangle$ runs over each of the $3N/2$ next-nearest-neighbour bonds on the CAVO lattice, again such that each bond is counted once only. Note again that $s = 1/2$ and that $T = 0$.

We may furthermore choose *two* model states in order to treat this model in different regimes.

1 The Neel Model State for the CAVO Lattice

The first is the Neel state in which we divide the lattice in to (nearest-) neighbouring sublattices populated with 'up' and 'down' spins, respectively. We again rotate the local spin coordinates of the 'up' spins by 180° about the y-axis (using the transformation of Eq. (11)) such that all spins now appear to lie in the downwards z-direction and the Hamiltonian is given by

$$
\begin{aligned}
H = -J_1 \sum_{\langle i,j \rangle} \{ \frac{1}{2} \{ s_i^+ s_j^+ + s_i^- s_j^- \} + s_i^z s_j^z \} \\
+ \qquad J_2 \sum_{\langle\langle i,k \rangle\rangle} \{ \frac{1}{2} \{ s_i^+ s_j^- + s_i^- s_j^+ \} + s_k^z s_k^z \} \quad . \qquad (15)
\end{aligned}
$$

Type in `ccm cavo_neel.ccm` at the command line in the *./GUI/* subdirectory. The main CCM panel should now appear and click on `Edit Script`. Check that the spatial dimension and spin quantum number are correct, namely, `2` and `0.5` respectively. *Always run the ccm graphical user interface from the ./GUI/ directory!*

In contrast to the previous examples, you should find that that the Bravais lattice vectors are given by

$$
\vec{a} = (2.0, 1.0, 0.0) \ ; \vec{b} = (1.0, -2.0, 0.0) \text{ and } \vec{c} = (0.0, 0.0, 1.0) \quad , \qquad (16)
$$

and that the `# of Sites in Unit Cell` in blue is set to four. The position of these sites in the unit cell is given by

```
x=-0.5, y=-0.5, z=0.0

x=0.5, y=-0.5, z=0.0

x=-0.5, y=0.5, z=0.0

x=0.5, y=0.5, z=0.0
```

The lattice nearest-neighbours of the sites in the unit cell should be:

```
site 1: translation=(0,0,0)  to site in new unit cell=2

        translation=(-1,0,0) to site in new unit cell=4

        translation=(0,0,0)  to site in new unit cell=3

site 2: translation=(0,0,0)  to site in new unit cell=1

        translation=(0,0,0)  to site in new unit cell=4

        translation=(0,1,0)  to site in new unit cell=3

site 3: translation=(0,0,0)  to site in new unit cell=1

        translation=(0,0,0)  to site in new unit cell=4

        translation=(0,-1,0) to site in new unit cell=2

site 4: translation=(0,0,0)  to site in new unit cell=2

        translation=(0,0,0)  to site in new unit cell=3

        translation=(1,0,0)  to site in new unit cell=1
```

where the "translation" denoted above refers to translations from the unit cell at the origin to another unit cell in terms of the Bravais lattice vectors.

It is left as an exercise to the reader to check that the Hamiltonian is correctly defined for both the next-nearest-neighbour (`H_2`) and the nearest-neighbour terms (`H_1`). (Note that J_2 is the parameter that we want to vary and that we also want to set $J_1 = 1.0$.) This follows on in exactly the same manner as for the previous examples, although we now have four sites in the unit cell. Defining the "Hamiltonian neighbours" for the Hamiltonian occurs in exactly the same way as for the lattice neighbours were above.

We set the lattice symmetries by choosing the `Total Number of Symmetries` to be 4 and then selecting each symmetry in turn. We note that there are four allowed point-group symmetries for this lattice (with this particular set-up for the unit cell). These are, namely, rotations of 0°, 90°, 180°, and 270° about the origin. The interested reader is left to check that these are correct in the supplied script.

Note again that the ground state lies in the $s_T^z = 0$ subspace. We can again check to see if this is true for each cluster by looking at its position. However, we must add on a factor of `0.5` to the lattice positions for the CAVO lattice here with the Néel state because lattice sites now lie at points for x, y, and z which are half-integers and *not* integers as for the *XXZ* model. Remove the comments (marked by //) from the following lines in `int subsetClusters(int *descriptor)`.

```
// x+=0.5;
// y+=0.5;
```

We finally note that we start from a point for J_2 which is given by $J_2 = 0.0$. We note that the initial 'guess' for the CCM coefficients (namely, that they are all zero) could be a bad one for higher orders of approximation. Thus, Newton-Raphson might fail in this case. Use a *negative* initial value for J_2 for which our initial 'guess' is a better starting point. One can also change the initial values for the CCM coefficients so that they lie with the radius of convergence (e.g., use the solution of a lower-order of approximation for those clusters common to both and set all other to zero for a particular starting value for, here, J_2). The relevant lines in `solve.C` are:

```
for(i=1;i<=NumberConfigurations;i++)
  x[i]=0.0001;
```

2 The 'Striped' Model State for the CAVO Lattice

The second model state is one in which the spins align in 'stripes.' We define this to mean here that if the coordinate x is odd then we have an 'up' spin and if its even then we have a 'down' spin. We thus again have two sublattice and we again rotate the local spin coordinates of the 'up' spins by 180° about the y-axis (using the transformation of Eq. (11))

such that all spins now appear to lie in the downwards z-direction. The Hamiltonian is given by:

$$\begin{aligned} H = & \; J_1/2 \sum_i \sum_{-\hat{x},\hat{x}} \{ \frac{1}{2} \{ s_i^+ s_{i+\hat{x}}^- + s_i^- s_{i+\hat{x}}^+ + s_i^z s_{i+\hat{x}}^z \} \\ & - J_1/2 \sum_i \sum_{-\hat{y},\hat{y}} \{ \frac{1}{2} \{ s_i^+ s_{i+\hat{y}}^+ + s_i^- s_{i+\hat{y}}^- \} + s_i^z s_{i+\hat{y}}^z \} \\ & - \quad J_2 \sum_{\langle\langle i,k \rangle\rangle} \{ \frac{1}{2} \{ s_i^+ s_j^+ + s_i^- s_j^- \} + s_k^z s_k^z \} \quad . \end{aligned} \tag{17}$$

where $\hat{x}$ and $\hat{y}$ are unit vectors in the x- and y-directions, respectively.

Type in `ccm cavo_striped.ccm` at the command line in the *./GUI/* subdirectory. The main CCM panel should now appear and click on `Edit Script`.

The setup of this problem is very similar to that for the Néel state and indeed we note that the information on the unit cell and the lattice nearest-neighbours (used in defining the fundamental clusters) is identical to that for this CAVO system using the Neel model state.

We note therefore that the main change is to the information stored on the Hamiltonian. In particular, different directions in the nearest-neighbour bonds have different types of interactions (i.e., s^-s^- and s^+s^+ for the bonds with lattice vectors $\hat{y}$ and s^+s^- and s^-s^+ for the bonds with lattice vectors $\hat{x}$). Note that this model state is a good choice in the limit $J_2 \to \infty$ and so we must set our initial value for $x(\equiv J_2)$ to be large in to reflect this and we sweep from large x to smaller x by choosing a negative value for the increment in x.

We note that only two of the rotations (of 0° and 180°) are allows and thus we have a two-point symmetry group.

Finally, replace the following lines in `int subsetClusters(int *descriptor)`.

```
// x+=0.5;
// y+=0.5;
if (((int)((x+y+z)/2.0)) == ((int)((x+y+z+1.0)/2.0)))
```

with

```
x+=0.5;
// y+=0.5;
if (((int)((x)/2.0)) == ((int)((x+1.0)/2.0)))
```

and re-compile. This code checks whether x is even or odd and sets and the values of s_i^z for $s_T^z = \sum s_i^z$ for each cluster accordingly. We note again that $s_T^z = 0$ is preserved.

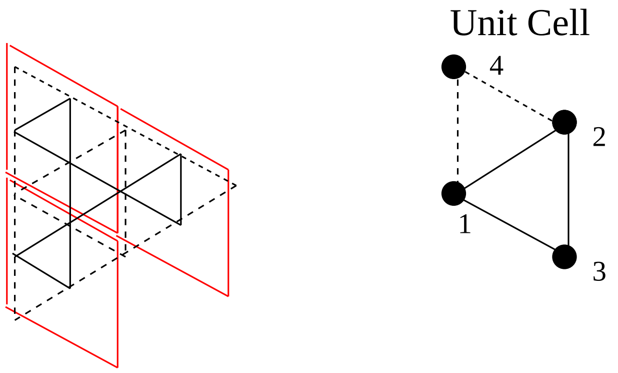


FIG. 3: The J–J' model is illustrated in diagram (a), where the bonds of strength J between Kagomé lattice sites are indicated by the thick solid lines and the non-Kagomé bonds of strength J' on the underlying triangular lattice sites are indicated by the "broken" lines.

D An Interpolating Triangular/Kagomé Lattice Antiferromagnet

The final example is also perhaps the hardest. This is for a model which interpolates between a triangular and Kagomé antiferromagnet. The Hamiltonian is given by

$$H = J\sum_{\langle i,j\rangle} \mathbf{s}_i \cdot \mathbf{s}_j + J'\sum_{\{i,k\}} \mathbf{s}_i \cdot \mathbf{s}_k \quad , \tag{18}$$

where $\langle i,j\rangle$ runs over all nearest-neighbour (n.n.) bonds on the Kagomé lattice, and $\{i,k\}$ runs over all n.n. bonds which connect the Kagomé lattice sites to those other sites on an underlying triangular lattice. Note that each bond is counted once and once only. We explicitly set $J = 1$ throughout this paper, and we note that at $J' = 1$ we thus have the triangular antiferromagnet (TAF) and at $J' = 0$ we have the Kagomé antiferromagnet (KAF).

Type in `ccm kagome_triangle.ccm` at the command line in directory *./GUI/* and `edit` the script file. This model has four sites in the unit cell at positions

#1	x=0.0	y=-1.0	z=0.0
#2	x=0.8660254	y=-0.5	z=0.0
#3	x=0.8660254	y=-1.5	z=0.0
#4	x=0.0	y=0.0	z=0.0

The underlying 2D Bravais lattice is defined by `a=(0.0,-2.0,0.0)` and `b=(1.7320508,-0.5,0.0)`. There are six point-group symmetries corresponding to rotations of 0°, 120°, and 240°, and reflections in the lines at 30° to the x-axis, at 150° to the x-axis, and in the y-axis.

For the interpolating J–J' model described by Eq. (18), we choose a model state $|\Phi\rangle$ in which the lattice is divided into three sublattices, denoted {A,B,C}. The spins on sublattice A are oriented along the negative z-axis, and spins on sublattices B and C are oriented at $+120^\circ$ and -120°, respectively, with respect to the spins on sublattice A. Our local axes are chosen by rotating about the y-axis the spin axes on sublattices B and C by -120° and $+120^\circ$ respectively, and by leaving the spin axes on sublattice A unchanged. Under these canonical transformations,

$$
\begin{aligned}
s_B^x \rightarrow -\frac{1}{2}s_B^x - \frac{\sqrt{3}}{2}s_B^z \ &; \ s_C^x \rightarrow -\frac{1}{2}s_C^x + \frac{\sqrt{3}}{2}s_C^z \ , \\
s_B^y \rightarrow s_B^y \ &; \ s_C^y \rightarrow s_C^y \ , \\
s_B^z \rightarrow \frac{\sqrt{3}}{2}s_B^x - \frac{1}{2}s_B^z \ &; \ s_C^z \rightarrow -\frac{\sqrt{3}}{2}s_C^x - \frac{1}{2}s_C^z \ . \qquad (19)
\end{aligned}
$$

The model state $|\Phi\rangle$ now appears mathematically to consist purely of spins pointing downwards along the z-axis, and the Hamiltonian (for $J = 1$) is given in terms of these rotated local spin axes as,

$$
\begin{aligned}
H = & \sum_{\langle i\rightarrow j\rangle} \Big\{ -\frac{1}{2}s_i^z s_j^z + \frac{\sqrt{3}}{4}(s_i^z s_j^+ + s_i^z s_j^- - s_i^+ s_j^z - s_i^- s_j^z) \\
& + \frac{1}{8}(s_i^+ s_j^- + s_i^- s_j^+) - \frac{3}{8}(s_i^+ s_j^+ + s_i^- s_j^-) \Big\} \\
& + J' \sum_{\{i\rightarrow k\}} \Big\{ -\frac{1}{2}s_i^z s_k^z + \frac{\sqrt{3}}{4}(s_i^z s_k^+ + s_i^z s_k^- - s_i^+ s_k^z - s_i^- s_k^z) \\
& + \frac{1}{8}(s_i^+ s_k^- + s_i^- s_k^+) - \frac{3}{8}(s_i^+ s_k^+ + s_i^- s_k^-) \Big\} \ . \qquad (20)
\end{aligned}
$$

Note that i and j run only over the N_K sites on the Kagomé lattice, whereas k runs over those non-Kagomé sites on the (underlying) triangular lattice. N indicates the total number of triangular-lattice sites, and each bond is counted once and once only. The symbol $\rightarrow$

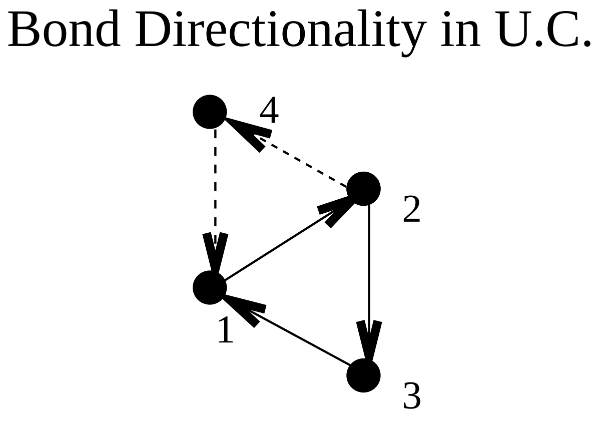
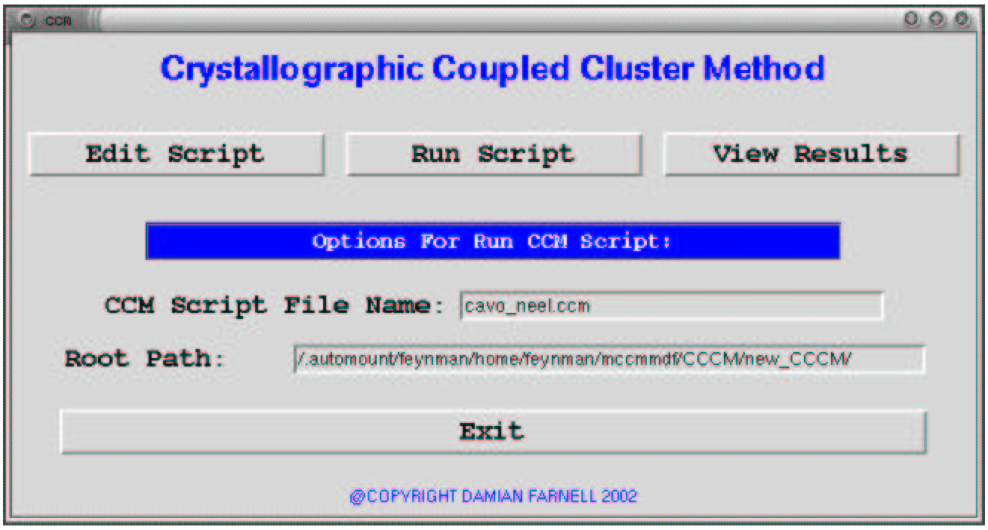


FIG. 4: Bond directionality of the J–J' model is indicated by the arrowed lines.

indicates an explicit *bond directionality* in the Hamiltonian given by Eq. (20), namely, the *three* directed nearest-neighbour bonds included in Eq. (20) point from sublattice sites A to B, B to C, and C to A for both types of bond.

We now set all of the terms which connect the non-Kagomé lattice sites to Kagomé lattice sites to be part of "H_1." Thus we also set $x = J'$ and we also set $J = 1$. We must furthermore take into account the directionality of the bonds in H. We do this by setting *Directionality?* in the `Edit Script` panel to be `Yes`. Thus, when we select any of the two-spin interaction `Terms` (e.g., $s^z s^z$) we now see that the directionality is indicated by `Directionality` → and `Directionality` ← respectively. Thus if an arrow indicated in the figure for the bond-directionalities starts at a given site in the unit cell and end at a "Hamiltonian Neighbour" then we define this term within the fields for `Directionality` →. However, if the arrow *ends* at a given site in the unit cell but starts at a "Hamiltonian Neighbour" then it goes into the fields for `Directionality` ←.

Note that the script starts the CCM solution from a point at which $x = J' = 1 (= J)$. This is therefore the pure triangular lattice. We decrement x until we reach the Kagomé point at $x = 0$.


ccm
Crystallographic Coupled Cluster Method
Edit Script
Run Script
View Results
Options For Run CCM Script:
Exit
@COPYRIGHT DAMIAN FARNELL 2002

CCM Script File Name: cavo_neel.ccm

Root Path: /.automount/feynman/home/feynman/mccmmdf/CCCM/new_CCCM/

FIG. 5: The main CCM user interface panel.

III. THE GRAPHICAL USER INTERFACE

The user interface consists of three main elements, which are namely: the main panel invoked when you type in `ccm` at the command line in the directory *./GUI/*; the `edit.tcl` panel which allows you to load, edit, and save CCM *scripts*; and a simple routine `plot.tcl` which allows the user to view the results of the CCM simulations.*Always run the ccm graphical user interface from the ./GUI/ directory!*

The rest of this section deals with the use of the GUI in order to define and run given CCM problems and how to view the results.

A The Main CCM Panel

Type in:

```
ccm cavo_neel.ccm
```

at the command line in the *./GUI/* directory. The main CCM panel, shown in Fig. 2, now appears with a CCM script file called `cavo_neel.ccm`. The main CCM panel allows the user to *edit* a script file, run the script file using the CCM program which determines the CCM equations and to solve them for a given approximation, and finally to view the results.

1 Defining the Script Name

The CCM script name may be entered in as a command-line argument, as shown above, although if no command-line argument the script file name is set to be `default.ccm`. The ccm script filename may be altered typing in a new name in the script name field. The root path is, as mentioned above, the directory in which all of the CCM subdirectories, also shown above, are placed. (This may be altered, although in most cases this will be unnecessary.)

2 Editing/Creating a Script

The script file may be edited (or indeed created) by now clicking on the `Edit Script` button. If the script does not already exist then default values (i.e., for the 1D spin-half transverse Ising model) are used and if the script does already exist then all of the information is loaded into memory and the edit script panel is opened.

3 Running a CCM Simulation for a Given Script

A script file is "run" by now clicking on the `run Script` button. Note that a number of options are available using the `Options For Run CCM Script` menu button. Clicking on this button reveals the following options:

- `Define Clusters Only` – the fundamental CCM clusters are defined only and no CCM equations are determined and/or solved. This routine reads in the CCM script file in order to get the name of this particular simulation (which not necessarily be the same as the CCM script file name!!) and creates a list of these clusters which are saved in the directory *./GUI/*. All such files are identified by the `.configs` appellation and example of such a file might be `lsub8_cavo.configs`.
- `Obtain CCM Equations Only` – determines the CCM equations for a given set of fundamental CCM clusters. This routine reads in the CCM script file in order to get the name of this particular simulation (e.g., `lsub8_cavo`) and adds the `.configs` appellation to this name and then looks for *this* file. If the file does not already exist then the routine to determine the CCM equations will fail. If the file does exist and the obtain equation routines is successful then the resulting CCM equations are stored in a file identified by the name of the particular simulation (e.g., again `lsub8_cavo`) with the appellation `.input`.
- `Solve CCM Equations Only` – solves the CCM equations for a given set of fundamental CCM clusters and given CCM equations. Again, the script file is read in order to determine the simulation name (e.g., again `lsub8_cavo`) and the corresponding `.configs` and `.input` files must already exist for the routine to work.
- `Define, Obtain, and Solve!` – this option allows you to do all of the above (in the correct order) at once.

The usefulness in this approach is that once the fundamental clusters have been defined and/or the CCM equations determined one doesn't have to re-do these parts, for example, if one runs the solve equations routine but with starting and finishing points for a tunable parameter within the Hamiltonian. Clicking on the `Run Script` button now performs the tasks in accordance with the options chosen.

The most time consuming part of the CCM calculation is generally involved in determining the CCM ground-state ket equations, although solving the CCM equations may also be time consuming – especially if one is using the "direct iteration" option for larger levels of approximation. Even determining the fundamental clusters is an exponentially increasing (and indeed open-ended) problem. (The current version should run adequately quickly up to about LSUB12 however.)

4 Plotting The Results

Clicking on the `View Results` button now again causes the script file to be read in and the particular simulation name (which we again note need not necessarily be the same as the CCM script file name) is read in. The results are stored in a file with appellation `.output` and so for simulation name `lsub8_cavo` this would read in the file `lsub8_cavo.output`. Furthermore, the initial maximum and minimum values for the x- and y-coordinates is automatically input from the script.

B The `Edit Script` Panel

The heart of the user interface in contained in the `Edit Script` panel. This panel allows one to fully define and control the CCM simulation from start to finish. Clicking on the `Edit Script` button in the main CCM panel now causes this panel to appear.

The results of this subsection describes what each option is and how it is used individually working from the top of the panel to the bottom.

1 *The CCM Script File Field*

The name of the script file can be changed at any time by changing the name in this field. Note however that this doesn't change the name of the script in the parent Main CCM panel. (Note the data in the script must be loaded from disk by clicking on the `Load` button before any changes to the information in any of the other fields takes place.)

2 *Exiting the* `Edit Script` *Panel*

The `Edit Script` panel is exited at any time by pressing the button marked `Exit`. Note that this does NOT automatically save any information that has been changed in the current CCM script file.

3 *Load a CCM Script File*

A CCM script is loaded from disk in the *./Data/* directory by clicking on the `Load` button.

4 *Save a CCM Script File*

A CCM script is saved from disk by clicking on the `Load` button. Any previously existing file of the same name in the *./Data/* directory is automatically overwritten. If you don't want this to happen then type in a different script name into the script name field, described above.

5 *Refreshing the Panel*

Note that changing certain fields and values in the panel causes the elements on the panel to change in real-time. However, the user must either click on the `Refresh` button or press ⟨`Return`⟩ on the keyboard when the `Edit Script` panel is selected for any changes to take effect.

The fields and options belonging the subject `Parameters` written in red under the button described above are now explained:

- `Work Directory` The path of the input/output created by running a script may be changed by amending this field. Note that choosing a directory on the local machine (e.g. */scratch/* or */tmp* on which the CCM process is running may quicken the speed of the CCM program by an order of magnitude, depending of course on the speed of the network. It is thus recommended that (especially for higher levels of approximation) that the "local" disk is used to store all files (e.g., often large files which contain the CCM equations. *Note that the results file (a ".`output`" file) will be stored in this directory.*
- `Run Name` This field defines the root name of of all of the input/output files associated with a given CCM simulation. This is **NOT** the same as the CCM script. Rather the 'results file' option defines the full name of the `.config`, `.input`, and `.output` files described above.
- `Output level` This determines amount of output produced by the CCM code in a given run. Choosing 0 gives a small amount of output whereas > 6 is verbose. Note that a special case of `-1` produces no output, which is useful for long background processes.
- `Dimension` Input the spatial dimension of the lattice in here (e.g., 2 for the square lattice.
- `Spin (0.5,1.0,...)` Input the spin quantum number s here, which must be a multiple of `0.5`. Note that this "global" parameter for the quantum spin number s must be equal to the largest value of s on any of the sites in the unit cell.
- `SUBn-m n=` Input the maximum number of spin flips in any cluster for a given approximation. Note that one sets $n = 2 \times s \times m$ for a spin-s system in order to define the

LSUBn approximation as one may only have a maximum of $2s$ spin flips at any particular site.

- `SUBn-m m=` This defines the "locale" of the CCM approximation. It is taken here to mean only those sites which are connected by m contiguous sites.

- `Sol. Strategy` The non-linear CCM equations may be solved via Newton-Raphson or via direct iteration, and this menu allows the user to select either of these options. Newton-Raphson is more memory intensive the direct iteration, although it is at least an order to magnitude quicker than direct iteration. Newton-Raphson is thus recommended for most approximation levels. However, for very high levels of approximation direct iteration may be the only possible solution strategy. *Note that the parallel version of the code works ONLY for direct iteration at this time.*

- `Restrict Clusters` The fundamental clusters for a given approximation are determined in the `Define Clusters` part of the CCM algorithm. However, often the ground state is known to lie within a subspace of the total Hilbert space. This option, which is a "pull-down menu" surrounded in blue, allows the user to restrict the clusters to lie within a given spin subspace. The first option is to make no restriction, whereas the second restriction allows only an even number of spin-flips in the CCM correlation operators (e.g., as for the *XXZ* with a *planar* model state [19] or the *XY* model [20]). Finally, a user-defined restriction which involves altering the CCM `C` code may also be used, for example, in order to stay in the $s_T^z = \sum_i s_i^z = 0$ subspace. More of this is said in the "Advanced" topics in the tutorials above.

7 Parameters Governing the Solution of the CCM Equations

The parameters which govern the solution of the CCM equations are now explained and this parameters are to the right of the word `Solve:` written in red in the `Edit CCM Script` panel.

As mentioned above, the prefactor for "H_1" may be varied from a starting to a finishing value in constant increments. Thus, input the minimal and maximal values of x in the relevant fields. Choosing a *positive* value for the `increment x` field means that the one sweeps from the minimal to maximal value of x and choosing a *negative* value for the `increment x` field means that the one sweeps from the maximal to minimal value of x.

The user is also asked to input maximum and minimum values for the y values (i.e., the ground-state energy, magnetisation, second-derivative of the energy, or the first ket-state CCM correlation coefficient) which is used when the `plot.tcl` program is called. Choose these parameters approximately at first and then use the supplied plotting program in order to refine your intial guesses. The user is advised to use a standard plotting packages (e.g., `xmgr` or `xmgrace`) for complicated cases and, e.g., for producing postscript figures. The supplied plotting program is only meant to be for quickly viewing CCM results only...

8 The Bravais Lattice

The Bravais lattice is uniquely defined by three vectors a, `b` and `c`. Cartesian coordinates are used to define these vectors and the so the relevant coefficients may be entered into the x-, y-, and z- components of (the 1×3 vectors) in the three fields in brackens for the three vectors a, `b` and `c` shown in the `Edit Script` panel, as required.

9 The Unit Cell

The number of sites in the unit cell is defined using the *blue* pull-down menu called `# Sites Unit Cell`. However, in this implementation only a maximum number of 6 sites in the unit cell is allowed for the sake of simplicity. This should covers most cases that the user is interested in, although in principle one can use the `C` CCM code within an arbitrary of sites in the unit although this is an "advanced" topic.

Please note that the user must click on `Refresh` or press ⟨`Return`⟩ in order to view and enable information (which may then be edited) if a *larger* number of sites in the unit cell is ever selected.

The `Edit Script` panel allows the user to input the x-, y-, and z- coordinates in the three fields clearly indicated for each site by `x=`, `y=`, and `z=`, respectively, *and* to input the value of the *quantum spin number* for this site indicated by `s=`. Thus, *ferrimagnetic* models may be treated by selecting different values for the quantum spin number on different sites within the unit cell. Note that the "global" parameter for the quantum spin number s must be equal to the largest value of s on any of the sites in the unit cell.

10 The Lattice Nearest-Neighbours

Part of the *Define Cluster* algorithm involves knowing the nearest-neighbours to each site in the unit cell. The user must now enter information about the nearest-neighbours in order to enable this. Each time the user selects a given number of sites in the unit cell, as described above, they must refresh the panel in order to display the relevant information for the lattice nearest-neighbours.

Firstly, the number of nearest-neighbours must be defined for each site in the unit cell. Each site in the unit cell (from one to six) is denoted as "`UC#1`," '`UC#2`," $\cdots$ "`UC#6`." A pull down menu in *blue* allows the user to select the number of lattice nearest-neighbours for a given site in the unit cell. Note that this therefore does not have to be the same for each site in the unit cell. The fields which describe each neighbour are "enabled" by again refreshing the panel by clicking on `Refresh` or by pressing ⟨`Return`⟩ anywhere in the `Edit Script` panel.

The nearest-neighbours are indicated by a Bravais lattice translation to a new site in one unit cell. Thus, each nearest neighbour (1, 2, $\cdots$, 8) for a given site in the unit cell is indicated by four fields which are namely $x, y, z, cell$, thus producing: $x\times$ `a` Bravais lattice vector translations (the first field); $y\times$ `b` Bravais lattice vector translations (the second field); $z\times$ `c` Bravais lattice vector translations (the third field); to a new site `cell` (the fourth and final field). (Note that certain nearest neighbours may be in the same unit cell and so x, y, and z would all be zero.)

11 Parameters for the Hamiltonian

The Hamiltonian is now defined by entering information into the fields below `Parameters for Hamiltonian` written in red in the `Edit Script` panel. However, there is often so much information for the Hamiltonian that no possible way of displaying all of the information for a given term in H is possible. Thus, the user must select a given term and then "enable" it by clicking on `Refresh` or by pressing ⟨`Return`⟩ anywhere in the `Edit Script` panel.

Firstly, the user must decide which part of the Hamiltonian they want to consider. We remember that `H_1` has a prefactor of x and `H_2` of 1. The blue pull-down menu marked `H_1 or H_2?` allows the user to select this. Secondly, each term within H_1 and H_2 is defined by selecting a site in the unit cell denoted as *Site in UC* in blue *and* an interaction type (e.g., $s^z s^z$, $s^+ s^z$, $s^- s^-$, s^z, etc.) denoted as `Term In Hamiltonian` also in blue. Again, the user must "enable" the fields within define each term within H_1 and H_2 by now clicking on `Refresh` or by pressing ⟨`Return`⟩ anywhere in the `Edit Script` panel.

Except for the interactions s^z, s^+, s^-, and $(s^z)^2$, all of the other interactions connect from one site in the unit cell (defined by the *Site in UC* pull-down menu in blue) to another site perhaps in another unit cell. These are called "`Hamiltonian Neighbours`." The number of such "Hamiltonian neighbours" is defined by the pull-down menu in blue marked #`Ham. Neighs.` and again the user is restricted to maximum of 8 of these. Again, changing information in this menu can be "enabled" by refreshing the panel.

There can be a maximum of 8 Hamiltonian neighbours and each one is represented by the appropriate number from 1 to 8 and four trailing fields. Again, these fields represent a translation of: $x\times$ `a` Bravais lattice vector translations (the first field); $y\times$ `b` Bravais lattice vector translations (the second field); $z\times$ `c` Bravais lattice vector translations (the third field); to a new site `cell` (the fourth and final field).

Note that the terms s^z, s^+, s^-, and $(s^z)^2$ apply to a single lattice site only and so never involve a lattice translation. Also, sometimes the bonds in H are "directional" and one may set the pull-down menu in blue marked `Directionality?` to `No` or `Yes` in order to define this. Directionality means that a given terms in the Hamiltonian is dependent on where it starts and finishes, and so the *direction* of this interaction is crucial. See Tutorial 4 for more information about this for the Kagome-triangular lattice Heisenberg model.

IMPORTANT Different interactions within H_1 and H_2 will probably have different coefficients. This is defined in the fields marked `Coeff` and so may be different for varying values for the blue pull-down menus `Site in the UC`, `Term in the Hamiltonian` and `H_1 or H_2?`, AND for each of the `Hamiltonian Neighbours`.

12 The Lattice Point-Group Symmetries

This is denoted by `Define Lattice Symmetries` written in red and the total number of lattice point-group symmetries is enter into the field directly after `Total Number Symmetries`. Again, there may be many such lattice symmetries and so these are dealt with one-at-time by selecting a particular symmetry operation by keying in the relevant number into the field `This Symmetry Operation`. Again, the user must "enable" all information for this symmetry operation by clicking on `Refresh` or by pressing ⟨`Return`⟩ anywhere in the `Edit Script` panel.

Symmorphic point-group operations are defined by setting the non-primitive translation vector in the three (x-, y-, and z-fields respectively) after `T=` to zero. Non-symmorphic point-group operations are defined by entering the relevant components of the non-primitive translation vector (*note that this is in terms of Cartesian coordinates and NOT components of the Bravais lattice vectors!*) into these fields for the symmetry operation defined the number in the field after `This Symmetry Operation`.

A 3×3 matrix denoted `R=` defines the "rotation" part of the point-group symmetry operation. Thus the new Cartesian coordinates (x',y',z'), with respect to the original Cartesian coordinates (x, y, z), is given by:

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} R_{1,1} & R_{1,2} & R_{1,3} \\ R_{2,1} & R_{2,2} & R_{2,3} \\ R_{3,1} & R_{3,2} & R_{3,3} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} T_x \\ T_y \\ T_z \end{pmatrix} \tag{21}$$

C The `View Results` Panel

The `View Results` Panel appears when one clicks on the `View Results` of the main CCM panel. The script currently being displayed in the main CCM panel is read in and the name of the particular simulation file with appellation `.configs` is read in. *Note that no data is read in at this point.* The initial maximum and minimum values for the x- and y-coordinates are also read in from the script file.

The maximum values of the x- and y-coordinates may be changed at any time by entering the changing the values in the relevant fields. These new values are enabled by clicking on the `Clear/Rescale` button. This also wipes out all previous plots of data. Note that this routine is supplied as a simple and quick way for the user to view their results immediately, although for more detailed and professional plots the user is urged to use a standard plotting package (such as `xmgrace`) directly on the relevant `.output` which is an ASCII file arranged in columns.

Data is read in by clicking of the button `Plot` which reads in the data of the simulation file indicated in the `File` field. Note that more data results files may be read in without deleting the existing data plotted to screen. Indeed, each new data set is given a different colour. Unfortunately, for this simple plotting routine all data is lost when the `Clear/Rescale` button is clicked on.

Different columns of the `.output` data file, which correspond to different quantities, may be read in by selecting the appropriate option in the pull-down menu in blue marked `Plot What?`.

We may now set the maximum and minimum values of the y-coordinate to be 0.5 and -0.1 respectively and click on `Clear/Rescale`. Thus, one sets the option in the pull-down menu in blue to `Magnetisation` and then replots the data.

APPENDIX A: THE CCM GROUND-STATE FORMALISM

The exact ket and bra ground-state energy eigenvectors, $|\Psi\rangle$ and $\langle\tilde{\Psi}|$, of a many-body system described by a Hamiltonian H,

$$H|\Psi\rangle = E_g|\Psi\rangle \; ; \quad \langle\tilde{\Psi}|H = E_g\langle\tilde{\Psi}| \; , \tag{A1}$$

are parametrised within the single-reference CCM as follows:

$$\begin{aligned} |\Psi\rangle &= \mathrm{e}^{S}|\Phi\rangle \; ; \quad S = \sum_{I\neq 0} \mathcal{S}_I C_I^+ \; , \\ \langle\tilde{\Psi}| &= \langle\Phi|\tilde{S}\mathrm{e}^{-S} \; ; \quad \tilde{S} = 1 + \sum_{I\neq 0} \tilde{\mathcal{S}}_I C_I^- \; . \end{aligned} \tag{A2}$$

The single model or reference state $|\Phi\rangle$ is normalised ($\langle\Phi|\Phi\rangle = 1$), and is required to have the property of being a cyclic vector with respect to two well-defined Abelian subalgebras of *multi-configurational* creation operators $\{C_I^+\}$ and their Hermitian-adjoint destruction counterparts $\{C_I^- \equiv (C_I^+)^\dagger\}$. Thus, $|\Phi\rangle$ plays the role of a vacuum state with respect to a suitable set of (mutually commuting) many-body creation operators $\{C_I^+\}$,

$$C_I^-|\Phi\rangle = 0 \quad , \quad I \neq 0 \; , \tag{A3}$$

with $C_0^- \equiv 1$, the identity operator. These operators are complete in the many-body Hilbert (or Fock) space,

$$1 = |\Phi\rangle\langle\Phi| + \sum_{I\neq 0} \frac{C_I^+|\Phi\rangle\langle\Phi|C_I^-}{\langle\Phi|C_I^- C_I^+|\Phi\rangle} \; . \tag{A4}$$

We note that although the manifest hermiticity, ($\langle\tilde{\Psi}|^\dagger = |\Psi\rangle/\langle\Psi|\Psi\rangle$), is lost in these parametrisations, the intermediate normalisation condition $\langle\tilde{\Psi}|\Psi\rangle = \langle\Phi|\Psi\rangle = \langle\Phi|\Phi\rangle \equiv 1$ is explicitly imposed. The *correlation coefficients* $\{\mathcal{S}_I, \tilde{\mathcal{S}}_I\}$ are regarded as being independent variables, even though formally we have the relation

$$\langle\Phi|\tilde{S} = \frac{\langle\Phi|\mathrm{e}^{S^\dagger}\mathrm{e}^{S}}{\langle\Phi|\mathrm{e}^{S^\dagger}\mathrm{e}^{S}|\Phi\rangle} \; . \tag{A5}$$

The full set $\{\mathcal{S}_I, \tilde{\mathcal{S}}_I\}$ thus provides a complete description of the ground state. For instance, an arbitrary operator A will have a ground-state expectation value given as

$$\bar{A} \equiv \langle\tilde{\Psi}|A|\Psi\rangle = \langle\Phi|\tilde{S}\mathrm{e}^{-S}A\mathrm{e}^{S}|\Phi\rangle = \bar{A}\left(\{\mathcal{S}_I, \tilde{\mathcal{S}}_I\}\right) \; . \tag{A6}$$

We note that the exponentiated form of the ground-state CCM parametrisation of Eq. (A2) ensures the correct counting of the *independent* and excited correlated many-body clusters with respect to $|\Phi\rangle$ which are present in the exact ground state $|\Psi\rangle$. It also ensures the exact incorporation of the Goldstone linked-cluster theorem, which itself guarantees the size-extensivity of all relevant extensive physical quantities [8].

The determination of the correlation coefficients $\{\mathcal{S}_I, \tilde{\mathcal{S}}_I\}$ is achieved by taking appropriate projections onto the ground-state Schrödinger equations of Eq. (A1). Equivalently, they may be determined variationally,

$$\delta\bar{H}/\delta\tilde{\mathcal{S}}_I = 0 \Rightarrow \langle\Phi|C_I^- \mathrm{e}^{-S} H \mathrm{e}^{S}|\Phi\rangle = 0, \ \ \forall I \neq 0 \ \ ; \tag{A7}$$

$$\delta\bar{H}/\delta\mathcal{S}_I = 0 \Rightarrow \langle\Phi|\tilde{S}\mathrm{e}^{-S}[H, C_I^+]\mathrm{e}^{S}|\Phi\rangle = 0, \ \ \forall I \neq 0 \ \ . \tag{A8}$$

Equation (A7) also shows that the ground-state energy at the stationary point has the simple form

$$E_g = E_g(\{\mathcal{S}_I\}) = \langle\Phi|\mathrm{e}^{-S} H \mathrm{e}^{S}|\Phi\rangle \ \ . \tag{A9}$$

We note that Eq. (A7) represents a coupled set of nonlinear multinomial equations for the *c*-number correlation coefficients $\{\mathcal{S}_I\}$. The nested commutator expansion of the similarity-transformed Hamiltonian,

$$\tilde{H} \equiv \mathrm{e}^{-S} H \mathrm{e}^{S} = H + [H, S] + \frac{1}{2!}[[H, S], S] + \cdots \ \ , \tag{A10}$$

together with the fact that all of the individual components of S in the sum in Eq. (A2) commute with one another, imply that each element of S in Eq. (A2) is linked directly to the Hamiltonian in each of the terms in Eq. (A10). Thus, each of the coupled equations (A7) is of linked-cluster type. Furthermore, each of these equations is of finite length when expanded, since the otherwise infinite series of Eq. (A10) will always terminate at a finite order, provided (as is usually the case) only that each term in the second-quantised form of the Hamiltonian H contains a finite number of single-body destruction operators, defined with respect to the reference (vacuum) state $|\Phi\rangle$. Therefore, the CCM parametrisation naturally leads to a workable scheme which can be efficiently implemented computationally.

The CCM formalism is exact in the limit of inclusion of all possible multi-spin cluster correlations within S and $\tilde{S}$, although in any real application this is usually impossible to achieve. It is therefore necessary to utilise various approximation schemes within S and $\tilde{S}$. The three most commonly employed schemes have been: (1) the SUBn scheme, in which all correlations involving only n or fewer spins are retained, but no further restriction is made concerning their spatial separation on the lattice; (2) the SUBn-m sub-approximation, in which all SUBn correlations spanning a range of no more than m contiguous lattice sites are retained; and (3) the localised LSUBm scheme, in which all multi-spin correlations over distinct locales on the lattice defined by m or fewer contiguous sites are retained. We also make the specific restriction that the creation operators $\{C_I^+\}$ in S preserve any additional symmetries of the Hamiltonian. Thus, the approximate CCM ground-state wave function is constrained to lie in the appropriate subspace defined by the additional quantum numbers corresponding to these additional symmetries. For the *XXZ* model illustrated later the additional symmetry is provided by the total z-component of spin $s_T^z = \sum_i s_i^z$, which commutes with the Hamiltonian. The ground state lies in the sector $s_T^z = 0$. We denote as distinct configurations those in such appropriately defined subspace which are inequivalent under the point- and space-group symmetries of both the lattice and the Hamiltonian. The number of such distinct (or fundamental) configurations for the ground state at a given level of approximation is labelled by N_F.

APPENDIX B: THE HIGH-ORDER CCM FORMALISM

In order to determine the CCM ground-state ket configurations we fundamentally need to *pattern-match* the configurations in the set $\{C_I^-\}$ to the spin-raising operators contained in $\tilde{H}|\Phi\rangle$. We note that for small values of the truncation indices $\{m, n\}$ mentioned in the previous section (and thus for low orders of approximation) this may readily be performed analytically. However, for higher orders of approximation we must use computational methods (see for example also Refs. [17, 18, 22, 26]) in order to do this. For the cases of interest here, we begin by defining a set of local spin axes in which all of the spins in the chosen model state $|\Phi\rangle$ point along the respective negative *z*-axes, namely

$$|\Phi\rangle = \bigotimes_{i=1}^{N} |\downarrow\rangle_i \,; \quad \text{in the local quantization axes,} \tag{B1}$$

where $|\downarrow\rangle_i \equiv |s, -s\rangle_i$. This is achieved by an appropriate set of local rotations. Since such rotations are canonical transformations, the underlying spin algebra is preserved, and the energy spectrum of the transformed Hamiltonian (i.e., written in the rotated local spin coordinate scheme) is unchanged.

The next step in the computational implementation of the CCM for lattice quantum spin systems of general spin quantum number, s, is to define a suitable set of multi-spin creation and destruction operators with respect to this model state. We thus define the CCM ket-state correlation operator S in terms of sums of products of single spin-raising operators, $s_k^+ \equiv s_k^x + \mathrm{i} s_k^y$, (again with respect to their local spin axes), such that

$$S = \sum_{i_1}^{N} \mathcal{S}_{i_1} s_{i_1}^+ + \sum_{i_1, i_2}^{N} \mathcal{S}_{i_1, i_2} s_{i_1}^+ s_{i_2}^+ + \cdots \,. \tag{B2}$$

The coefficients $\mathcal{S}_{i_1}$, $\mathcal{S}_{i_1,i_2}$, and so on, now represent the spin-correlation coefficients specified by the sets of site indices, $\{i_1\}$, $\{i_1, i_2\}$ and so on, on the regular lattices under consideration. We note that these indices run over *all* lattice sites, and that different indices may thus indicate the same lattice site. For the case of general spin quantum number s we note that we have a maximum number of spin-raising operators at any specific site l which is $2s_l$, where s_l is the spin quantum number of the spin situated at site l. For the spin-half case, we are thus limited to only *one* spin-raising operator per lattice site as required, and we note that in this manner we build in a previous spin-half high-order CCM formalism (see Refs. [17, 18, 22, 26]) into the new high-order formalism for general-s directly from the outset. In order to simplify the high-order CCM formalism, it is also found to be useful to define the following operators:

$$\left.\begin{array}{ll} F_k & \equiv \sum_l \ \sum_{i_2,\cdots,i_l} \ l \ \mathcal{S}_{k,i_2,\cdots,i_l} \ s^+_{i_2} \cdots s^+_{i_l} \\ G_{k,m} & \equiv \sum_{l>1} \ \sum_{i_3,\cdots,i_l} \ l(l-1) \ \mathcal{S}_{k,m,i_3,\cdots,i_l} \ s^+_{i_3} \cdots s^+_{i_l} \\ M_{k,m,n} & \equiv \sum_{l>2} \ \sum_{i_4,\cdots,i_l} \ l(l-1)(l-2) \ \mathcal{S}_{k,m,n,i_4,\cdots,i_l} \ s^+_{i_4} \cdots s^+_{i_l} \\ N_{k,m,n,p} & \equiv \sum_{l>3} \ \sum_{i_5,\cdots,i_l} \ l(l-1)(l-2)(l-3) \ \mathcal{S}_{k,m,n,p,i_5,\cdots,i_l} \ s^+_{i_5} \cdots s^+_{i_l} \end{array}\right\} \tag{B3}$$

Hence, we determine the similarity-transformed expressions of the single-spin operators s^α ; $\alpha \equiv \{+,-,z\}$, where $s^- \equiv (s^+)^\dagger = s^x - \mathrm{i}s^y$, by using Eq. (A6) and the usual spin commutation relations, such that

$$\left.\begin{array}{l} e^{-S}s^+_k e^S \equiv \tilde{s}^+_k \ = s^+_k \\ e^{-S}s^z_k e^S \equiv \ \tilde{s}^z_k \ = s^z_k + F_k s^+_k \\ e^{-S}s^-_k e^S \equiv \tilde{s}^-_k \ = s^-_k - 2F_k s^z_k - G_{kk}s^+_k - (F_k)^2 s^+_k \quad . \end{array}\right\} \tag{B4}$$

For the specific case of $s = 1/2$ at site k we note that $G_{kk} = 0$ because "double occupancy" of the lattice site k is prohibited in this case. In order to determine the similarity transformed version of a given Hamiltonian, we also need to know the commutation relations of the operators defined in Eq. (B3) with the single-spin operators s^α ; $\alpha \equiv \{+,-,z\}$, and these are stated in the Appendix.

We now define the set of CCM destruction operators $\{C^-_I\}$ (of l number of spin-lowering operators), as follows

$$C^-_I \equiv s^-_{j_1} s^-_{j_2} \cdots \ s^-_{j_l} \ , \tag{B5}$$

where the indices $j_1, j_2, \cdots, j_l$ represent any given lattice site. We choose only *one* of the $N_B(l!)\nu_I$ symmetry-equivalent configurations to *pattern-match* with the terms within $\tilde{H}$ in order to determine the CCM ket-state equations of Eq. (A7). Note that N_B is the number of

Bravais lattice sites and that, for a given cluster I, ν_I is a symmetry factor dependent purely on the point-group symmetries (and *not* the translational symmetries) of the crystallographic lattice.

The process of the enumeration of all possible fundamental clusters and the process of "pattern-matching" are both ideally suited to an efficient computational implementation, and a full description of these processes is also given in the Appendix. Furthermore, it is a simple matter to determine and solve the CCM bra-state equations, once the ket-state equations have been obtained and solved, as described in the Appendix, where we also explain the technicalities involved in obtaining ground-state expectation values.

1 Commutation Relations and The High-Order General-s CCM Formalism

In this article we present a new formalism and results for high-order ground-state CCM calculations for general spin quantum number, s, based on a model state in which all spins on the crystallographic lattice point downwards along the local z-axes. A large part of the new formalism relies on the new "high-order" CCM operators defined by Eq. (B3) and also their commutation relations with the single-spin operators in order to determine the similarity transforms of various operators, such as the Hamiltonian for example. In order to determine these commutation relations we firstly remind ourselves that the ket-state correlation operator S is given by Eq. (A2) with $C_I^+ \equiv s_{i_1}^+ s_{i_2}^+ \cdots s_{i_l}^+$ and $\mathcal{S}_I \equiv \mathcal{S}_{i_1,i_2,\cdots,i_l}$, and hence

$$S = \sum_l \sum_{i_1,i_2,\cdots,i_l} \mathcal{S}_{i_1,i_2,\cdots,i_l} \; s_{i_1}^+ s_{i_2}^+ \cdots s_{i_l}^+ \quad , \tag{B6}$$

where each of the indices $\{i_1, i_2, \cdots, i_l\}$ runs over all lattice sites with the condition that there can be no more than $2s$ of them at any particular lattice site. The usual spin commutation relations of the spin operators also apply,

$$[s_l^+, s_{l'}^-] = 2s_l^z \delta_{l,l'} \quad ; \quad [s_l^z, s_{l'}^{\pm}] = \pm s_l^{\pm} \delta_{l,l'} \quad . \tag{B7}$$

We also note that the commutation of a given operator with S must be distributive, such that

$$\begin{aligned}[s_k^\alpha, S] \; = \; & \sum_l \sum_{i_1,i_2,\cdots,i_l} \mathcal{S}_{i_1,i_2,\cdots,i_l} \Big\{ [s_k^\alpha, s_{i_1}^+] s_{i_2}^+ \cdots s_{i_l}^+ + s_{i_1}^+ [s_k^\alpha, s_{i_2}^+] s_{i_3}^+ \cdots s_{i_l}^+ \\ & + \; \cdots \; + s_{i_1}^+ s_{i_2}^+ \cdots [s_k^\alpha, s_{i_l}^+] \Big\} \; , \end{aligned} \tag{B8}$$

where $\alpha = \{z, +, -\}$. As pairs of spin-raising operators always commute, we may therefore state that $[s_k^+, S] = 0$. Furthermore, for the case of $[s_k^z, S]$ we note again that each index runs over all lattice sites, which implies that each term on the right-hand side of Eq. (B8) is equivalent and that, as there are l such terms, we may write this expression as

$$[s_k^z, S] = \sum_l \sum_{i_2,\cdots,i_l} l \; \mathcal{S}_{k,i_2,\cdots,i_l} s_{i_2}^+ \cdots s_{i_l}^+ s_k^+ = F_k s_k^+ \quad . \tag{B9}$$

Note again that the "high-order" operators such as F_k are defined by Eq. (B3). We lastly calculate the commutator $[s_k^-, S]$ in Eq. (B8), and using the basic commutation relations of Eq. (B7) we thus have

$$\begin{aligned}[s_k^-, S] \; = \; & -2 \sum_l \sum_{i_1,i_2,\cdots,i_l} \mathcal{S}_{i_1,i_2,\cdots,i_l} \Big\{ \delta_{k,i_1} s_k^z s_{i_2}^+ \cdots s_{i_l}^+ + \delta_{k,i_2} s_{i_1}^+ s_k^z s_{i_3}^+ \cdots s_{i_l}^+ \\ & + \; \cdots \; + \delta_{k,i_l} s_{i_1}^+ s_{i_2}^+ \cdots s_k^z \Big\} \; , \end{aligned} \tag{B10}$$

We now commute the operator s_k^z past the strings of spin-raising operators in Eq. (B10) using the basic commutation relations of Eq. (B7). Thus, for example,

$$\begin{aligned} s_k^z s_{i_2}^+ s_{i_3}^+ \cdots s_{i_l}^+ = & \; (\delta_{k,i_2} + \delta_{k,i_3} + \; \cdots \; \delta_{k,i_l}) s_{i_2}^+ s_{i_3}^+ \cdots s_{i_l}^+ \\ & + \; s_{i_2}^+ s_{i_3}^+ \cdots s_{i_l}^+ s_k^z \end{aligned} \tag{B11}$$

By inserting Eq. (B11) into Eq. (B10) we find that

$$\begin{aligned}[s_k^-, S] \; = \; & -2 \sum_l \sum_{i_3,i_4,\cdots,i_l} \Big(\sum_{n=1}^{l-1}\Big) \mathcal{S}_{k,k,i_3,\cdots,i_l} \; s_{i_3}^+ s_{i_4}^+ \cdots s_{i_l}^+ s_k^+ \\ & -2 \sum_l \sum_{i_2,i_3,\cdots,i_l} l \mathcal{S}_{k,i_2,\cdots,i_l} \; s_{i_2}^+ s_{i_3}^+ \cdots s_{i_l}^+ s_k^z \\ = & -G_{k,k} s_k^+ - 2 F_k s_k^z \quad , \end{aligned} \tag{B12}$$

using the definitions in Eq. (B3). We note again that for the case $s = 1/2$ the operator $G_{k,k} \equiv 0$.

By making use of the nested commutator expansion for the similarity-transformed operators [c.f., Eq. (A10)], it is now a simple matter to verify the relations in Eq. (B4), by using Eqs. (B9) and (B12).

In order to determine the similarity transform of the Hamiltonian however it is also necessary to also know the commutation relations of the single-spin operators with respect to F_k, F_k^2, $G_{k,m}$, and $M_{k,m,n}$. The proofs of these commutation relations follow a similar pattern to the proofs given above, and so we merely state them here:

$$\left.\begin{array}{lcl} [s_k^z, F_m] & = & G_{k,m} s_k^+ \ , \\ [s_k^z, G_{m,n}] & = & M_{k,m,n} s_k^+ \ , \\ [s_k^z, F_m^2] & = & 2F_m G_{k,m} s_k^+ \ , \\ [s_k^-, F_m] & = & -2G_{k,m} s_k^z - M_{k,k,m} s_k^+ \ , \\ [s_k^-, F_m^2] & = & -2G_{k,m}^2 s_k^+ - 2F_m M_{k,k,m} s_k^+ - 4F_m G_{k,m} s_k^z \ , \\ [s_k^z, M_{m,n,p}] & = & N_{k,m,n,p} s_k^+ \ , \\ [s_k^-, G_{m,n}] & = & -2M_{k,m,n} s_k^z - N_{k,k,m,n} s_k^+ \ . \end{array}\right\} \qquad \text{(B13)}$$

We note once more that the operators F_k, $G_{k,m}$, $M_{k,m,n}$, and $N_{k,m,n,p}$ are defined by Eq. (B3).

2 Enumeration of the Fundamental Clusters

At a given level of approximation, we choose only *one* of the $N_B(l!)\nu_I$ possible symmetry-equivalent configurations for a given *fundamental* configuration of l spin-raising operators, where N_B is the total number of Bravais lattice sites, and where ν_I is a symmetry factor dependent purely on the point-group symmetries (and *not* the translational symmetries) for the crystallographic lattice in question and for fundamental configuration I. We note that there are N_F such *fundamental* configurations. The first part of the computational algorithm is to enumerate all of the "lattice animals" which define the "locale" in which the clusters must lie. For the levels of approximation shown in this article it is possible to do this by using a simple recursive algorithm which enumerates all possible lattice animals of m contiguous sites. This "locale" is explicitly assumed here to be the same for both the LSUBm and SUBn-m approximation schemes. Secondly, one then needs to enumerate all possible ways in which one can place $(2s)$ or less spin-raising operators on each of the positions of the m sites within each of these lattice animals. There are thus $(2s)^m$ possibilities for each lattice animal. However, one must also restrict the total number of spin-raising operators to be less than or equal to n for the SUBn-m approximation scheme. We note however that there is no such restriction on the total number of spin-raising operators for the LSUBm approximation. This process thus enumerates all possible connected and disconnected clusters, and we make a restriction that we include only those clusters which are inequivalent under the point and space group symmetries of both the lattice and the Hamiltonian. A further restriction for the systems under consideration in this article is that we must restrict the set of fundamental clusters to include only those which preserve the relationship, $s_T^z = \sum_i s_i^z = 0$, with respect to the original ("unrotated") Néel model state since $[s_T^z, H] = 0$ and the ground state lies in the $s_T^z = 0$ sector.

3 The Ket-State Equations

We now wish to determine the CCM ket-state equations, where the I-th such equation is given by

$$E_I \equiv \frac{1}{A_I} \langle \Phi | C_I^- e^{-S} H e^S | \Phi \rangle = 0 \ \ , \forall I \neq 0 \ \ , \tag{B14}$$

where A_I is a normalisation factor given by $A_I \equiv \langle \Phi | C_I^- C_I^+ | \Phi \rangle = \langle \Phi | (s_{i_1}^- s_{i_2}^- \cdots s_{i_l}^-) \ (s_{i_1}^+ s_{i_2}^+ \cdots s_{i_l}^+) | \Phi \rangle$. We note once more that we choose only *one* of the $N_B(l!)\nu_I$ possible symmetry-equivalent configurations for a given *fundamental* configuration in C_I^- in order to pattern-match with the terms within $\tilde{H}|\Phi\rangle$ and thus determine the I-th CCM ket-state equation. We then computationally match the individual spin-lowering operators in C_I^-, defined by Eq. (B5), to the spin-raising operators in $\tilde{H}|\Phi\rangle$. We therefore put constraints on the indices in the CCM ket-state correlation coefficients, $\{\mathcal{S}_{i_1,i_2,\cdots,i_l}\}$, and these constraints on the indices allow us to enumerate all possible terms which contribute to the CCM ket-state equations. For example, we may consider a specific term in the evaluation of the CCM ket-state equations, given by

$$\begin{aligned}\langle\Phi|C_I^- \ \tilde{s}_k^z\tilde{s}_m^z|\Phi\rangle = \langle\Phi|C_I^- \Big(& s_k^z s_m^z + F_m s_m^+ s_k^z + F_k s_k^+ s_m^z \\ & + G_{k,m} s_k^+ s_m^+ + F_k F_m s_k^+ s_m^+ \Big)|\Phi\rangle \quad . \end{aligned} \tag{B15}$$

For the case of the linear chain Heisenberg model we let k run over all lattice sites on the 1D chain and we set $m = i + 1$. Now consider a specific term within Eq. (B15), given by $\langle\Phi|C_I^- F_k F_m s_k^+ s_m^+|\Phi\rangle$. We match the indices of the spin-lowering operators in C_I^- to the spin raising-operators in F_k, F_m, s_k^+, and s_m^+. Hence, both k and m are fully constrained to take on site values dependent on those indices of the spin-lowering operators in the fundamental configuration chosen for C_I^-. We note however that one may not always have such complete constraints on the indices k and m in $\tilde{H}$. For example, we may attempt to evaluate such a term as, $\langle\Phi|C_I^- F_k F_m s_k^z s_m^z|\Phi\rangle$. In this case, both k and m are free to cover all lattice sites *independently* from the fundamental cluster utilised in C_I^-. However, we may retain only those configurations in the set $\{\mathcal{S}_{i_1,i_2,\cdots,i_l}\}$ in F_k and F_m which are equivalent to the fundamental set of configurations under the symmetries of the lattice. This condition is sufficient to render the computational problem both tractable and efficient. Finally, the resulting coupled, non-linear CCM ket-state equations are easily solved computationally (for example, via the Newton-Raphson method) at a given value of the anisotropy parameter Δ.

4 The Bra-State Equations

The bra-state coefficients $\mathcal{S}_I$ of Eq. (A2) are formally determined by Eq. (A8). However, this form of the bra-state equations is slightly cumbersome to use, and a simpler and more elegant approach is possible by defining the following *new* set of CCM correlation coefficients given by

$$\left.\begin{aligned} x_I &= \mathcal{S}_I \\ \tilde{x}_I &= \tfrac{N_B}{N}\tilde{\mathcal{S}}_I A_I \nu_I(l!) \end{aligned}\right\} \quad , \tag{B16}$$

where again $A_I \equiv \langle\Phi|C_I^- C_I^+|\Phi\rangle = \langle\Phi|(s_{i_1}^- s_{i_2}^- \cdots s_{i_l}^-)\ (s_{i_1}^+ s_{i_2}^+ \cdots s_{i_l}^+)|\Phi\rangle$. Note that N_B is the number of Bravais lattice sites. Note also that for a given cluster I then ν_I is a symmetry factor which is dependent purely on the point-group symmetries (and *not* the translational symmetries) of the crystallographic lattice and that l is the number of spin operators. We note that the coefficients A_I, ν_I, and N_B however do not need to be explicitly determined because they *always* cancel out when obtaining ground-state expectation values (see below). The CCM bra-state operator may thus be rewritten as

$$\tilde{S} \equiv 1 + N \sum_{I=1}^{N_F} \frac{\tilde{x}_I}{A_I} C_I^- \quad , \tag{B17}$$

such that

$$\bar{H} = E_0 + N \sum_{I}^{N_F} \tilde{x}_I E_I \quad . \tag{B18}$$

We note again that the ground-state energy expectation value is defined by $E_0 = \langle\Phi|e^{-S} H e^S|\Phi\rangle$ and that E_I is the I-th CCM ket-state equation defined by Eq. (B14). The CCM ket-state equations are easily rederived by taking the partial derivative of $\bar{H}/N$ with respect to $\tilde{x}_I$, where

$$0 = \frac{\delta(\bar{H}/N)}{\delta\tilde{x}_I} \equiv E_I \quad . \tag{B19}$$

We now take the partial derivative of $\bar{H}/N$ with respect to x_I such that the bra-state equations take on a particularly simple form, given by

$$0 = \frac{\delta(\bar{H}/N)}{\delta x_I} = \frac{\delta(E_0/N)}{\delta x_I} + \sum_{J=1}^{N_F} \tilde{x}_J \frac{\delta E_J}{\delta x_I} \quad . \tag{B20}$$

This equation is easily solved computationally, once the CCM ket-state equations have been determined and solved, and the numerical values of the coefficients $\{\tilde{x}_I\}$ may thus be obtained. We note that this approach greatly simplifies the task of determining the bra-state equations because we never need to explicitly determine the factors N_B, A_I, or ν_I.

5 Expectation Values

Expectation values of spin operators may be treated in an analogous manner to that of the expectation value of the Hamiltonian, given by $\bar{H}$. For example, the sublattice magnetisation for the spin-half and spin-one antiferromagnets of Eq. (**??**) may be written as

$$\begin{aligned} M &= -\frac{1}{sN}\sum_{i=1}^{N}\langle\tilde{\Psi}|s_i^z|\Psi\rangle \\ &= 1-\frac{1}{s}\sum_{I=1}^{N_F}\frac{\tilde{x}_I}{A_I}\langle\Phi|C_I^-\sum_{i=1}^{N}(F_is_i^+)|\Phi\rangle \\ &= 1-\frac{1}{s}\sum_{I=1}^{N_F}\ l(l!)\ \tilde{x}_Ix_I \quad , \end{aligned} \tag{B21}$$

where i runs over all lattice sites. We again note that the factors A_I and ν_I in Eq. (B21) have cancelled out. Equation (B21) is easily evaluated once the ket- and bra-state equations have been solved at a given value of the anisotropy parameter Δ. (Results for both the ground-state energy and the sublattice magnetisation are given in Section III of this article.)

The situation for the ferrimagnet is slightly different, because the unit cell now contains two spins. Thus the magnetisation on N_1 the spin-half sites ($s=1/2$) is given by,

$$\begin{aligned} M_1 &= -\frac{1}{sN_1}\sum_{i_1}^{N_1}\langle\tilde{\Psi}|s_{i_1}^z|\Psi\rangle \\ &= 1-\frac{N}{sN_1}\sum_{I=1}^{N_F}\frac{\tilde{x}_I}{A_I}\langle\Phi|C_I^-\sum_{i_1=1}^{N_1}(F_{i_1}s_{i_1}^+)|\Phi\rangle \\ &= 1-4\sum_{I=1}^{N_F}\frac{\tilde{x}_I}{A_I}\langle\Phi|C_I^-\sum_{i_1=1}^{N_1}(F_{i_1}s_{i_1}^+)|\Phi\rangle \quad . \end{aligned} \tag{B22}$$

Note that i_1 runs over all $N_1(=N/2)$ spin-half lattice sites, such that we have the factor $\frac{N}{sN_1}=4$. It is a simple matter to explicitly enumerate all combinations of orderings of the l spin-raising operators in $\sum_{i_1=1}^{N_1}F_{i_1}s_{i_1}^+$ which match with spin-lowering operators in C_I^-, although we must also explicitly restrict i_1 to be a spin-half site. (We note that the factors A_I again cancel out with coefficients in $\langle\Phi|C_I^-\sum_{i_1=1}^{N_1}(F_{i_1}s_{i_1}^+)|\Phi\rangle$ at this point.) A similar expression may be obtained for M_2 which is given by

$$\begin{aligned} M_2 &= -\frac{1}{sN_2}\sum_{i_2}^{N_2}\langle\tilde{\Psi}|s_{i_2}^z|\Psi\rangle \quad ; \\ &= 1-2\sum_{I=1}^{N_F}\frac{\tilde{x}_I}{A_I}\langle\Phi|C_I^-\sum_{i_2=1}^{N_2}(F_{i_2}s_{i_2}^+)|\Phi\rangle \quad , \end{aligned} \tag{B23}$$

where i_2 runs over all $N_2(=N/2)$ spin-one lattice sites. Note that this approach was also utilised for a spin-half model which interpolated between a triangular-lattice antiferromagnet and a Kagomé-lattice antiferromagnet [29]. In this case, we wished to evaluate the sublattice magnetisation on the $N_K = 3N/4$ Kagomé-lattice sites only.

We may also determine numerical values for other expectation values, such as the spin-spin correlation function, given by

$$M_r^{zz} \equiv \frac{1}{N} \sum_{k=1}^{N} \langle \tilde{\Psi} | \tilde{s}_k^z \tilde{s}_{k+r}^z | \Psi \rangle \quad .$$

We see from Eq. (B15) above that this expression may be written in terms of our high-order CCM operators as

$$\begin{aligned} M_r^{zz} = \langle \Phi | \{ \frac{1}{N} + \sum_{I=1}^{N_F} \frac{\tilde{x}_I}{A_I} C_I^- \} \sum_{k=1}^{N} \Big(& s_k^z s_{k+r}^z + F_m s_{k+r}^+ s_k^z + F_k s_k^+ s_{k+r}^z \\ & + G_{k,k+r} s_k^+ s_{k+r}^+ + F_k F_{k+r} s_k^+ s_{k+r}^+ \Big) | \Phi \rangle \quad . \end{aligned} \tag{B24}$$

The right-hand-side of Eq. (B24) may be evaluated computationally in exactly the same manner as for the ket-state equations, In this case the only difference is that index r is *not* constrained to be a nearest-neighbour lattice vector, and a numerical value for M_r^{zz} for a specific lattice vector r is obtained once the ket- and bra-state equations have been solved at a given value of Δ. Within the NCCM formalism we note that $M_r^{zz} \rightarrow \frac{1}{N} \sum_{k=1}^{N} \langle \Phi | s_k^z s_{k+r}^z | \Phi \rangle$ as $r \rightarrow \infty$ using the localised LSUBm or SUBm-m approximation schemes. (Note that this is not true for the extended coupled cluster method [25], even for such localised approximation schemes.) However, Eq. (B24) may be successfully used to study the short-range behaviour of M_r^{zz}.

6 The High-Order Excited-State Formalism

Finally, we wish to outline an excited-state formalism developed by Emrich [9] which may be utilised in order to perform high-order CCM calculations for the exited state [26] of quantum spin systems of general spin quantum number. We note however that no excited-state results are presented in this article. The excited-state wave function, $|\Psi_e\rangle$, is determined in Emrich's formalism [9] by linearly applying an excitation operator X^e to the ket-state wave function of

Eq. (A2), such that

$$|\Psi_e\rangle = X^e \; e^S|\Phi\rangle \;\; . \tag{B25}$$

This equation may now be used to determine the excited state of H, where the Schrödinger equation is given by, $H|\Psi_e\rangle = E_e|\Psi_e\rangle$. This expression may be combined with its ground-state counterpart of Eq. (A1) to yield the CCM excited-state equation, given by

$$\epsilon_e X^e|\Phi\rangle = e^{-S}[H, X^e]e^S|\Phi\rangle \; (\equiv \hat{R}|\Phi\rangle) \;\; . \tag{B26}$$

(Note that $\epsilon_e \equiv E_e - E_g$ is the excitation energy.) By analogy with the ground-state formalism, the excited-state correlation operator is given by,

$$X^e = \sum_{I \neq 0} \mathcal{X}_I^e C_I^+ \;\; , \tag{B27}$$

where the set $\{C_I^+\}$ of multi-spin creation operators may differ from those used in the ground-state parametrisation in Eq. (A2), if the excited state has different quantum numbers than the ground state. Hence, by applying $\langle\Phi|C_I^-$ to Eq. (B26) we find that

$$\epsilon_e \mathcal{X}_I^e = \langle\Phi|C_I^- e^{-S}[H, X^e]e^S|\Phi\rangle \;\; ,\forall I \neq 0 \;\; . \tag{B28}$$

Equation (B28) thus forms a generalised set of eigenvalue equations with eigenvalues ϵ_e and corresponding eigenvectors $\mathcal{X}_I^e$ for each of the excited states. We note that in analogy to the ground-state high-order formalism we may define operators analogous to the F_k, $G_{k,m}$, $M_{k,m,n}$, and $N_{k,m,n,p}$ operators in order to facilitate the pattern-matching process of the fundamental configurations of the excited state in the set $\{C_I^-\}$ to the terms in $e^{-S}[H, X^e]e^S$. (We note that the lowest-lying excited states for the *XXZ* model lie in the $s_T^z = +1$ and $s_T^z = -1$ subspaces with respect to the "unrotated" ground state, and thus we would restrict the "fundamental" clusters in the set $\{C_I^+\}$ used in Eq. (**??**) to be those which reflect this property.) The resulting set of eigenvalue equations may be easily determined and solved, once the ket-state equations have been solved at a given value of Δ. Finally, the interested reader is referred to Ref. [26] for a full account of applications of high-order CCM calculations to the limiting spin-half case of the *XXZ* model for the linear chain, the square lattice, and the cubic lattice.

APPENDIX C: PARALLEL PROCESSING

Note that the CCM program may be run in parallel. We assume here that the *MPI* interface has been installed correctly on your machines. There are a number of steps in order to enable parallel processing:

1. Edit the `Makefile` so that it the top of this file now reads

```
# NON-PARALLEL MAKEFILE
#CC            = g++
#CCFLAGS       = -O -g
#LD       = g++
#LDFLAGS       = -O -g
#LIBS          = -lm

# PARALLEL MAKEFILE
CC            = mpiCC
CCFLAGS       = -O -g -lm
LD            = mpiCC
LDFLAGS       = -O -g -lm
LIBS          = -I../mpi/
```

2. Edit the file `Constants.h` in the directory *./include* so that the option `PARALLEL_FLAG` is set to 1. Now recompile the CCM code in *./bin*.

3. Ensure that the solution strategy is set to `Direct Iteration`. (A parallel version using Newton-Raphson to solve the CCM equations is envisaged for the future.)

4. Choose all other parameters as required. In particular, make sure that the *Work Directory* is set to a directory which is *local* to a given machine and that these directories exist on each machine.

A CCM simulation may now be run from the /it ./bin directory using:

```
mpirun -n 5 -p ccm.x
```

where the option `-n 5` sets the number of machines to be used in parallel – and is set at the discretion of the user. Note that the `-p` option may not be necessary on systems and that one might need to specify a *machinefile* on others. See your network administrator regarding these options.

REFERENCES

[1] F. Coester, *Nucl. Phys.* **7**, 421 (1958); F. Coester and H. Kümmel, *ibid.* **17**, 477 (1960).

[2] H. Kümmel, K.H. Lührmann, and J.G. Zabolitzky, *Phys Rep.* **36C**, 1 (1978).

[3] R.F. Bishop and K.H. Lührmann, *Phys. Rev. B* **17**, 3757 (1978).

[4] J.S. Arponen, *Ann. Phys. (N.Y.)* **151**, 311 (1983).

[5] J.S. Arponen, R.F. Bishop, and E. Pajanne, *Phys. Rev. A* **36**, 2519 (1987); *ibid.* **36**, 2539 (1987); *ibid.* **37**, 1065 (1988).

[6] R.F. Bishop, *Theor. Chim. Acta* **80**, 95 (1991).

[7] J.S. Arponen, and R.F. Bishop, *Ann. Phys.* (N.Y.) **207**, 171 (1991); *ibid.* **227**, 275 (1993); *ibid.* **227**, 2334 (1993).

[8] R.F. Bishop, in *Microscopic Quantum-Many-Body Theories and Their Applications*, edited by J. Navarro and A. Polls, Lecture Notes in Physics, Vol. **510** (Springer-Verlag, Berlin, 1998), p. 1.

[9] K. Emrich, *Nucl. Phys.* **A351**, 379, 397 (1981).

[10] M. Roger and J.H. Hetherington, *Phys. Rev. B* **41**, 200 (1990); M. Roger and J.H. Hetherington, *Europhys. Lett.* **11**, 255 (1990).

[11] R.F. Bishop, J.B. Parkinson, and Y. Xian, *Phys. Rev. B* **44**, 9425 (1991).

[12] R.F. Bishop, J.B. Parkinson, and Yang Xian, *Phys. Rev. B* **46**, 880 (1992).

[13] R.F. Bishop, J.B. Parkinson, and Y. Xian, *J. Phys.: Condens. Matter* **5**, 9169 (1993).

[14] D.J.J. Farnell and J.B. Parkinson, *J. Phys.: Condens. Matter* **6**, 5521 (1994).

[15] Y. Xian, *J. Phys.: Condens. Matter* **6**, 5965 (1994).

[16] R. Bursill, G.A. Gehring, D.J.J. Farnell, J.B. Parkinson, T. Xiang, and C. Zeng, *J. Phys.: Condens. Matter* **7**, 8605 (1995).

[17] R. Hale, *Ph.D. Thesis*, UMIST Manchester, United Kingdom, (1995).

[18] R.F. Bishop, R.G. Hale, and Y. Xian, *Phys. Rev. Lett.* **73**, 3157 (1994).

[19] R.F. Bishop, D.J.J. Farnell, and J.B. Parkinson, *J. Phys.: Condens. Matter* **8**, 11153 (1996).

[20] D.J.J. Farnell, S.A. Krüger, and J.B. Parkinson, *J. Phys.: Condens. Matter* **9**, 7601 (1997).

[21] R.F. Bishop, Y. Xian, and C. Zeng, in *Condensed Matter Theories*, Vol. **11**, edited by E.V. Ludeña, P. Vashishta, and R.F. Bishop (Nova Science, Commack, New York, 1996), p. 91.

[22] C. Zeng, D.J.J. Farnell, and R.F. Bishop, *J. Stat. Phys.*, **90**, 327 (1998).

[23] R. F. Bishop, D.J.J. Farnell, and J.B. Parkinson, *Phys. Rev. B* **58**, 6394 (1998).

[24] R. F. Bishop, D. J. J. Farnell, and Chen Zeng, *Phys. Rev. B* **59**, 1000 (1999).

[25] J. Rosenfeld, N.E. Ligterink, and R.F. Bishop, *Phys. Rev. B* **60**, 4030 (1999).

[26] R. F. Bishop, D. J. J. Farnell, S.E. Krueger, J. B. Parkinson, J. Richter, and C. Zeng, *J. Phys.: Condens. Matter* **12**, 6887 (2000).

[27] R.F. Bishop, D.J.J. Farnell, and M.L. Ristig, *Int. J. Mod. Phys. B* **14**, 1517 (2000).

[28] S.E. Krüger, J. Richter, J. Schulenberg, D.J.J. Farnell, and R.F. Bishop, *Phys. Rev. B* **61**, 14607 (2000).

[29] D. J. J. Farnell, R. F. Bishop, and K. A. Gernoth, *Phys. Rev. B* **63**, 220402R (2001).

[30] D. J. J. Farnell, K. A. Gernoth, and R. F. Bishop, *Phys. Rev. B* **64**, 172409 (2001).

[31] D. J. J. Farnell, K. A. Gernoth, and R. F. Bishop, *J. Stat. Phys.* **108**, 401 (2002).