ETHER

Efficient Tool for THermodynamics Exploration via Relaxations

developed by

Dr. Mukesh Kumar Sharma

PhD, Indian Institute of Technology Roorkee, India

Preface

Monte Carlo (MC) simulations are powerful tools for validating analytical methods and solving large, complex systems where analytical approximations become challenging. Motivated by these compelling features, I have developed a simulation package, named ETHER (Efficient Tool for THermodynamics Exploration via Relaxations), designed to uncover temperature-dependent phenomena in magnetic materials. Using ETHER, users can perform extensive MC simulations for any spin system to study phase transitions, critical phenomena, and complex magnetic structures.

The initial development of ETHER began during my Ph.D. tenure (2017–2023) at the Indian Institute of Technology Roorkee, India, with the objective of investigating magnetic materials and their role in multiferroic properties. The simulation package is primarily written in FORTRAN, with core scientific calculations implemented in well-structured and easily understandable subroutines. In addition, ETHER supports parallel computation in hybrid mode using both Open Multi-Processing (OpenMP) and Message Passing Interface (MPI) protocols, significantly improving computational efficiency. Over time, the package has been rigorously tested, and several bugs have been addressed.

Accurate MC simulations for complex spin Hamiltonians require precise information about the spin lattice and neighboring magnetic ions. Incorrect lattice definitions can lead to unexpected or erroneous results. While defining crystal lattices via space group symmetry operators is straightforward, handling doped or antisite-disordered systems is more tedious. Doping and disorder are typically introduced at the unit-cell level, but achieving correct statistical representations requires the construction of larger supercells [**?** **?**]. However, using large supercells increases computational demands and time, especially when studying disorder effects.

ETHER overcomes this limitation. Unlike conventional methods where doping or disorder is confined to the unit cell, ETHER allows the implementation of desired concentrations directly in a supercell. This feature gives users the flexibility to perform MC simulations for varying concentrations within a fixed supercell size, thus saving computational resources and allowing systematic exploration of disorder effects. For a given magnetic material, ETHER constructs the spin lattice network internally using the input structure file structure.vasp. This file, similar in format to POSCAR files used in VASP [**?** **?** **?**], can easily be generated using visualization software such as VESTA [**?**]. Exchange interactions, which define how a magnetic ion interacts with its neighbours, are provided via the *j_exchange* input file. ETHER identifies neighbouring ions either by their names or by bond lengths specified in this file. These neighbour lists can be confirmed and visualized using tools like XCrySDen [**?**]. Additionally, several other input files help the program capture detailed spin-system information. Upon completion of MC simulations, ETHER generates multiple output files, including final spin states that can be visualized using VESTA. The package also provides post-processing tools to help users analyze simulation data effectively.

The primary objective of developing ETHER is to offer researchers a flexible, user-friendly tool for temperature-dependent Monte Carlo simulations of complex magnetic materials. I hope users will find the code intuitive and easy to work with. Detailed instructions on installation, input/output formats, and visualization methods are provided in this user guide. Researchers are encouraged to contribute to the ongoing development of ETHER.

Dr. Mukesh Kumar Sharma

Contents

1 Hamiltonian

Currently, Ether uses the following Hamiltonian for evaluating the thermodynamic property of any spin system.

$$H = \sum_{<i,j>} \left(J^{ij}_{xx} S^i_x S^j_x + J^{ij}_{yy} S^i_y S^j_y + J^{ij}_{zz} S^i_z S^j_z \right) + \sum_i [D^i_x (S^i_x)^2 + D^i_y (S^i_y)^2 + D^i_z (S^i_z)^2] - g\mu_B \sum_i [H_x S^i_x + H_y S^i_y + H_z S^i_z] \tag{1}$$

Classical spin vector:	$\vec{S}$ $[S_x, S_y, S_z]$
Exchange interaction:	J $[J_{xx}, J_{yy}, J_{zz}]$
Magnetic field:	$\vec{H}$ $[H_x, H_y, H_z]$
Single ion anisotropy vector:	$\vec{D}$ $[D_x, D_y, D_z]$
g-factor:	g
Bohr magneton:	μ_B

First, *Second* and *Third* term represents the classical XYZ spin model, single ion anisotropy, and Zeeman terms, respectively. <> stands for nearest neighbours and here i representing the i-th spin site. Given Hamiltonian can converted into classical Heisenberg model at the condition when $J_{xx} = J_{yy} = J_{zz} = J$ or into any anisotropic model like XXZ ($J_{xx} = J_{yy}$) or, XY model ($J_{xx} = J_{yy} = J, J_{zz} = 0$). Further *User* can switch into the well known Ising model by selecting the choice of model in '*in.ether*' input file. Here if $J > 0$ then interaction will be anti-ferromagnetic else if $J < 0$ then system will have ferromagnetic interactions whereas spins prefers to align towards the direction where negative component of $\vec{D}$ is present .

If logical variable *PARA* (*see: Units*) is set to *.true.* for a given $|J_{para}| > 0$ then *Ether* considers the following effective Hamiltonian,

$$\tilde{H} = \frac{H}{J_{para}} \tag{2}$$

2 Units

Ethers by default takes:

IF [PARA = .FALSE., *see in.ether*]

J in terms of milli-electron-Volts (meV)
$\vec{D}$ in terms of meV
$\vec{h}$ in Tesla
μ_B in eV/Tesla (internally fixed)
Bond lengths in terms of Angstrom (Å)
Temperature(T) range in terms of Kelvin (K)

ELSE [PARA = .TRUE. with given J_{para}, *see in.ether*]

J as unit less
$\vec{D}$ as unit less
$\vec{h}$ as unit less
μ_B = 1
Bond lengths in terms of Angstrom (Å)
Temperature range in terms of k_BT/J such that $J = \mathrm{J}_{para}$

Note : Spin S will always be in multiples of $\frac{1}{2}$ and is independent of choice of *parameter*.

3 Observables A

The framework of the conventional Monte Carlo technique is built on the basis of statistical mechanics. Where, we calculate any thermodynamic property by taking its average over all possible states. Suppose, A is our thermodynamical observable in which we are interested, and A_s denotes the computed value when the system was in state s with energy E_s. Then the expectation (<A>) of observable A can be defined as,

$$\langle A \rangle = \frac{1}{Z} \sum_s A_s e^{-\beta E_s} \tag{3}$$

where Z is the partition function and is defined as $\sum_s A_s e^{-\beta E_s}$ such that it covers all states through which the system undergoes. Knowledge of exact Z could derive the <A> exactly, and unfortunately, we do not know Z**??**. In 1953, Nicholas Metropolis and his co-workers**??** deduced an astonishing method to evaluate the expectation value of targeted thermodynamic observable without evaluating Z and thus empowered the MC simulation which we currently know.

During MC simulation, the basic idea is to obtain a series of states, called *Markov chain*, such that each state, say s_i, only depends on its immediately preceding state s_{i-1}. In this process, the inclusion of new states can only be possible if it satisfies the specific transition probability criteria. Suppose a system at state s_i with energy E_i such that Boltzmann factor is $p_i \propto e^{-\beta E_i}$, and our trial new configuration s_j with energy E_j having Boltzmann factor p_j. Then the acceptance or rejection of this new trial configuration will be done according to the following condition

$$P_{ij} = \begin{cases} e^{-\beta(E_j - E_i)} & \text{if } E_j > E_i \\ 1, & \text{if } E_j \leq E_i \end{cases} \tag{4}$$

or, often above condition can also be written as,

$$P_{ij} = min[e^{-\beta \Delta_{ij}}, 1] \tag{5}$$

Here, P_{ij} is the acceptance probability, and $\Delta_{ij} = E_j - E_i$ is the energy difference. If trail state s_j is accepted, the associated configuration will be saved and considered as the new member of the series, and if it is rejected, the next member will be again s_i. Generally, the criteria given in equation 4 is implemented using random number $r \in [0, 1)$ and the new state will be accepted only if $r < P_{ij}$.

Following the above condition, suppose we have a list of states s_1, s_2, ..., s_n and such that calculated observable A_i is associated with the state s_i. Then the thermodynamic property A, can be estimated by simply averaging the A_i, i.e.,

$$\langle A \rangle \approx \frac{1}{n} \sum_{i=1}^{n} A_i \tag{6}$$

Ether can calculate the following observables at temperature (T),

Total internal energy per site,

$$E(T) = \frac{1}{N}\langle H \rangle \tag{7}$$

Specific heat per site,

$$C(T) = \frac{1}{k_B T^2}\left(\left\langle E^2 \right\rangle - \langle E \rangle^2\right) N \tag{8}$$

Magnetization,

$$M = \frac{1}{N}\langle |\sum_{i}^{n}\sum_{j}^{l} c_j \vec{S}_j| \rangle \tag{9}$$

Susceptibility per site,

$$\chi(T) = \frac{1}{k_B T}\left(\left\langle M^2 \right\rangle - \langle M \rangle^2\right) N \tag{10}$$

Structure factor,

$$S_f(q) = \frac{1}{N}\sum_{r,r'} \langle S_r S_{r'} \rangle \, e^{iq(r-r')} \tag{11}$$

Cumulant,

$$U_A = 1 - \frac{1}{3}\frac{\left\langle A^4 \right\rangle}{\left\langle A^2 \right\rangle^2} \tag{12}$$

4 Standard deviation and error of A

For an observable A, if $\langle A \rangle$ is calculated m times repeatedly such that obtained values are $\langle A \rangle_1$, $\langle A \rangle_2$,...,$\langle A \rangle_m$ and their evaluated mean is $\langle A \rangle_{mean}$,

$$\langle A \rangle_{mean} = \frac{1}{m}\sum_{i=1}^{m} \langle A \rangle_i \tag{13}$$

then,

Standard deviation in calculated A is:

$$\sigma_A = \sqrt{\frac{1}{m-1}\sum_{i=1}^{m} (\langle A \rangle_i - \langle A \rangle_{mean})^2} \tag{14}$$

Standard error (uncertainty) is:

$$\Delta A = \frac{\sigma_A}{\sqrt{m}} \tag{15}$$

thus A can be reported as: $A = \langle A \rangle_{mean} \pm \Delta A$

5 Inputs

5.1 *in.ether*

'*in.ether*' file contains the necessary input information based on which *Ether* will build-up the internal framework and initiate the MC simulation accordingly. User can provide the following currently existing input tags before initiating the MC simulation:

Tags	Description	Default
MODEL	Defines the choice of model consider during MC simulation. (*Ising/XYZ*)	Ising
MCS	Monte Carlo steps per spins (MCS) Total MCS for simulation, Total MCS for equilibration process.	5000 3000
TEMP	Sets the temperature range. *Note: Simulation will start from higher temperature.* (T_{Start}, T_{End}, $T_{interval}$)	100 10 5
SPIN	Spin values of each ion present in the *structure.vasp* file. *Note: For non magnetic ion, kindly assign 0.0*	
SPECIES	Symbol of species for which MC simulations is intended to perform.	
SC	Extent of supercell along x, y, and z direction. (*X, Y, Z*)	2 2 2
STG	Logic for staggered magnetization. If .TRUE. then *Ether* will require the mandatory '*staggered*' file for further inputs, *see: staggered*. (*Logic*)	.FALSE.
BC	Boundary conditions (c: Closed, o: Open) along (X, Y, Z) direction	c c c
REPEAT	For any observable, repeat the same MC simulations *m* times for better results, *check equation* 13.	10
ANGLE	To update the direction of spin vector. *Note*: If this logic is .TRUE. then spins will be updated according to $S_x = \sqrt{1-S_z^2}\,cos\phi$, $S_y = \sqrt{1-S_z^2}\,sin\phi$, and $-1 \leq S_z \leq +1$ such that $\phi = \phi_{old} + \phi^{'}\delta\,(0 \leq \delta \leq 1)$ where ϕ_{old} is the earlier ϕ. Otherwise, new spins follows the George Marsaglia[**?**] rule. (*logic, ϕ*)	.FALSE. 5
COA	Calculate observables at each defined MCS.	20
ZEEMAN	Zeeman logic with external magnetic field. *see equation* 1. (*logic, H_x, H_y, H_z*)	.FALSE. 0 0 0
G_FACTOR	g_factor value	2
SIA	Single ion anisotropy logic. *see equation* 1 (*logic*)	.FALSE.
PARA	Parameter logic with J_{para} in meV. *see equation* 2. (*logic, J_{para}*)	.FALSE. 1.0
OVRR	Logic for performing over-relaxation (OVRR) [**?**] method applied on total (over_para×total_magnetic_lattices) lattices after each ovrr_MCS steps. (*logic, over_para, ovrr_MCS*)	.FALSE. 0 0
SEED	Seed number for random number generation	1992
DOPE	Logic for performing MC simulations with doping/disorder cases. *Note* : If .TRUE., then user must provide the list of dopant list, *see: dopants*. (*logic*)	.FALSE.

5.2 *j_exchnage*

'*j_exchange*' file contains the list of magnetic exchange interactions (J) acted between the magnetic ions. Since *Ether* recognizes the J^{AB} between the ions from theirs species symbols(A, B) or their connected bond lengths d_{AB}, *user* must pay attention while making this file during providing these J, d, and *species symbol* information.

Let's assume, we have two magnetic ions (A, B) present in the unit cell and d_{AB}, d_{AA}, d_{BB} are the connected bond lengths* between magnetic ion pair A-B, A-A, and B-B respectively such that $d_{AA} = d_{BB} = 7\mathring{A}$ and $d_{BB} = 5\mathring{A}$ then, following is the expected file formate for making *j_exchange* file,

```
================================================
 2                  ! Number of distinct bonds (2: d_AB and d_AA = d_BB)
 5.0 1              ! Bond length d_AB, number of similar bonds (1)
 A B -1.0 2.0 3.0   ! Ion A, Ion B, J_xx^AB, J_yy^AB, J_zz^AB
 7.0 2              ! Bond length d_AA, number of similar bonds (2)
 A A 1 1 1          ! Ion A, Ion A, J_xx^AA, J_yy^AA, J_zz^AA
 B B 2 3 4.0        ! Ion B, Ion B, J_xx^BB, J_yy^BB, J_zz^BB
================================================
```

Note: J values presented above are just for an example

* *Use VESTA[**?**] software to see how many distinct bonds are present among the magnetic ions.*

5.3 *single_ion_anisotropy*

For single ion anisotropy (SIA), SIA = .TRUE., calculations. *User* must provide the lists of SIA vector components (D_x, D_y, D_z) for desired species, present in the *structure.vasp* file, into the file: *single_ion_anisotropy*.

Formate of *single_ion_anisotropy* file is given below,

```
================================================
 Species_i      D_x^i D_y^i D_z^i      ! For Species(i) in unit cell
 Species_i+1    D_x^i+1 D_y^i+1 D_z^i+1  ! For Species(i+1) in unit cell
 . .
 . .
 . .
================================================
```

5.4 *staggered*

If we are dealing with the antiferromagnetic system, evaluating magnetization or its derivatives like susceptibility or cumulant becomes tedious as they solely depend on the average sum of spins, which becomes zero or negligibly small below the critical temperature.

Staggered magnetization, defined as the difference of magnetization obtained antiferromagnetically coupled two magnetic sublattices, can help in fetching the information which was absent in the total magnetization. In the context of evaluating staggered amgnetization, *users* must set the *STG* variable as .TRUE. in *in_ether* input file and provide the staggered coefficient (c_i, *see equation 9*) with +1, -1 or any real values into the '*staggered*' file. *Ether* will form such two distinct magnetic sublattice based on the sign given to c_i and perform the MC simulation accordingly.

Note: Performing staggered calculations will not affect the observables calculated from the total internal energy E.

Formate of *staggered* file is given below,

```
==========================================
 c_i      ! For atom(i) in unit cell
 c_{i+1}  ! F or atom(i+1) in unit cell
 . .
 . .
 . .
==========================================
```

6 Outputs

6.1 <tt.tttt>spK.xsf

Files with '<tt.tttt>spK.xsf' names are the output files generated during MC simulation at temperature <tt.tttt> and contains the calculated final ground spin states. Once the simulation completes, all files with names ending with $*spK.xsf$ can be found in the $/spin$ directory. Thus, *user* can analyze the spin configuration at any temperature using VESTA [**?**] program. Figure 1 illustrating the ground spin configurations calculated at 5 K temperature for supercell size $2\times2\times2$.

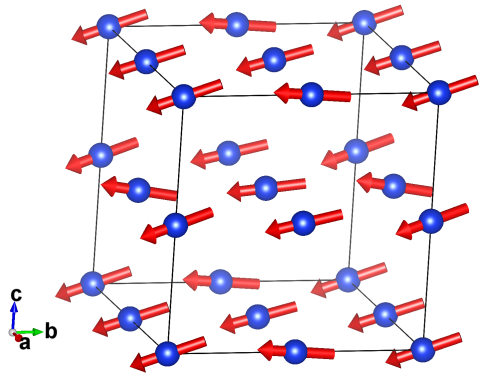

Figure 1: The observed ground state spin arrangements using *Ether*.
(*Results shown for:* /examples/Simple_Cubic/222_2nbd/)

6.2 *nbd.dat*

List of nearest neighbourhood with respect to any central ion in a supercell can be found in this file. User can visualize the such list in well details by using the well known Xcrysden[**?**] software.

The format of *nbd.dat* can be understood as follows:

The first four lines, starting with "This file can be .... information, " explaining the process for visualizing the neighbouring ion details. The information presented in the given figure 2 is shown for simple cubic case ((*see:* /examples/Simple_Cubic/222_2nbd/)) with supercell size $2\times2\times2$ in which ions with second nearest neighbours are considered. *Users* are advised to rerun the calculation in the given directory to reproduce neighbourhood (nbd) *nbd.dat* file for further analysis. Here number of distinct bonds (specified in *j_exchange* file) defines the number of nbd. Depending on the demand, *user* can extend to the third or higher nearest neighbour by specifying the total distinct bonds, their lengths and their associated connected ions in the *j_exchange*.

```
This file can be used to confirm the neighbourhood details of any central ion.
To do so, use 'XCrySDen' (http://www.xcrysden.org/) and open the generated
'starting_spin_conf.xsf' as file > Open Structure > Open XSF > starting_spin_conf.dat.
Check nbd's details by clicking the <Atom Info> (at bottom) and see the 'Selected Atom No.' informations.

ION no.     1
~~~~~~~~~~~~~~
For bond length no.  1 nbds are ==>      37      7      2
For bond length no.  2 nbds are ==>      43     38      8

ION no.     2
~~~~~~~~~~~~~~
For bond length no.  1 nbds are ==>       1     38      8      3
For bond length no.  2 nbds are ==>      37      7     44     39      9

ION no.     3
~~~~~~~~~~~~~~
For bond length no.  1 nbds are ==>       2     39      9      4
For bond length no.  2 nbds are ==>      38      8     45     40     10

......

......

......
```

Figure 2: List of neighbouring ions details for ***i-th* ION no.** arranged in simple cubic lattice. (*Results shown for:* /examples/Simple_Cubic/222/data/nbd.dat)

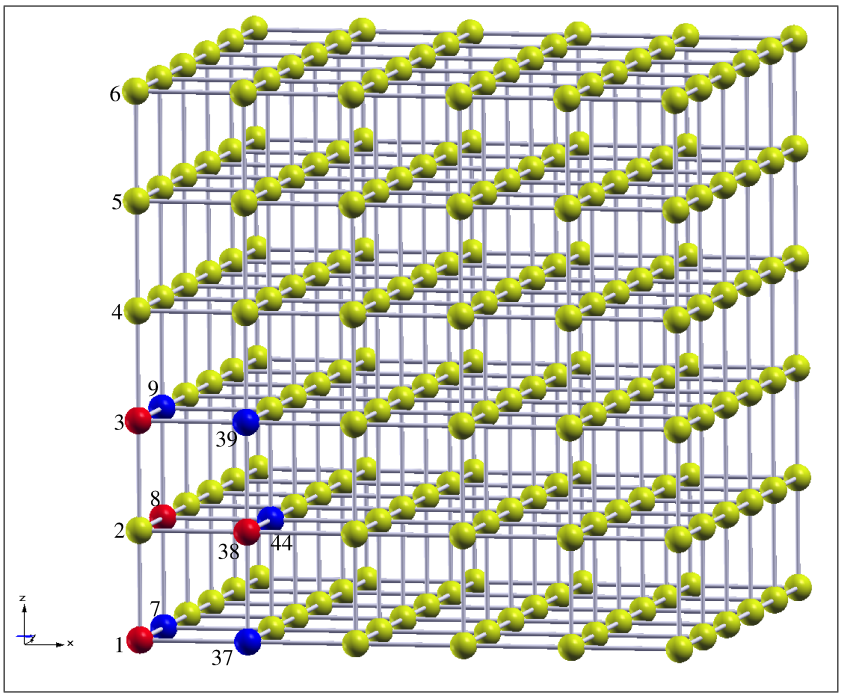

Figure 3: Visualizing the generated '*starting_spin_conf.xsf*' file using XCrySDen for illustrating the 1^{st} (in red spheres) and 2^{nd} (in blue spheres) neighbourhood **ID** details for the central ion with **ID 2**. *Note*: The above result is generated for supercell size $2\times2\times2$, but the picture depicts the lattice network of $2\times2\times2$, which is normal here because Ether uses extra layers around core supercell $2\times2\times2$, for setting periodic boundary conditions. The thickness of this layer (in terms of the unit cell) depends on the extent of the largest bond length described in the '*j_exchange*' file.

6.3 *gss.dat*

The ground spin states file (*gss.dat*) contains information about the final spin states found at each temperature during the simulation. Furthermore, it also includes the necessary information related to the simulation box used during the initialization of the Monte Carlo simulation. It is recommended that users must not

edit this file as it contains the core data, which will be used by post-processing tools present in the Ether's utility folder.

6.4 *moment_vector.dat*

The *moment_vector.dat* file contains the information about the simulation box's temperature-dependent net magnetic moment vector for each magnetic species. This helps the users visualize how the magnetic moment evolves on specific magnetic lattice networks as a function of temperature. The format of the *moment_vector.dat* file can be understood as follows:

1. *The first line* contain the definition.
2. *second line* gives the ordering of the necessary magnetic information.
3. From the *third line*, as shown in the figure (Figure 4), has the actual data structured in columns. The first column has temperature information. From the second column onward are the magnetic moment vectors (ordering: M_x, M_y, M_z) started from $1st$-species (sp_1) till the last Nth-species (sp_N) then after rest column, following the same order, presenting the magnitude of ith-species ($sp_i|M|$). Please note that the ordering of species written here will follow precisely the same order as in the *structure.vasp* file.

```
# spN = Nth species, M = magnitude of magnetic moment vector (Mx, My, Mz)
# Temp, sp1(Mx,My,Mz), sp1(Mx,My,Mz),..spN(Mx,My,Mz), sp1|M|, sp2|M|,..spN|M|
 100.00000     -0.454   -0.457    0.006    0.644
  95.00000     -0.244   -0.429   -0.056    0.497
  90.00000     -0.299   -0.535    0.176    0.638
  85.00000      0.396   -0.571   -0.109    0.703
  80.00000     -0.369   -0.351   -0.288    0.585
  75.00000     -0.089   -0.572    0.056    0.582
  70.00000      0.286   -0.662   -0.267    0.768
  65.00000     -0.086   -0.490   -0.016    0.498
  60.00000      0.137   -0.705   -0.187    0.743
  55.00000      0.287   -0.619   -0.164    0.701
  50.00000     -0.095   -0.756   -0.087    0.767
  45.00000     -0.213   -0.635   -0.153    0.687
  40.00000     -0.004   -0.797   -0.161    0.813
  35.00000     -0.203   -0.850   -0.128    0.883
  30.00000     -0.239   -0.850   -0.101    0.888
  25.00000     -0.175   -0.903   -0.110    0.926
  20.00000     -0.011   -0.826   -0.393    0.915
  15.00000     -0.230   -0.902   -0.216    0.956
  10.00000      0.024   -0.932   -0.158    0.946
   5.00000     -0.033   -0.964   -0.209    0.987
```

Figure 4: Illustrating the format of the 'moment.dat' file evaluated for the ferromagnetic spin system arranged in a Simple Cubic lattice network. *see:* /examples/Simple_Cubic/data/moment.dat

6.5 *magnetization.dat*

This file contains the calculated magnetization as a function of temperature. Data are structured in 7 columns (see figure 5) which are as follows,

1. Temperature
2. Total average magnetization $|M|$, [eqn. 9]
3. Susceptibility (χ), [eqn. 10]
4. Error in $|M|$, [eqn. 15]
5. Error in χ, [eqn. 15]
6. Cumulant of magnetization, (U_{Mag}), [eqn. 12]
7. Error in U_{Mag}, [eqn. 15]

```
# 1 Temp.    2 avg_mag    3 Chi        4 err_mag    5 err_chi.   6 U_mag.     7 err_U
  100.00000  5.69934E-01  2.04842E+01  4.41508E-03  7.23390E-01  5.92025E-01  2.96322E-03
   95.00000  5.87972E-01  2.11610E+01  3.42509E-03  7.02190E-01  5.98026E-01  2.21771E-03
   90.00000  5.99071E-01  2.21641E+01  3.43480E-03  7.34342E-01  6.02184E-01  2.54878E-03
   85.00000  6.24976E-01  2.14011E+01  3.61803E-03  9.41113E-01  6.11653E-01  2.25374E-03
   80.00000  6.41514E-01  2.26427E+01  2.94396E-03  5.38847E-01  6.14313E-01  1.64149E-03
   75.00000  6.64985E-01  2.09243E+01  3.24126E-03  7.59763E-01  6.24520E-01  1.39746E-03
   70.00000  6.89164E-01  1.97042E+01  2.84959E-03  7.11472E-01  6.31697E-01  1.07973E-03
   65.00000  7.09721E-01  1.97783E+01  3.58290E-03  7.38541E-01  6.35208E-01  1.20499E-03
   60.00000  7.31947E-01  1.75887E+01  1.91633E-03  7.48837E-01  6.42472E-01  9.35795E-04
   55.00000  7.57340E-01  1.57922E+01  2.66344E-03  5.56514E-01  6.47768E-01  5.73807E-04
   50.00000  7.83787E-01  1.45891E+01  2.15357E-03  7.37165E-01  6.51700E-01  6.99717E-04
   45.00000  8.08414E-01  1.28446E+01  1.83031E-03  4.97806E-01  6.55318E-01  4.09515E-04
   40.00000  8.30910E-01  1.07312E+01  1.35647E-03  4.05106E-01  6.58409E-01  2.86321E-04
   35.00000  8.54216E-01  8.98732E+00  1.46633E-03  4.22263E-01  6.60878E-01  2.51144E-04
   30.00000  8.79277E-01  7.46655E+00  1.39413E-03  3.61850E-01  6.62746E-01  1.79328E-04
   25.00000  8.98679E-01  5.83974E+00  9.37729E-04  2.35324E-01  6.64183E-01  9.17600E-05
   20.00000  9.20045E-01  4.44191E+00  1.05648E-03  1.75566E-01  6.65196E-01  5.85192E-05
   15.00000  9.40251E-01  3.04390E+00  4.79610E-04  1.46018E-01  6.65938E-01  3.44906E-05
   10.00000  9.60872E-01  2.08778E+00  5.15319E-04  1.09732E-01  6.66346E-01  1.65671E-05
    5.00000  9.80656E-01  1.12782E+00  2.87148E-04  6.50608E-02  6.66583E-01  4.79904E-06
```

Figure 5: Illustrating the format of the 'magnetization.dat' file evaluated for the ferromagnetic spin system arranged in a Simple Cubic lattice network. *see:* /examples/Simple_Cubic/data/magnetization.dat

6.6 *energy.dat*

This file contains the calculated energy-dependent observables as the function of temperature. Data are structured in 8 columns (see figure 5) whose descriptions are as follows,

1. Temperature
2. Total internal energy (E), [eqn. 7]
3. Specific heat (C_V/k_B), [eqn. 8]
4. Error in E, [eqn. 15]
5. Error in C_V/k_B, [eqn. 15]
6. Cumulant of E, (U_E), [eqn. 12]
7. Error in U_E, [eqn. 15]
8. Acceptance ratios (in %) defined as how much MCS were accepted out of total MCS for generating new states.

# 1 Temp.	2 avg_eng	3 Cv	4 err_eng	5 err_Cv	6 U_eng	7 err_U	8 accpt. (%)
100.00000	-3.61978E-03	3.04433E-01	4.73629E-05	9.51909E-03	4.61594E-01	7.97193E-03	52.16
95.00000	-3.84982E-03	3.32303E-01	3.51423E-05	1.01767E-02	4.89085E-01	4.85356E-03	49.86
90.00000	-3.98656E-03	3.55124E-01	3.48669E-05	9.53616E-03	5.03043E-01	6.47545E-03	47.50
85.00000	-4.31745E-03	3.84012E-01	4.35223E-05	1.25493E-02	5.28842E-01	4.44342E-03	45.04
80.00000	-4.50534E-03	4.36720E-01	3.54444E-05	8.35706E-03	5.35460E-01	3.84916E-03	42.38
75.00000	-4.79277E-03	4.48410E-01	3.85049E-05	1.33632E-02	5.61500E-01	3.02790E-03	39.71
70.00000	-5.09247E-03	4.66623E-01	3.41816E-05	1.82434E-02	5.81842E-01	2.35041E-03	36.93
65.00000	-5.40427E-03	5.09963E-01	4.06442E-05	1.67325E-02	5.93418E-01	2.45007E-03	33.86
60.00000	-5.68658E-03	5.07639E-01	2.27209E-05	1.71975E-02	6.09537E-01	1.87600E-03	31.35
55.00000	-6.01694E-03	5.22419E-01	3.25633E-05	1.54487E-02	6.21735E-01	1.32448E-03	28.52
50.00000	-6.35845E-03	5.57763E-01	2.87932E-05	2.63255E-02	6.30354E-01	1.70862E-03	25.41
45.00000	-6.70728E-03	5.52811E-01	2.25043E-05	1.68431E-02	6.39870E-01	7.72901E-04	22.58
40.00000	-7.02767E-03	5.49776E-01	1.98719E-05	1.74329E-02	6.46706E-01	5.91987E-04	19.81
35.00000	-7.36689E-03	5.34002E-01	2.10377E-05	1.53272E-02	6.53086E-01	3.86319E-04	17.07
30.00000	-7.72485E-03	5.46066E-01	1.96481E-05	2.42128E-02	6.57275E-01	3.95693E-04	14.41
25.00000	-8.01359E-03	5.38519E-01	1.32727E-05	1.80143E-02	6.60545E-01	1.86850E-04	11.97
20.00000	-8.34340E-03	5.12681E-01	1.43539E-05	1.74078E-02	6.63148E-01	1.18918E-04	9.33
15.00000	-8.65107E-03	4.92302E-01	7.33744E-06	2.29997E-02	6.64878E-01	8.32596E-05	7.04
10.00000	-8.97281E-03	5.19779E-01	6.59825E-06	2.57402E-02	6.65884E-01	3.75043E-05	4.66
5.00000	-9.28964E-03	5.46812E-01	4.88377E-06	3.16103E-02	6.66473E-01	1.11714E-05	2.34

Figure 6: Sample preview of 'energy.dat' file estimated for the ferromagnetic spin system arranged in Simple Cubic lattice network. *see:* /examples/Simple_Cubic/data/energy.dat

6.7 *graph.sh*

File '*graph.sh*' contains the minimal 'gnuplot' script to plot the obtained results for quick analysis.

COMMAND $\Longrightarrow$ gnuplot graph.sh

Above command will generate a 2×2 plot (file name: *results_Ether.png*, Figure 7) formed by total internal energy, specific heat, net magnetization, and susceptibility plots. As per the requirement, user may modify the *graph.sh* for plotting any other observable.

7 Utility tools

7.1 Vector

Vector program 'vector.f90' (/utility/Vector/vector.f90) can help in visualizing the final spin vectors obtained at temperature (T) using Ether. The main task of this utility program is to plot the spin vectors on the surface of unit sphere such that they are originated from the center of the sphere. This special visual technique is important when you are dealing with a larger supercell and want to see the orientation of spin vectors. *User* must provide the spin file ($< tt.tttt >$**spK.xsf**) obtained at temperature (*tt.tttt*) which usually present in the **spin** directory after completion of the simulation. The vector program reads this special file and gathers the data for visualization. Following are the commands to compile and run this tool,

COMMAND $\Longrightarrow$ gfortran -o vector vector.f90

RUN $\Longrightarrow$ ./vector

The descriptions of input/output files for this utility tool are as follows:

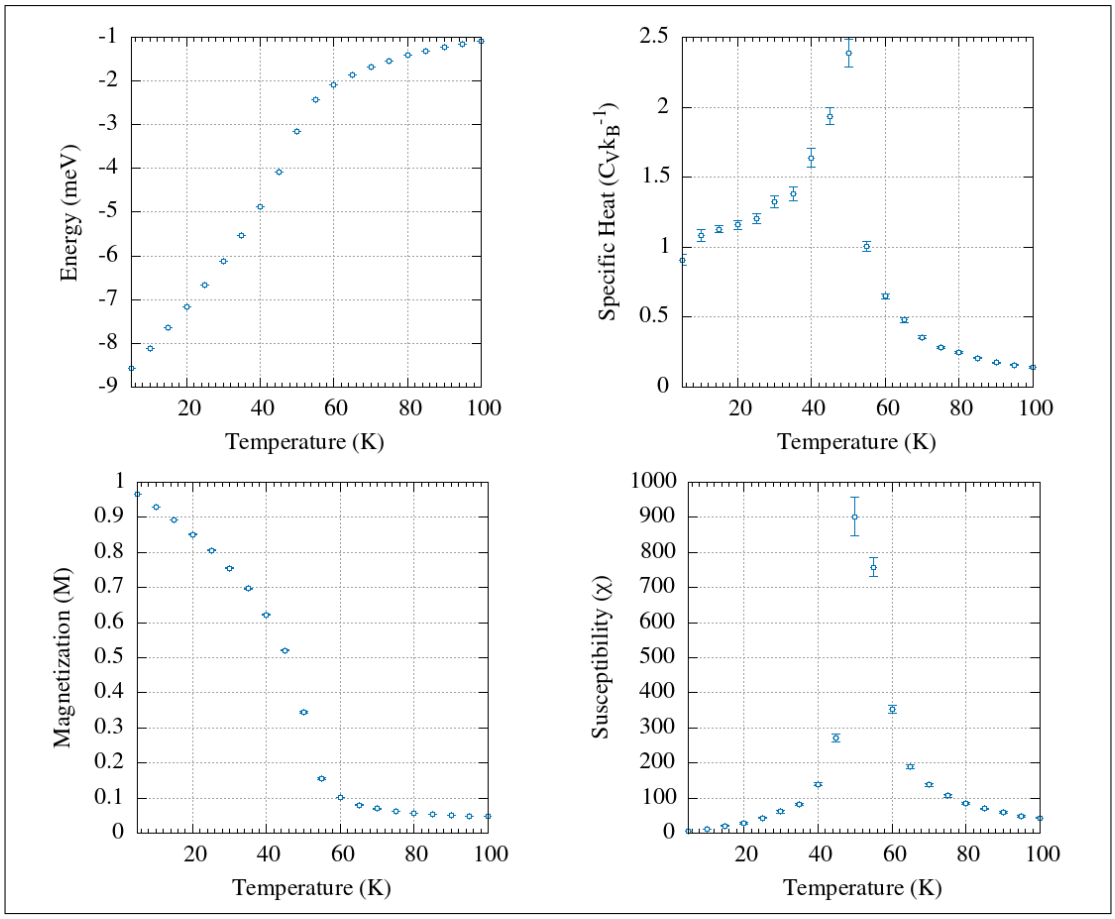

Figure 7: Presenting the 2×2 plots generated by *graph.sh* file in which the data points with error bars have been displayed. *see:* /examples/Simple_Cubic/data/graph.sh

7.1.1 *in_vec.ether*

This input file should be present before running the *vector* program. The user must remove (if any) the older $< tt.tttt > spK.xsf$ file before running the *vector* program. It seeks only two pieces of information whose descriptions are as follows,

1. Total number of ions which we want to include for vector plotting
2. Label of ions, must be present in the '*structure.vasp*' file, separated by blank space.

A typical example of this file is given below,

```
==============================================
3           ! total number of ions to include, First line
A B C       ! their ionic symbols, Second line
==============================================
```

7.1.2 <species_name>_vec

Files with the name *<species_name>_vec* will be generated after the execution of *./vector*, such that label <species_name> will be replaced as per the species symbols mentioned in second line of *in_vec.ether* file e.g., *A_vec, B_vec,* and *C_vec* if *A, B, C* ions are listed in '*structure.vasp*' file.

7.1.3 *plot_vector.sh*

This is the *gnuplot* script file and will be generated after the *vector* program completion. The content of this file will also appear on the screen, and the user may directly copy this content and paste it directly into the *gnuplot* terminal for 3D visualization (figure 8) of vectors on the unit sphere.

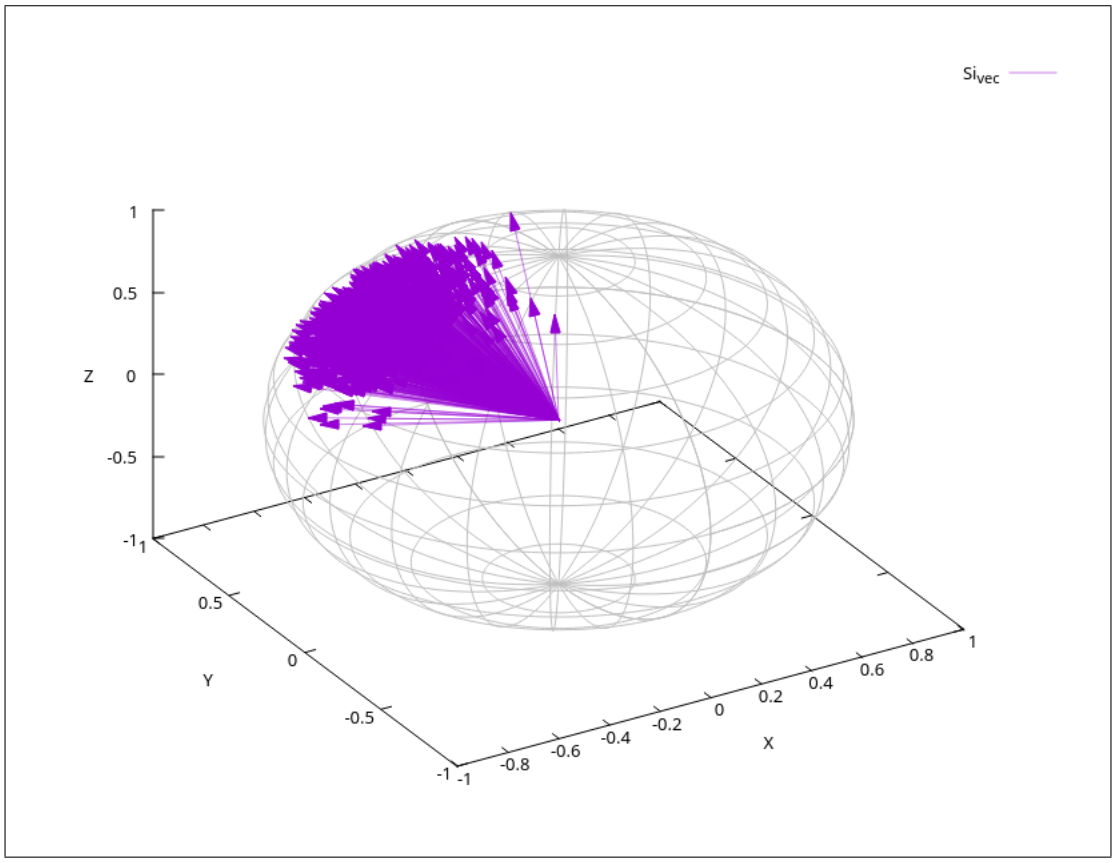

Figure 8: Using vector program the ground spin state configurations, observed at 5 Kelvin, are plotted on unit sphere. *see:* /examples/Simple_Cubic/spin/5.0000spK.xsf

7.1.4 *sp_vector.xsf*

This file is the exact copy of the provided $< tt.tttt > spK.xsf$ file and is used by *vector* program internally.

7.2 Structure Factor

Magnetic structures (commensurate/incommensurate) and associates to the magnetic wave vector (q) are crucial in understanding the non-collinear magnetic aspects. Evaluating the structure factor (S_f), defined as the thermal average of the Fourier-transformed spin-spin correlation, can help distinguish any complex magnetic phase transitions as the function of temperature.

Using utility tool '*structure_factor.f*90', (/utility/Vector/structure_factor.f90), *user* can generate the heat map plot for S_f as the functions of q and temperature using the relation as follows.

$$S_f(q) = \frac{1}{N} \sum_{r,r'} \langle S_r S_{r'} \rangle \, e^{iq(r-r')} \tag{16}$$

Since q is proportional to $\frac{1}{L}$, L: lattice parameter, spins arranged in simple cubic lattice network with ferromagnetic and anti-ferromagnetic interactions will show S_f maxima at the integer multiple of q and $\frac{q}{2}$, respectively. Figure 9 validates such theoretical perceptions very clearly.

Figure 9: Structure factor and low-temperature final spin vectors using *Ether* have been demonstrated here. Plots depicted in the left and right panels are associated with the ferromagnetic and anti-ferromagnetic cases, respectively. *see:* /examples/Simple_Cubic/Structure_Factor/

COMMAND $\Longrightarrow$ gfortran -o sf structure_factor.f90

RUN $\Longrightarrow$ ./sf

*structure_factor.f*90 requires two input files: '*in_SF.nlime*' and ground spin states (*gss.dat*) for structure factor evaluations.

7.2.1 *in_SF.nlime*

File *in_SF.nlime* contains the necessary information which are as follows,

1. Direction in which we want to calculate the structure factor.
2. Latency (in terms of degree) used to calculate spin presents in the provided *directions*.
3. q value extension: q_{final}, $q_{initial}$, dq
4. Scaling use to scale the length of lattice parameters.

A typical example of this file is given below,

```
==========================================
0 0 1          ! Direction
1              ! Least latency in pattern
2 0.01 0.01    ! Magnetic wave vector: q(to), q(from), q(step size)
0.5 0.5 0.5    ! (a, b, c), use to scale the Lattice parameters (a*L, b*L, c*L)
==========================================
```

7.2.2 *plot_sf.sh*

It is an output file generated by *structure_factor.f*90 containing the necessary *gnuplot* script to plot the heat map plot and save it into the file name *figSF.png*. Kindly note that the above script file seeks output file *outputSF.dat* generated by *structure_factor.f*90 tool. To plot, please follow the given command in terminal,

COMMAND $\Longrightarrow$ gnuplot plot_sf.sh