Uppsala Atomistic Spin Dynamics User Guide

Johan Hellsvik, Björn Skubic and Andrea Taroni

March 9, 2011

1 Introduction

This user guide describes the essential features of Uppsala Atomistic Spin Dynamics program (UppASD). The emphasis is on the input files necessary to run calculations using UppASD and the output files it generates. Some information concerning the analysis of data generated by the code is also given. The ASD method and our implementation of it is described in the article by Skubic *et al.* [1]. The underlying theory is described in the articles by Antropov *et al.* [2] and García-Palacios and Lázaro. [3] An old, yet remarkably lucid overview is also given by Watson *et al.* [4]

The program evolves the equations of motion for atomic magnetic moments in a solid. These take the form of the Landau-Lifshitz-Gilbert (LLG) equation,

$$\frac{d\mathbf{m}_i}{dt} = -\gamma \mathbf{m}_i \times [\mathbf{B}_i + \mathbf{b}_i(t)] - \gamma \frac{\alpha}{m} \mathbf{m}_i \times (\mathbf{m}_i \times [\mathbf{B}_i + \mathbf{b}_i(t)]). \tag{1}$$

In this expression, γ is the gyromagnetic ratio and $\mathbf{b}_i(t)$ is a stochastic magnetic field with a Gaussian distribution, the magnitude of which is related to the damping parameter α, which eventually brings the system into thermal equilibrium. The typical time step for solving the differential equations is $\Delta t = 0.1$ femtoseconds, *i.e.* 10^{-16} seconds.

The effective field, $\mathbf{B}_i$, experienced by each atom i is given by

$$\mathbf{B}_i = -\frac{\partial \mathscr{H}}{\partial \mathbf{m}_i}, \tag{2}$$

where $\mathscr{H}$ is the exchange Hamiltonian,

$$\mathscr{H} = -\frac{1}{2} \sum_{i \neq j} J_{ij} \mathbf{m}_i \cdot \mathbf{m}_j, \tag{3}$$

in which i and j are atomic indices, and J_{ij} is the strength of the exchange interaction. This may be obtained from first-principles calculations, and often even inferred from experiment. It is also possible (and in many cases even essential) to include other terms to the Hamiltonian, such as anisotropies and external magnetic fields. As discussed below, the code is able to implement these with a high level of flexibility.

2 License

The UppASD code is developed by the Division of Materials Theory, in the Department of Physics and Astronomy at the University of Uppsala, Sweden. It is not (yet) open source software. Further information concerning the license and contact information of the developers may be found on the UppASD webpage: http://www.physics.uu.se/en/page/UppASD. The current version of the code (3.0) is still under active development.

3 Installation

The source code is distributed along with documentation and a growing set of examples. To install, uncompress the file `UppASD.tar.gz`, enter the `source` directory, and compile with the supplied Makefiles. If possible, compilation with the Intel Fortran compiler is recommended, using the command

```
make -f MakefileIfort
```

Other than the `source` directory, the current UppASD distribution also contains an `examples` directory, and a `docs` directory, in which this user manual is located, along with some programming notes for developers (`UppASDnotes.tex`).

4 Principles of the Code

When run, UppASD essentially goes through three stages:

1. Initialization: the system is set up.
2. Initial phase: an optional stage in which the system is brought into thermal equilibrium, with limited data sampling.
3. Measurement phase: the system is evolved in time, with complete data sampling being made.

During the initialization phase, all the parameters necessary to describe the system of interest, such as its geometry, dimensions, exchange couplings and boundary conditions, are set up. In addition, the initial phase also sets the simulation parameters, such as the number of simulation steps to record data over, which SDE solver to use, and the temperature at which the simulation should be run.

The initial phase, which is optional, is typically performed in order to bring the system into thermal equilibrium, so that the data recorded in the measurement phase is for a thermalized system. Obviously, if one is interested in out-of-equilibrium dynamics, then there is no need to perform this phase. The initial phase can either be performed using Spin Dynamics (SD), or a Monte Carlo (MC) algorithm using Metropolis dynamics. [5] The latter is convenient for ground state searches, provided the system is not too complex (*i.e.* a spin glass).

During the measurement phase, the data sampling is performed. Simulations can be run in either MC or SD mode. In MC mode only magnetization averages may be measured. In SD mode, a much richer set of observables are measured, including spin correlations and dynamics structure factors.

5 Input

A file with the hardcoded name `inpsd.dat` is the main input file necessary to run UppASD. Contained in this file are also the names of the files containing the exchange interactions, the atomic positions, and the atomic moments. Although the names of these files is arbitrary, in this manual they are referred to as the `jfile`, `posfile` and `momfile`, respectively. Other optional files containing information such as the uniaxial anisotropy and the Dzyaloshinskii-Moriya vectors may also be included, as described below.

The input format is keyword based. The code is programmed to search for given keywords, and then read in the values that follow. If no keyword is given, a default value is set. An example `inpsd.dat` file for bcc Fe is shown below:

```
simid bccFe100
ncell     10        10        10        System size
BC        P         P         P         Boundary conditions
cell      1.00000   0.00000   0.00000
          0.00000   1.00000   0.00000
          0.00000   0.00000   1.00000
Sym       1                             Symmetry of lattice

posfile   ./posfile
momfile   ./momfile
exchange  ./jfile

SDEalgh   1                             SDE solver
Initmag   1                             Initial config of moments

Temp      100.000         K             Temperature of the system
damping   0.1                           damping parameter (alpha)
Nstep     10000                         Number of time steps
timestep  1.000e-16       s             Time step size

do_avrg   Y                             Measure averages

do_tottraj Y                            Measure moments
tottraj_step   100
```

While the meaning of most of the entries in this particular example should be obvious, it is also clear that more information must be read in from external files for the system to be fully defined. These are:

posfile

The positions of the atoms in the unit cell are given in basis vector coordinates. These can be listed directly in the `inpsd.dat` file, but in our examples they are oftened read in from the `posfile`. For a bcc structure:

```
1 1   0.000000  0.000000  0.000000
2 1   0.500000  0.500000  0.500000
```

The first entry indicates the atom number, whereas the second one indicates the atom type. In this specific case there is only one atom type, namely Fe.

momfile

This file lists the magnetic moments of the atoms in the unit cell. Also, if the `initmag` entry is set to 3, the initial direction of the moments is read from this file. For bcc Fe:

```
1 1 2.23 1.0 0.0 0.0
2 1 2.23 1.0 0.0 0.0
```

The first entry indicates the atom number, the second entry the atom type, and the third entry indicates the magnitude of the magnetic moment in μ_B. The last three entries indicate the x, y, and z components of the moment (assuming `initmag` is set to 3).

exchange

This file lists the exchange couplings within the system. The content of this file depends somewhat on the symmetry of the system, and the number of atom types present. For the present bcc Fe example:

```
1 1  0.5       0.5       0.5        1.33767
1 1  1.0       0.0       0.0        0.75703
1 1  1.0       1.0       0.0       -0.05975
1 1  1.5       0.5       0.5       -0.08819
```

The first two entries indicate atom type i and j. The third, fourth and fifth entries specify the coordination shell in basis vector coordinates. The last entry is the exchange energy in mRy. This `jfile` lists the J_{ij} values for the first four coordination shells, and assumes a cubic symmetry (`sym` is set to 1).

In systems with more than one atom type, the exchange parameters need to be defined between all atoms (*i.e.* for 2 atoms the interactions are between the 1-1, 1-2, 2-1 and 2-2 atoms). Note also that if no symmetry is assumed (`sym` is set to 0), then the J_{ij} parameters need to be specified for all neighbours, even those belonging to the same coordination shell.

5.1 Input Entries

The following entries are currently implemented in UppASD. Where applicable, the default entry setting is underlined.

simid The 8 character long simulation id.

cell The three lattice vectors describing the cell.

ncell Number of repetitions of the cell.

bc Boundary conditions (P=periodic, 0=free).

natoms Number of atoms in one cell.

ntypes Number of types atoms in one cell.

sym Flag to determine the assumed symmetry of the system (0=none, 1=cubic, 2=2d cubic (in xy plane), 3=hexagonal).

maptype Flag to change between 1=cartesian or 2=BGFM style for exchange coupling vectors.

posfile External file for the positions of the atoms in one cell, accompanied with the site number and type of the atom.

momfile External file for magnitudes and directions of magnetic moments.

exchange External file for Heisenberg exchange couplings.

dm External file for Dzyaloshinskii-Moriya (DM) exchange coupling, which takes the form

$$\mathcal{H}_{\mathrm{DM}} = \sum_{i,j} \mathbf{D}_{ij} \left(\mathbf{m}_i \times \mathbf{m}_j \right), \tag{4}$$

where $\mathbf{D}_{ij}$ is the DM vector. The format is similar to that of the exchange file, *i.e.* in a 2d square lattice it may look something like:

```
1 1  1.0000  0.0000 0.0000  2.50000  0.00000  0.00000
1 1 -1.0000  0.0000 0.0000 -2.50000 -0.00000 -0.00000
1 1  0.0000  1.0000 0.0000  0.00000  2.50000  0.00000
1 1  0.0000 -1.0000 0.0000 -0.00000 -2.50000 -0.00000
```

The first two entries indicate the site number and the type of atom, respectively. The third to fith entries indicate the components of the DM vector $\mathbf{D}_{ij}$, and the final three entries indicate the strength of these components. Note that in this case the `maptype` flag has been set to 2 in the input file.

anisotropy External file for anisotropy strengths and directions. The single-ion, or uniaxial, anisotropy is defined as

$$\mathcal{H}^{\mathrm{U}}_{\mathrm{ani}} = \sum_i K^{\mathrm{U}}_1 (\mathbf{m}_i \cdot \mathbf{e}_i)^2 + K^{\mathrm{U}}_2 (\mathbf{m}_i \cdot \mathbf{e}_i)^4, \tag{5}$$

where K_1 and K_2 are the strength of the linear and four-fold term along an axis with direction $\mathbf{e}_i$. In a cubic system, one must also define the so-called cubic anisotropy, given by

$$\mathcal{H}^{\mathrm{C}}_{\mathrm{ani}} = \sum_i K^{\mathrm{C}}_1 (m^2_{i,x} m^2_{i,y} + m^2_{i,y} m^2_{i,z} + m^2_{i,z} m^2_{i,x}) + K^{\mathrm{C}}_2 m^2_{i,x} m^2_{i,y} m^2_{i,z}, \tag{6}$$

where $\mathbf{m} = (m_x, m_y, m_z)$. UppASD is able to read in either Eq. (5) or Eq. (6), or even both. For bcc Fe, a plausible `kfile` might be:

```
1   7   -0.020    0.000    0.0    1.0    0.0    0.1
2   7   -0.020    0.000    0.0    1.0    0.0    0.1
```

The first entry lists the atom number, whereas the second entry indicates if the uniaxial (`1`), cubic (`2`) or both (`7`) anisotropies are to be mounted. The second and third entries list the strength of K_1 and K_2, respectively. The fifth to seventh entries indicate the components of the vector $\mathbf{e}_i$. Finally, in the instance of the second entry being set to `7`, the final entry indicates the ratio between K^{U} and $K^{\mathrm{C}}_{\mathrm{ani}}$

do_jtensor (0=off/1=on). This switch allows the exchange data to be read in according to the tensorial representation of the Heisenberg Hamiltonian, as implemented in the Vienna-Budapest SKKR code. [6] In this case, the exchange Hamiltonian is defined as

$$\mathcal{H}_{\mathrm{Tens}} = \frac{1}{2} \sum_{i,j} \mathbf{m}_i \mathcal{J}_{ij} \mathbf{m}_j. \tag{7}$$

Here, $\mathcal{J}_{ij} = J_{i,j}\mathcal{I} + \mathcal{J}^S_{i,j} + \mathcal{J}^A_{i,j}$ is a 3×3 tensor (in which $\mathcal{I}$ is the unit matrix), the trace of which is equal to the exchange constant as defined in Eq. (3) by [6]

$$J_{ij} = \frac{1}{3} \mathrm{Tr}(\mathcal{J}_{ij}). \tag{8}$$

In this formalism, the off-diagonal terms of the tensor are proportional to the components of the DM vector $\mathbf{D}_{ij}$ in Eq. (4), as $D^x_{ij} = \frac{1}{2}(J^{yz}_{ij} - J^{zy}_{ij})$, $D^y_{ij} = \frac{1}{2}(J^{xz}_{ij} - J^{zx}_{ij})$ and $D^z_{ij} = \frac{1}{2}(J^{xy}_{ij} - J^{yx}_{ij})$. In order to define the first shell of exchange parameters in bcc Fe using this formalism, the exchange file would be changed to look as follows:

```
0 0 1 2 0.00134 0.0 0.0 0.0 0.00134 0.0 0.0 0.0 0.00134
0 0 2 1 0.00134 0.0 0.0 0.0 0.00134 0.0 0.0 0.0 0.00134
```

NB: `maptype` must be set to 2 in order to use the tensorial format. In addition, since SKKR prints the exchange in Ry, UppASD reads this input in Ry and not in mRy as usual.

do_dip Flag for enabling dipole-dipole interactions (0=off/1=on).

do_prnstruct Flag to print lattice structure (0=off/1=on). Useful for checking if the system geometry and couplings are correctly set up.

do_ralloy Flag to set if a random alloy is being simulated (0=off/1=on).

aunits Implement atomic units, *i.e.* set k_B, $\hbar$, ... = 1 (Y/N). If this is switched on, the `timestep` in SD mode should be roughly $0.1J_{ij}$.

sdealgh Switch for choosing SDE solver (1=Midpoint, 4=Heun_proper, 5=Depondt). The default option runs the semi-implicit midpoint solver developed by Mentink *et al.* [7] In this case, the `timestep` can be as large as 10^{-16} seconds, but this should *always* be checked.

mensemble Number of ensembles to simulate. The default value is 1, but this may be increased to improve statistics, especially if investigating laterally confined systems, such as chains.

tseed Random number seed for the stochastic field simulating the fluctuations due to temperature. Default value is 1.

initmag Switch for setting up the initial configuration of the magnetic moments (1=Random distrubution, 2=Cone, 3=aligned along direction defined in momfile, 4=Read from restartfile).

restartfile External file containing stored snapshot from previous simulation (used when initmag=4). The format coincides with the format of the output file `restart.simid.out`.

mseed Random number seed for magnetic moments if initmag=3. Set to 1 by default.

theta0 If `initmag`=3, the magnetic moments are randomly distributed in a cone prescribed by this angle, and `phi0`. Set to 0 by default.

phi0 Cone angle for initmag=2. Set to 0 by default.

roteul Perform global rotation of magnetization. Set to 0 by default.

rotang Euler angles describing the rotation if roteul=1.

ip_mode Mode for initial phase run (S=SD, M=Monte Carlo, N=none).

ip_temp Temperature for initial phase run.

ip_hfield External applied field for initial phase run. This is given in cartesian coordinates, *e.g.*,

```
hfield   1.0   0.0   0.0
```

ip_mcnstep Number of Metropolis sweeps over the system if ip_mode=M.

ip_damping Damping parameter α for SD initial phase. Default value is 0.05.

ip_nphase Number of initial phases to be done with SD. This must be followed by `ip_nphase` lines containing number of steps and timestep for each phase.

mode Mode for measurement phase run (S=SD, M=Monte Carlo).

temp Temperature for measurement phase.

hfield External applied field for measurement phase.

mcnstep Number of Metropolis sweeps over the system if mode=M.

damping Damping parameter α for SD measurement phase. Default value is 0.05.

timestep Time step between SD iterations. Unless `aunits`=Y, this should be set to a value between 10^{-17} and 10^{-15} seconds, depending on the system and SDE solver.

do_avrg Sample and print average magnetization, and its higher order moments. (Y/N).

do_proj_avrg Sample and print type (*i.e.* sublattice) projected average moments. (Y/N).

do_projch_avrg Sample and print chemical (*i.e.* sublattice) projected average moments. (Y/N).

avrg_step Time step between sampling of averages. Set to 100 by default.

avrg_buff Number of samplings of averages to buffer between printing to file. Set to 10 by default.

do_tottraj Sample and print all trajectories (moments) in the system. (Y/N). Generates the (rather large) `moment.simid.out` file.

tottraj_step Time step between samplings of moments. Set to 1000 by default.

tottraj_buff Number of samplings of moments to buffer between printing to file. Set to 10 by default.

ntraj Number of individual trajectories to sample and print. Followed by `ntraj` lines describing atoms to sample, time step between samples and steps to buffer. Set to 0 by default.

set_bpulse Add magnetic field pulse (0=no, $1-4$ for different shapes)

plotenergy Flag to enable the calculation of the energy of the system projected to the different components of the Hamiltonian. (0=off/1=on).

set_landeg Flag for assigning different values of the gyromagnetic factor for the moments. Set to 0 by default.

do_sc Flag to determine if spin correlations should be analysed (Y=yes, N=no, C=G(r) only). Setting this flag to Y measures the space- and time-displaced correlation function,

$$C^k(\mathbf{r}-\mathbf{r}',t) = \langle m_{\mathbf{r}^k}(t) m_{\mathbf{r}'^k}(0)\rangle - \langle m_{\mathbf{r}^k}(t)\rangle\langle m_{\mathbf{r}'^k}(0)\rangle, \tag{9}$$

where the angular brackets signify an ensemble average and k the cartesian component, and its Fourier Transform, the dynamical structure factor

$$S^k(\mathbf{q},\omega) = \frac{1}{\sqrt{2\pi}N}\sum_{\mathbf{r},\mathbf{r}'} e^{i\mathbf{q}\cdot(\mathbf{r}-\mathbf{r}')}\int_{-\infty}^{\infty} e^{i\omega t} C^k(\mathbf{r}-\mathbf{r}',t)dt, \tag{10}$$

which yields information concerning the spin waves present in the system. [8] The quality of the measured data depends sensitively on the values of `sc_nstep`, `sc_step`, `sc_navrg`, `sc_nsep` and `sc_volume`. If the flag is set to C, only the static correlation function is measured.

do_sc_proj Flag to determine if type projected spin correlation should be analyzed (Y=yes, N=no). This works only if `sc_mode`=2.

sc_mode Flag to determine when to transform the spin correlations (0=in memory, 1=in scratch file, 2=on the fly). Options 0 and 1 generate enormous files.

sc_nstep Number of steps to sample.

sc_step Number of time steps between each sampling.

sc_navrg Number of spin correlation measurements to average over.

sc_nsep Number of time steps between the start of subsequent spin correlation measurements.

sc_volume Sub-volume of system to sample the correlation functions over.

qpoints Flag for for generation of q-point mesh necessary for $S(q,\omega)$ calculations. (F=external file, A=automatic, C=full cell, D=external file with direct coordinates).

qfile External file containing the q point mesh for $S(q,\omega)$ calculations.

rdirfile External file containing directions to calculate $S(r,t)$ for.

do_autocorr Flag to enable autocorrelation sampling (Y=yes, N=no).

acfile External file containing waiting times for the autocorrelation measurements.

6 Output files

Depending on the settings chose in the input file, UppASD prints out a varying number of output files. These all share the suffix `.out`.

aniso1.simid.out

Occurs if the anisotropy is defined. Prints the anisotropy parameters for each atom in the format

$$e_x, e_y, e_z, k_1, k_2$$

where the three first entries are the direction of the anisotropy axis.

averages.simid.out

Occurs if measurement phase is run in SD mode. Prints out the average magnetization as a function of simulation time, in the format

$$step, m_x, m_y, m_z, m, m^2, m^4$$

where *step* is the simulation time expressed in terms of the number of time steps, m_x, m_y and m_z are the components of the intensive average magnetization (*i.e.*, $m_x = \frac{1}{N}\sum_i m_{x,i}$), $m = \sqrt{m_x^2 + m_y^2 + m_z^2}$, and so on.

coord.simid.out

Occurs if `do_prnstruct` is switched on. Prints out the coordinates of each moment in the system.

cumulants.simid.out

Prints out the running time averages of the intensive magnetization and its higher order moments, in the format

$$step, m_t, m_t^2, m_t^4, U_4$$

where, $U_4 = 1 - \frac{1m_t^4}{3m_t^2}$ is the fourth order 'Binder' cumulant, useful for estimating transition temperatures. [5]

dmdata.simid.out

Occurs if the DM interaction is defined. Prints out the DM vector components and their strengths for each atom.

inp.simid.out

Extensive output of the values assigned to global variables after reading `inpsd.dat` and accompanying files.

mcinitial.simid.out

Occurs if initial phase is set to MC mode. Prints out the final iterations of the MC initial phase, in the format

$$mcstep, m, U_4, \chi$$

where χ is the magnetic susceptibility. This is useful for checking whether or not the initial run has thermalized before entering the measurement stage.

mcmeasure.simid.out

Occurs if measurement phase is set to MC mode. Prints out the quantities measured in MC mode, using the same format used for `mcinitial.simid.out`.

moments.simid.out

Occurs if the `do_tottraj` flag is switched on. Prints the configuration of all magnetic moments at regular interval in time in the format

$$step, atom, m_{x,i}, m_{y,i}, m_{z,i}, m_i \tag{11}$$

Note that this file is very large. It is useful for creating animations of the evolution in time of the magnetic configuration of the system.

projavs.simid.out

Occurs of the `do_projavs` flag is switched on. Prints out the same thermodynamic averages printed in `averages.simid.out`, but projected to each atom type sublattice. The format is also identical to `averages.simid.out`, except for the addition of a column indicating the sublattice.

projsqt.simid.out

Occurs of the `do_sc_proj` flag is switched on. Prints out the same information printed in `sqt.simid.out`, but projected to each atom type present in the system. This file can be very large.

projsqw.simid.out

Occurs of the `do_sc_proj` flag is switched on. Prints out the same information printed in `sqw.simid.out`, but projected to each atom type present in the system. This file can be very large.

restart.simid.out

The magnetic configuration of the system at a specific point in time. Can be used as input when the `initmag` flag is set to 4.

trajectory.simid.out

The trajectory as a function of time step for an individual magnetic moment on format

$$step, atomnr, m_{x,i}, m_{y,i}, m_{z,i}, m_m, m_i$$

If the number `ntraj` is also defined to be greater than 1, the code prints out ntraj files named `trajectory.simid.ntraj.out`.

sqt.simid.out

Occurs of the `do_sc` flag is switched on. Prints out the time-resolved structure factor $S(q,t)$ in the format

$$nstep, nq, \mathrm{Re}[S^x(q,t)], \mathrm{Im}[S^x(q,t)], \mathrm{Re}[S^y(q,t)], \mathrm{Im}[S^y(q,t)], \mathrm{Re}[S^z(q,t)], \mathrm{Im}[S^z(q,t)]$$

This file can be very large.

sqw.simid.out

Occurs of the `do_sc` flag is switched on. Prints out the frequency-resolved structure factor $S(q,\omega)$ in the format

$$nq, q_x, q_y, q_z, nstep, S^x(q,\omega), S^y(q,\omega), S^z(q,\omega), S(q,\omega)$$

This file can be very large.

struct1.simid.out

Occurs if `do_prnstruct` is switched on. Prints out the structure and exchange couplings of the system. Handy for checking if the system is set up correctly.

struct.simid.out

Occurs if `do_prnstruct` is switched on. Also prints out the structure and exchange couplings of the system. Again, handy for checking if the system is set up correctly.

swdos.simid.out

Occurs of the `do_sc` flag is switched on. Prints out the $S(q,\omega)$ 'density of states' as a function of energy.

totenergy.simid.out

Occurs if the `plotenergy` flag is switched on. Prints out the total energy of the system as a function of time step.

7 Examples

The best way to learn the code is to work through the examples in the directory `examples`. The current set of examples illustrates the range of the functionality of UppASD, and gives a feel for the data analysis that is necessary in order to interpret the results of the simulations.

7.1 bcc Fe

The `bccFe` directory contains the necessary input in order to simulate bcc Fe. There are a couple of Matlab scripts enclosed that plot quantities such as the average magnetization as a function of time. By varying input parameters such as the temperature, one may get a feel for how these affect these observable quantities.

The `bccFe-Tsweep` directory contains very similar input to the `bccFe` directory. However, a small wrapper is included in order to run SD simulations as a function of temperature. This is often useful for checking the transition temperature of a given system.

7.2 Heisenberg Spin Chain

The `HeisChain` directory contains possibly the simplest system that can be set up with UppASD: a Heisenberg ferromagnetic chain. This example is set up to allow the spin wave spectrum of the system to be probed. Since the dispersion of this system can be derived analytically (as discussed any many textbooks concerning solid state physics), it is a handy example to get acquainted with spin correlations and dynamics in UppASD.

By copying over the `sqw.simid.out` and `inpsd.dat` files into the `Sqw` directory, and running the `sqw_map.sh` script, $S(q,\omega)$ as a function of q is plotted, giving rise to the spin wave displersion. The dynamics of spin waves have been studied in this way in the paper by Bergman *et al.* [8]

7.3 Two-dimensional Systems

The `2d-systems` directory contains a couple of two-dimensional systems: the surface simulating the Co/Cu(001) monolayer system (`fcc001`), and the square lattice Heisenberg system (sc, which stands for simple cubic). In the latter example, the flag aunits is switched on. By running the simulation as a function of temperature, the magnetization can be benchmarked against data obtained from the ALPS software package[1], which is contained in the scL64ALPS.dat file.

For both the fcc001 and sc examples, a directory Snapshots is included, that allows snapshots of the lattices to be taken. The scripts are written in VTK, and require the inp.simid.out, coord.simid.out and moment.simid.out files to work.

7.4 fcc Co

Another very common structure in solid state physics is the fcc structure. This example allows the measurement of the dyamical structure factor, $S(q, \omega)$, for fcc Co, as in the Heisenberg spin chain example.

7.5 GaMnAs

The GaMnAs directory contains input data that sets up a GaMnAs dilute magnetic semiconductor.

7.6 Random Alloy

The Ralloy directory contains input data necessary to set up a dilute random alloy.

7.7 SKKR Input (test)

The xctensortest directory contains the same data as in the bccFe directory, but set up in the tensorial format that arises from the Vienna-Budapest SKKR code. [6]

References

[1] B. Skubic, J. Hellsvik, L. Nordström, and O. Eriksson. A method for atomistic spin dynamics simulations: implementations and examples. J. Phys.: Condens. Matter, 20:315203, 2008.

[2] V. P. Antropov, M. I. Katsnelson, B. N. Harmon, M. van Schilfgaarde, and D. Kusnezov. Spin dynamics in magnets: Equation of motion and finite temperature effects. Phys. Rev. B, 54:1019, 1996.

[3] J. L. García-Palacios and F. J. Lázaro. Anisotropy effects on the nonlinear magnetic susceptibilities of superparamagnetic particles. Phys. Rev. B, 55:1006, 1997.

[1] http://alps.comp-phys.org

[4] R. E. Watson, M. Blume, and G. H. Vineyard. Spin motions in a classical ferromagnet. Phys. Rev., 181:811, 1969.

[5] K. Binder and D. P. Landau. A Guide to Monte Carlo Simulation in Statistical Physics. Cambridge University Press, Cambridge, 3rd edition, 2009.

[6] L. Udvardi, L. Szunyogh, K. Palotás, and P. Weinberger. First-principles relativistic study of spin waves in thin magnetic films. Phys. Rev. B, 68:104436, 2003.

[7] J. H. Mentink, M. V. Tretyakov, A. Fasolino, M. I. Katsnelson, and Th. Rasing. Stable and fast semi-implicit integration of the stochastic Landau-Lifshitz equation. J. Phys.: Condens. Matter, 22:176001, 2010.

[8] A. Bergman, A. Taroni, L. Bergqvist, J. Hellsvik, B. Hjörvarsson, and O. Eriksson. Magnon softening in a ferromagnetic monolayer: A first-principles spin dynamics study. Phys. Rev. B, 81:144416, 2010.