Spin-Glass Behaviour in Ordered Solids

Erik Karpelin

Supervisor: Anders Bergman
Subject reader: Oscar Grånäs

Bachelors thesis, 15hp - 1FA599
Uppsala University

July 12, 2023

Abstract

The spin-glass is a peculiar magnetic phase, exhibiting non-trivial dynamics at low temperatures, characterized by an continuously evolving state without long-range order. The behavior requires some degree of disorder to occur, often in the way of impurities or random exchange energy between the spins. However, recent research have found structurally ordered systems exhibiting glassy behaviour. This project aims to further investigate these self-induced spin-glasses. The report provides a short introduction to atomistic spin-dynamics and applies the theory to study self-induced spin-glasses in hexagonal systems with the help of simulations. A variation approach was applied by running simulation using a range of spin-exchange couplings in the Heisenberg Hamiltonian. These systems were then studied by the means of their autocorrelation function and compared to known glassy systems from the Edwards-Andersson model. The resulting behaviour is presented for three different hexagonal structures and glassy behaviour is indicated in stacked hexagonal systems. It is however argued that the autocorrelation function is not sufficient to classify these systems, instead further observables are needed. Nevertheless, the method of studying self-induced spin-glasses by varying couplings in the Heisenberg Hamiltonian is promising. As even with the few spin interactions used in this report we observe the slow relaxation time associated with spin-glasses. Given some extra considerations when choosing the exchange used for the simulation, a self-induced glassy state should be able to be recreated using the method described in this report.

Sammanfattning

Spinn-glas är en speciell magnetisk fas som uppvisar icke trivial dynamik vid låga temperaturer, en kontinuerlig utveckling samt en avsaknad av ordning på stora skalor. Detta beteende kräver en viss grad av oordning för att uppstå, ofta i form av föroreningar i materialet eller slumpmässiga interaktioner mellan olika spinn. Forskning har dock visat att även strukturellt ordnade system kan uppvisa spinn-glas beteende. Därmed är målet med detta projekt att fortsätta undersöka dessa själv-inducerade spinn-glas. Rapporten ger en kort introduktion till atomistisk spinn-dynamik och applicerar denna teori för att studera själv-inducerade spin-glas i hexagonala system. I projektet simulerades system med varierande spin-interaktioner i Heisenberg Hamiltonianen. Dynamiken undersöktes med hjälp av en korrelationsfunktion som jämfördes mot kända spinn-glas från Edwards-Andersson modellen. Resultat presenteras för tre hexagonala strukturer och spinn-glas-liknande beteende observeras i de tre-dimensionella systemen. Det kan dock argumenteras att korrelationsfunktionen inte är tillräcklig för att klassificera dessa system och att mer kvantitativa mått krävs. Trots detta anses metoden, att variera spinn-interaktioner i Heisenberg Hamiltonianen, vara lovande. Detta eftersom den långa avslappningstiden associerad med spinn-glas påträffades, trots de få interaktioner som användes i denna rapport. Ett själv-inducerat spin-glass borde därmed kunna skapas med de metoder som presenteras i rapporten, givet en mer systematisk metod vid val av interaktionsparametrar för simuleringen.

Contents

1 Introduction **1**
1.1 The spin-glass phase 1
1.2 Motivation and report structure 2

2 Theory **3**
2.1 Many-particle systems and DFT 3
2.2 The Stochastic Landau-Lifshitz-Gilbert Equation 4
2.2.1 Precessional motion 5
2.2.2 Damping motion 6
2.2.3 Numerical solutions 7

3 System structure **8**
3.1 The magnetic Hamiltonian 8
3.2 Crystal structure 10
3.2.1 Hexagonal plane 10
3.2.2 Hexagonal compact packing 11
3.2.3 Double hexagonal compact packing 11

4 Spin-glasses and correlation functions **12**

5 Methodology **13**
5.1 Simulation configuration 13
5.2 System configurations 14
5.2.1 Hexagonal plane 14
5.2.2 Hexagonal compact packing 14
5.2.3 Double hexagonal compact packing 15

6 Results **16**
6.1 HEX - systems 16
6.2 HCP - systems 18
6.3 DHCP - systems 19

7 Discussion **21**
7.1 Investigated systems 21
7.2 Improvements to analysis 22

8 Conclusion **24**

Appendix **27**
A UppASD 27
B Code 27
C Input files 32

1 Introduction

1.1 The spin-glass phase

A spin-glass (SG) is a most curious magnetic phase of matter, exhibiting properties not commonly found in solid materials. It can be defined as a system of spin, or atomistic magnetic moments, which undergoes a phase transition to a frozen, disordered state at low temperatures [1, Ch. 1]. Apart from this phase transition, the system also exhibits two other qualities of interest; a lack of long range magnetic order and a property known as aging [2]. The former provides a clue to the origin of the glass name of the state, borrowing the word from silica *glass* which possess the same lack of long range order, however only in the structural sense. The latter characteristic is more indigenous to SGs and refers to the long relaxation time of the system together with a continuous reorganization of the magnetic structure [3]. This property is partly due to the large degeneracy of ground-states for the spin-glass phase [4]. To create such a system one needs to fulfill a few criteria. Some sort of competing interaction, known as frustration, needs to be present, and a certain degree of random behaviour in the interactions of the spins [1, Ch. 1]. These criteria can be fulfilled experimentally by diluting noble metals with transition metal impurities [1, Ch. 1], creating both structural and magnetic disorder, or through simple mathematical models. Spin-glasses are therefore good candidates for numerical studies. As an example, the Edwards-Andersson (EA) model [5] uses normally distributed random variables in the interaction between the spins, creating the necessary conditions for a SG to occur. A illustration of such a system, together with a ferromagnetic system, can be seen in Figure 1.

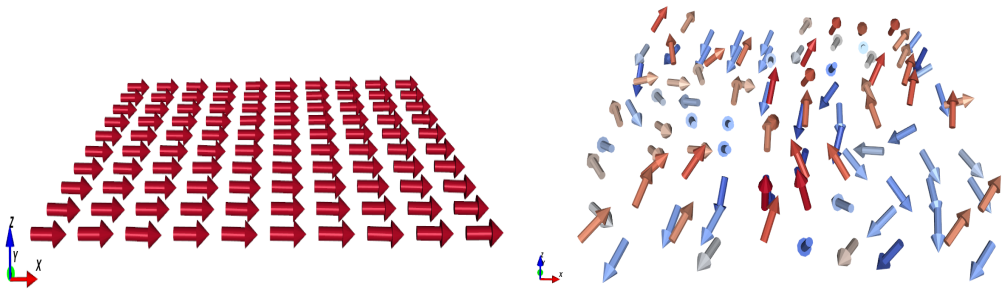


Figure 1: Schematic depiction of aligned spin in a ferromagnetic state (left) and a spin glass state (right). Both systems are considered to be at zero temperature.

1.2 Motivation and report structure

Since the discovery of the SG phase in 1972 [6] and the initial development of spin-glass theory in 1975 [7] the system has undergone a plethora of thorough investigations. This is partly due to it being an disordered system which is deceptively easy to model, as with the EA-model mentioned previously [3]. Furthermore, they have been of interest because of newly discovered applications, which provides a more ethically justifiable reason to study them. Some of these applications include error correcting codes [8] and the hot topic of artificial intelligence, where the theory of spin-glasses has been used to study the dynamics of deep neural networks [9]. Further studies have also shown that elemental neodymium can exhibit what is known as self-induced spin-glass behaviour at low temperatures [2]. These systems were studied with the help of simulations which uses atomistic spin dynamics together with exchange calculated from density functional theory. This implies that spin-glasses can be found in systems normally possessing long range magnetic and structural order, contradicting one defining characteristic of the SGs. The question is then raised, if other self induced systems can be found. This is the inquiry this report will try to answer.

The aim is henceforth to find these self-induced glassy states by varying exchange between atomic, magnetic moments and studying a time dependent correlation function, which relates the spin at a site with itself after a certain waiting time [10, Ch. 7]. This will provide further insight into how and when self-induced spin-glass behaviour occurs. The investigation will be conducted on three different, hexagonal lattice structures and simulated using the spin-dynamics software UppASD [11]. The different exchange couplings will be chosen to lie within ranges normally observed in ordered solids.

The report is structured as follows. As spin-glasses continuously evolve they constitute a dynamical system. We therefore start by covering some introductory atomistic spin dynamics together with a background in density functional theory which is needed when studying systems of larger scales. Thereafter, we introduce the system structure, the magnetic Hamiltonian of our system and also an overview of crystal lattice structure to get some intuition of how the spin interact. With the theory covered, we move forward with the methodology which covers specifics of the procedure such as simulation times, temperatures and variational procedures. Lastly, we present the main finding of the investigation, analyze the results and discuss these. We then finish of with a conclusion and outlook for further studies.

2 Theory

This theoretical overview will cover the main aspects one has to consider when studying spin dynamics and magnetic properties from an atomistic viewpoint. We begin by covering the problem of multiple-particle systems in quantum mechanics. This naturally leads into the backbone of atomistic studies, density functional theory (DFT). Although the subject of DFT is well beyond the scope of this report we will aim to present the necessary material to gain some intuition here. From density functional theory we present how one could derive the main equation of study for atomistic spin dynamics, the stochastic Landau-Lifshitz-Gilbert (SLLG) equation [10, Ch. 4]. Lastly, we cover some numerical methods for solving the SLLG, we will however start in the world of quantum mechanics.

2.1 Many-particle systems and DFT

In fundamental quantum mechanics the main hurdle to understand a system is to find its associated wave function $|\Psi\rangle$. This is done by solving the Schrödinger equation

$$i\hbar\frac{\partial}{\partial t}|\Psi\rangle = \mathcal{H}|\Psi\rangle,$$

which is a N dimensional, second order, partial differential equation where N is the degree of freedom for the system. Suppose we could solve this equation, we then require a Hamiltonian which for a system of many particles, interacting only via Coulomb interaction, have the form

$$\begin{aligned}\mathcal{H} = &-\frac{\hbar^2}{2}\sum_I \frac{\nabla_I^2}{M_I} + \frac{1}{2}\sum_{I\neq J}\frac{1}{4\pi\epsilon_0}\frac{Z_I Z_J e^2}{|\mathbf{R}_I - \mathbf{R_J}|} - \frac{\hbar^2}{2m}\sum_i \nabla_i \\ &+ \frac{1}{2}\sum_{i\neq j}\frac{1}{4\pi\epsilon_0}\frac{e^2}{|\mathbf{r}_i - \mathbf{r_j}|} - \sum_{i,I}\frac{1}{4\pi\epsilon_0}\frac{Z_I e^2}{|\mathbf{r}_i - \mathbf{R_I}|}.\end{aligned}$$

Here M_I is the mass of atmoic core I, Z_I the mass number, m the electron mass and finally $\mathbf{R_I}$ and r_i the nucleus and electron coordinates. This can be simplified using Hartree atomic units, $e = m = \hbar = 4\pi\epsilon = 1$, and the Born-Oppenheimer approximation. The latter assumes that the cores of the particles do not move much relative to the electrons, we then get a simplified version

$$\hat{\mathcal{H}} = -\frac{1}{2}\sum_i \nabla_i^2 + \frac{1}{2}\sum_{i\neq j}\frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} - \sum_{i,I}\frac{Z_I}{|\mathbf{r}_i - \mathbf{R}_I|}.$$

The only piece now missing is the wave function which can be provided using, for example, Hartree-Fock theory. This approach is however quite time

consuming, see [10, Ch. 1]. For this reason we can instead study density functional theory. In the DFT regime we shift our focus from the wave function to a more physical quantity of our system, the electron and charge density $n(\mathbf{r})$. The theory is based on a couple of theorems [10, Ch. 1] which in the Hohenberg-Kohn formulation take the form:

Theorem 1: *The total energy of system is a unique functional of the ground state electron density.*

Theorem 2: *The exact ground state density minimizes $E[n(\mathbf{r})]$ in the equation:*

$$E[n(\mathbf{r})] = T[n(\mathbf{r})] + W[n(\mathbf{r})] + V_{ext}[n(\mathbf{r})].$$

In short, these two theorems state that our charge density $n(\mathbf{r})$ uniquely describes our system and all information otherwise provided by the Hamiltonian and our wave-function Ψ is contained within it. The DFT method also comes with some computational advantages, as discussed in [12], which is of importance when investigating systems of many particles ($N > 10$), as is the case in this report. The DFT method is a powerful tool with wide application in physics and chemistry to the point that it has become the standard for ordered solids.

2.2 The Stochastic Landau-Lifshitz-Gilbert Equation

The governing equation of spin-dynamics is the stochastic Landau-Lifshitz-Gilbert Equation given by:

$$\boxed{\frac{d\mathbf{m}_i}{dt} = -\gamma_L \mathbf{m}_i \times [\mathbf{B}_i + \mathbf{B}_i^{fl}] - \gamma_L \frac{\alpha}{m_i} \mathbf{m}_i \times \{\mathbf{m}_i \times [\mathbf{B}_i + \mathbf{B}_i^{fl}]\}} \tag{2.1}$$

Here, γ_L is the renormalized gyromagnetic ratio and α being the Gilbert damping constant [10, Ch. 4]. Furthermore, $\mathbf{m}_i$ is the magnetic-spin moment at site i and $\mathbf{B}_i$ is the effective magnetic field with $\mathbf{B}_i^{fl}$ being a stochastic fluctuation field due to thermal properties of the material. The magnetic field can easily be obtained using the expression

$$\mathbf{B}_i = \frac{\partial \mathcal{H}}{\partial \mathbf{m}_i},$$

with $\mathcal{H}$ being the Hamiltonian of the system. The two terms constituting Eq. (2.1) have an intuitive physical meaning; the first describes the precessional motion of the spin and the second term convey the motion due to damping

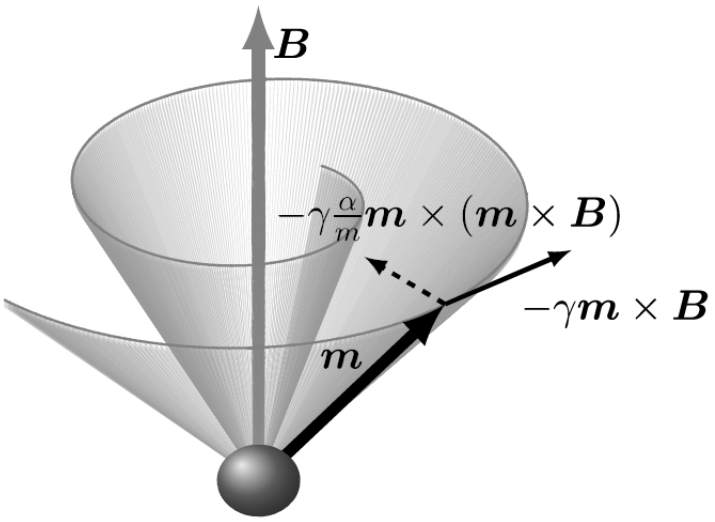


Figure 2: Figure depicting the motion described by the Landau-Lifshitz-Gilbert equation. As before, **m** denotes the atomic magnetic moment, the dashed vector represent the damping motion and the smaller bold arrow describes the precessional motion and B is an effective magnetic field. Figure used with permission from Dr. Danny Thonig.

[10, Ch. 4]. The resulting motion is well explained by Figure 2. The following section, will cover the origin of these two terms. The discussion will initially be restricted to the non-stochastic version of the equation, the *Landau-Lifshitz-Gilbert* equation

$$\frac{d\mathbf{m}_i}{dt} = -\gamma_L \mathbf{m}_i \times \mathbf{B}_i - \gamma_L \frac{\alpha}{m_i} \mathbf{m}_i \times \{\mathbf{m}_i \times \mathbf{B}_i\}, \tag{2.2}$$

using the same notation as for the SLLG. The stochastic part of Eq. (2.1), $\mathbf{B}_i^{fl}$, can later be added back as a perturbation to the magnetic field. It is also important to emphasize that this is not a rigorous derivation, instead it serves to provide some intuition to the origin of Eq. (2.1). Let us start with discussing the precessional term.

2.2.1 Precessional motion

We study the Hamiltonian

$$\mathcal{H}_{\alpha\beta}^{KS} = -\frac{1}{2}\nabla^2\delta_{\alpha\beta} + V_{\alpha\beta}^{eff}(\mathbf{r},t) + \left\{\frac{1}{2c}\hat{\boldsymbol{\sigma}}\cdot\mathbf{B}^{eff}(\mathbf{r},t)\right\}_{\alpha\beta}. \tag{2.3}$$

This operator is closely related to the Kohn-Sham Hamiltonian [10, Ch. 4], the only difference being the omitted spin-orbit coupling term. The subscript

α and β represent spin up and spin down, respectively, V^{eff} is an effective potential, and $\mathbf{B}^{eff}$ is an effective magnetic field as described previously. Lastly, $\hat{\boldsymbol{\sigma}}$ is a vector of the Pauli-matrices $\hat{\boldsymbol{\sigma}} = (\sigma_x, \sigma_y, \sigma_z)$ with

$$\sigma_x = \frac{1}{2}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y = \frac{1}{2}\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_z = \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

It may be shown that, the eigenstates of Eq. (2.3) is the charge and spin densities $n(\mathbf{r}, t)$ and $\mathbf{s}(\mathbf{r}, t)$, respectively [10, Ch. 4]. These physical quantities must be conserved and the corresponding continuity equations will then be

$$\begin{aligned} \frac{\partial n(\mathbf{r}, t)}{\partial t} + \nabla J_n &= 0 \\ \frac{\partial \mathbf{s}(\mathbf{r}, t)}{\partial t} + \nabla J_s + \gamma \mathbf{s} \times \mathbf{B}^{eff} &= 0, \end{aligned}$$

with ∇J being the divergence of the corresponding current due to the changing charge or spin, for further details see [10, p. 55] . If we omit the current term ∇J_s in the last equation and integrate the spin over a small atomic volume we can make the substitution $\mathbf{s}(\mathbf{r}, t) \rightarrow \mathbf{m}(\mathbf{r}, t)$ with $\mathbf{m}$ being the magnetic moment [10, Ch. 4]. We then have

$$\frac{\partial \mathbf{m}(\mathbf{r}, t)}{\partial t} = -\gamma \mathbf{m} \times \mathbf{B}^{eff}, \tag{2.4}$$

which gives the precessional part of Eq. (2.2). Although omitting the spin-current might seem arbitrary, it is motivated by the separation of time-scales of the electron motion and the spins [10, Ch. 4].

2.2.2 Damping motion

Although, the origin of the latter term is intuitively attributed to damping, the derivation is more ambiguous. We will base our discussion on [13] which takes a more classical approach than otherwise discussed in this report. Through Lagrangian mechanics one can show that the Euler-Lagrange equation, in functional form, for a system undergoing damped motion will be

$$\frac{d}{dt}\frac{\delta \mathcal{L}[\mathbf{m}, \dot{\mathbf{m}}]}{\delta \dot{\mathbf{m}}} - \frac{\delta \mathcal{L}[\mathbf{m}, \dot{\mathbf{m}}]}{\delta \mathbf{m}} + \frac{\delta \mathcal{R}[\dot{\mathbf{m}}]}{\delta \dot{\mathbf{m}}} = 0, \tag{2.5}$$

with $\mathcal{R}$ being the dissipative force causing the damping and $\mathcal{L} = T - U$ being the Lagrangian of the system. T here denotes the kinetic energy and U the potential, as per usual. As spin is a quantum mechanical property, we would now like to find corresponding quantum mechanical operators to these energies.

There is an issue though; although we can represent the potential U through a quantum operator, the same can not be said about the kinetic energy. As stated in [13], it seems impossible to find a corresponding quantum operator to the classical rotational kinetic energy of the spin. It can however be argued that for zero damping, Eq. (2.5) should correspond to the precessional term in Eq. (2.4). From this one can deduce that the damping term will have the form

$$\mathcal{R} = -\frac{\alpha}{m}\mathbf{m} \times \frac{\partial \mathbf{m}}{\partial t},$$

which is equivalent to the damping term given in Eq. (2.2). It should be noted here that $\mathcal{R}$ in the above expression is not equal to the damping term previously stated in Eq. (2.5). It is however more convenient to use the same notation.

2.2.3 Numerical solutions

We now shift our focus to solving Eq. (2.1). The equation is a stochastic differential equation (SDE) due to the temperature dependent magnetic field $\mathbf{B}_i^{fl}$ and in principle requires different treatment from that of an ordinary differential equation (ODE), such as Eq. (2.2). The analytical procedure will not be covered in this report, we will however present the numerical solution in short.

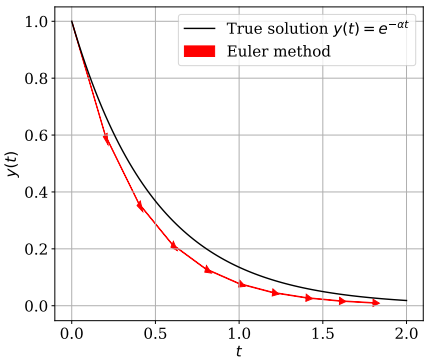


Figure 3: Euler method with a large step size for Eq. (2.6).

The basis for solving a SDE numerically is equal to that of the ODE, only with a stochastic extension. To exemplify this we will discuss the well-known Euler method. We start by posing a simple problem. Say we would like to solve the following first order ODE

$$\frac{dy}{dt} = -\alpha y(t) = f(t, y). \tag{2.6}$$

This can then be done using the Euler method as described in [14]. The iterative, numerical solution is then given by

$$y_{i+1} = y_i + hf(t_i, y_i),$$

with h being the step size in t. Given some initial conditions, $y(0) = y_0$, we can approximate the value of the solution for the following step, as in Figure

3. The same procedure can be extended to stochastic differential equations. We study the SDE

$$\frac{\partial Y}{\partial t} = f(\{Y\}, t) + \sum_{j}^{n} g_j(\{Y\}, t)\Gamma_j(t).$$

The first term is attributed to the deterministic drift, in the same way as $f(t, y)$ in Eq. (2.6), and the latter term is due to diffusion and is stochastic. The analog to the Euler method presented above, in the stochastic case [10, Ch. 5], is given by

$$Y_{i+1} = Y_i + hf(\{Y\}, t) + \sum_{j}^{n} g_j(\{Y\}, t)dW_j,$$

with dW_j being a normally distributed increment of the Wiener process on the interval h [15]. Numerically, this is almost identical to the ODE case and can be easily implemented in code. Although the implementation is simple, the error of the Euler method in particular, often exceeds its usefulness and other numerical methods are preferred. One example of another procedure is the geometric Depondt-Mertens method, which is briefly covered in [10, Ch. 7, p. 110] . The main improvement, compared to many other numerical methods, is that the magnitude of the magnetic moments is conserved throughout the calculation. Hence, a much greater stability is achieved together with a relaxation on the requirement on the step size h. Therefore, it is also the preferred numerical method for this report.

3 System structure

3.1 The magnetic Hamiltonian

To accurately model magnetic properties in a given material, we need to include interactions from several physical effects. A few examples are the exchange interaction, antisymmetric exchange interaction and anisotropy interactions. However, the former often suffices for studies of SGs [3] and is therefore the main attraction for our studies. For further elaboration of other contributions, see [10, Ch. 4].

Those who have studied statistical mechanics and solid state physics are probably familiar with the Heisenberg Hamiltonian

$$\mathcal{H}_{HE} = -\frac{1}{2}\sum_{i \neq j} J_{ij}\mathbf{m}_i \cdot \mathbf{m}_j. \tag{3.1}$$

It is a simple, yet effective model for describing magnetic interactions in a system [16, Ch. 8] and will be our model of choice in this report. To clarify notation, we sum over all sites i and their neighbors j, the exchange energy is governed by a constant J_{ij} which determines the strength and sign of the contribution. As the Hamiltonian is defined with a negative prefactor, a positive J_{ij} will lower the energy for parallel aligned spins and vice versa. Finally $\mathbf{m}_i$ and $\mathbf{m}_j$ are magnetic moments at two different sites. Although this expression will not be derived here, it is rather easy to justify by studying a two-electron system. The two electron spins can choose to be in a few different configurations described as; a singlet state, with anti-aligned spins and a triplet state, with aligned spins; each with different corresponding energies. Now, imagine our system is in its ground state, with lowest possible energy. As we cannot lower the energy more, the flip of a spin will result in an excitation of our system and a higher total energy. The energy difference between the ground state and this excited state is related to J_{ij}. This thought experiment can be expanded onto a system of many particles giving us the full Hamiltonian in Eq. (3.1). As the study of these exchange interactions are important to this report we give a few examples of how J_{ij} create different magnetic phases in a material.

Ferromagnetic system: In Figure 1 we observed a completely aligned system of spins which we called ferromagnetic. In this case, the exchange couplings are all positive $J_{ij} > 0$ which results in the totally aligned state having the lowest possible energy.

Anti-Ferromagnetic system: In contrast to the ferromagnetic state, in the antiferromagnetic state the spins want to align themselves anti-parallel. The exchange will, in this case, be negative $J_{ij} < 0$.

Paramagnetic system: Paramagnets are an example of a system with no magnetic ordering. The explanation for the behaviour is either weak, or no, interaction between spins, $J_{ij} \approx 0$, compared to the thermal fluctuations in the material, resulting in a totally random spin configuration. Most chemical compounds are found in their paramagnetic phase at room temperature.

Spin glass: Again, in Figure 1 we observed another completely disorganized spin system. This system differs from the paramagnetic phase by having an associated magnetic order parameter, together with dynamical properties such as ageing and memory. In the case of Figure 1 the behaviour is due to random exchange with the Edward-Anderson model.

It is also important to note that, even though the exchange parameters J_{ij} in this report will be chosen without considerations to any physical material, they can be calculated with a foundation in DFT. The expression is given by

$$J_{ij} = \frac{\mathrm{Im}}{4\pi} \int_{-\infty}^{E_F} \mathrm{Tr}\Big\{ [\delta_i(E) G_{ij}^{\alpha}(E) \delta_j(E) G_{ij}^{\beta}(E)] \Big\} dE,$$

where $\delta_i(E)$ is related to exchange splitting and G_{ij}^{σ} is a Greens function which connects sites i, j with the spin direction $\sigma = \alpha, \beta$. For further explanation and derivations see [10, Ch. 2].

3.2 Crystal structure

For the discussion of different exchange parameters, it is important to introduce some basic theory of crystal lattice structures. Let us start by introducing the concept of lattice vectors. For any three-dimensional, ordered lattice, one can define a set of three translational vectors, here denoted as $\mathbf{a}_i$ with $i = 1, 2, 3$. These vectors are defined such that if one translates along any of the three directions, the lattice structure will repeat itself. From these translational vectors, one can also define a general lattice vector

$$\mathbf{R} = n_1 \mathbf{a}_1 + n_1 \mathbf{a}_2 + n_2 \mathbf{a}_3,$$

with $n_i \in \mathbb{Z}_0$. This vector can then reach any point in the lattice and may be used together with the notion of a primitive unit cell, which is the smallest repeating pattern of lattice points, to construct the whole lattice structure. Therefore, $\mathbf{a}_i$ is known as basis vectors for the lattice.

3.2.1 Hexagonal plane

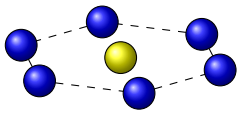


Figure 4: Illustration of a hexagonal lattice plane.

The simplest structure covered in this report is the two dimensional, hexagonal lattice plane (HEX). The basis vectors are given by

$$\mathbf{B}_{hex} = \frac{a}{2} \begin{pmatrix} 1 & -\frac{a}{2} & 0 \\ 0 & \frac{\sqrt{3}}{2} & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

with a being the lattice parameter of the material. The primitive unit cell is a single lattice point located at $\mathbf{p}_1 = (0, 0, 0)$.

3.2.2 Hexagonal compact packing

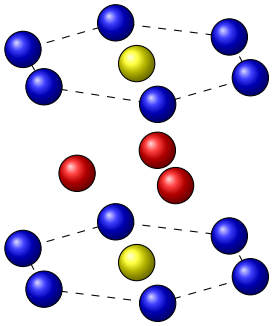


Figure 5: Illustration of hexagonal compact packing. The middle layer in red can intuitively be seen as a shifted hexagonal layer.

An example of a tightly packed lattice configuration is hexagonal compact packing (HCP). This structure consists of alternating hexagonal layers, see Figure 5, and the basis vectors for the lattice is given by

$$\mathbf{B}_{hcp} = \frac{a}{2}\begin{pmatrix} 1 & -\frac{a}{2} & 0 \\ 0 & \frac{\sqrt{3}}{2} & 0 \\ 0 & 0 & 2\sqrt{\frac{2}{3}} \end{pmatrix}.$$

The primitive unit cell of the structure is given by two particles with corresponding position vectors $\mathbf{p}_1 = (0,0,0)$, $\mathbf{p}_2 = (1/3, 2/3, 1/2)$, in the specified basis.

3.2.3 Double hexagonal compact packing

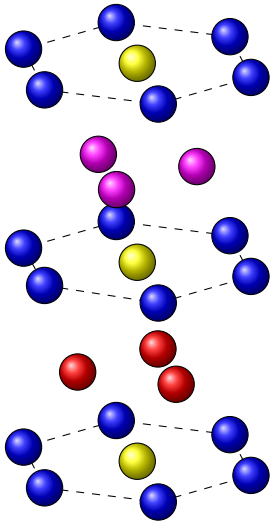


Figure 6: Illustration of double hexagonal compact packing.

The last system of interest is the DHCP structure. As the composition is similar to that of HCP the basis vectors are almost identical, only differing by a doubling of the last basis vector to accommodate the purple layer as seen in Figure 6. The basis vectors are given by

$$\mathbf{B}_{dhcp} = \frac{a}{2}\begin{pmatrix} 1 & -\frac{a}{2} & 0 \\ 0 & \frac{\sqrt{3}}{2} & 0 \\ 0 & 0 & 4\sqrt{\frac{2}{3}} \end{pmatrix}$$

This is the most complicated structure that will be covered in this report, needing four particles in the primitive unit cell to construct the full lattice. The position vectors of these particles are given by $\mathbf{p}_1 = (0,0,0)$, $\mathbf{p}_2 = (1/3, 2/3, 1/4)$, $\mathbf{p}_3 = (0,0,1/2)$ and $\mathbf{p}_4 = (-1/3, -2/3, 3/4)$.

4 Spin-glasses and correlation functions

Although a short overview of spin-glasses has already been covered in the introduction, some properties with regards to observation require further elaboration. As spin-glasses exhibit many interesting dynamical properties, such as ageing, slow relaxation and memory, it is convenient to study these properties using some sort of time-dependant correlation function. The correlation function of choice for this report is the autocorrelator given by

$$\boxed{C(t + t_w, t_w) = \langle \mathbf{m}_i(t_w + t) \cdot \mathbf{m}_i(t) \rangle} \tag{4.1}$$

where t_w is the waiting time, $\mathbf{m}_i$ is the magnetic moment at site i and the brackets indicate a thermal average [10, Ch. 7]. In short, Eq. (4.1) provides information about how a system of magnetic moments relaxes over time, due to their dynamics, and also show how close to equilibrium the system is. For a system in thermal equilibrium, the autocorrelation will be independent of the waiting time t_w [10, Ch. 7] and therefore

$$C(t + t_w, t_w) = C(t).$$

Furthermore, Eq. (4.1) have some specific behaviour with regards to spin-glasses. By letting the waiting time go to zero $t_w \approx 0$, the autocorrelation takes the form of a sum of exponentials [17] and can be described as

$$C(t, 0) \approx (1 - A)e^{-t/\tau_1} + Ae^{-t/\tau_2}. \tag{4.2}$$

Here τ_i describes the characteristic time scale for the evolution and the parameter A gives the crossing point between the two different time scales. Hence, spin-glasses exhibit a separation of dynamics dependant on these timescales. The different dynamics follow intuitively from the precessional and damping motion of Eq. (2.1) and the crossover depends on the damping parameter α [17]. Although Eq. (4.2) is not applicable for all waiting times, the property of multiple dynamical timescales is always true for SGs.

If we now would increase t_w we will eventually enter the ageing regime of the system, characterized by an initial drop in $C(t+t_w, t)$ followed by a plateau [17]. The location of this plateau is related to the spin-glass order parameter. This quantity differentiates the spin-glass phase from other disordered states such as paramagnetic phases [1, Ch. 2]. The order parameter will not be discussed further, it is however given by

$$q_{EA} = \lim_{t\to\infty} \lim_{t_w\to\infty} \lim_{L\to\infty} C(t + t_w, t_w),$$

with L being the macroscopic size of the system. A historical note, the EA subscript refers to the same Edwards and Andersson whom proposed the random spin-interaction model mentioned earlier in the introduction and section 3.1 [1, Ch. 2].

Finally, the autocorrelation function found in Eq. (4.1) has been extensively used in the study of spin-glasses [2, 17, 18]. It is therefore the most important quantity to study for this report.

5 Methodology

5.1 Simulation configuration

The spin-dynamics software UppASD was used for all simulations, which employ a few hard coded input files to construct and solve for the dynamics of the system, as with probing the correlation function in Eq. (4.1) [11]. Generating these files in an efficient manor then becomes the main goal to effectively make simulations using a varying range of parameters. Hence, a script was written in Python to create all necessary files for a simulation and sweep J_{ij} over a given range with a specified step size. The full code can be found in the Appendix, see section B, and below is a pseudo-code representation.

```
Given system structre, temperature, initial moments:
    generate: inpsd.dat, posfile, momfile

for i in (range of J_ij):
    generate jfile
    run simulation
    save output
```

In all cases only one coupling was varied at a time. The reasoning behind this was purely computational as the number of possible configurations for k number of couplings and R number of parameter values goes as $N = R^k$. As an example, say we want to vary three different couplings in the range $[-2, 2]$ with a step size of $h = 0.1$. Then the total number of simulations N would be

$$N = \left(\frac{|-2-2|}{0.1} \right)^3 = 40^3 = 64000,$$

which is unreasonable for the scope of this project.

The specific systems of study are the crystal structures HEX, HCP and DHCP as introduced previously. All systems start from a ferromagnetic state, with aligned magnetic moments in the x-direction, a temperature of 1 K and with a damping factor $\alpha = 0.5$. In all cases, unless otherwise stated, the time step for the simulation was set to $dt = 10^{-16}$(s) and $5{\cdot}10^5$ time steps were made

corresponding to a total simulation time of 50 (ps). Then, the python code is run to set up the system, give it a name, and create the necessary input files using the specific basis vectors in Eqs. (3.2.1), (3.2.2) and (3.2.3). Finally, the sweep is started by specifying an interval for the exchange, together with a step-size. The specific cases are given below.

5.2 System configurations

5.2.1 Hexagonal plane

Two systems of size 60×60 and 120×120 unit cells are investigated. With all the input files generated two couplings are set up in the lattice plane, one to the nearest neighbor J_1 and one to the next nearest J_2. The spin interactions can be seen in Figure 7. J_1 is considered fixed to $J_1 = -1$ (mRy) and J_2 is varied through the range $J_2 \in [-1, 1]$ with a step-size of $h = 0.1$. The correlation function $C(t + t_w, t_w)$ is sampled throughout the run.

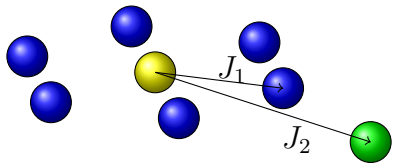


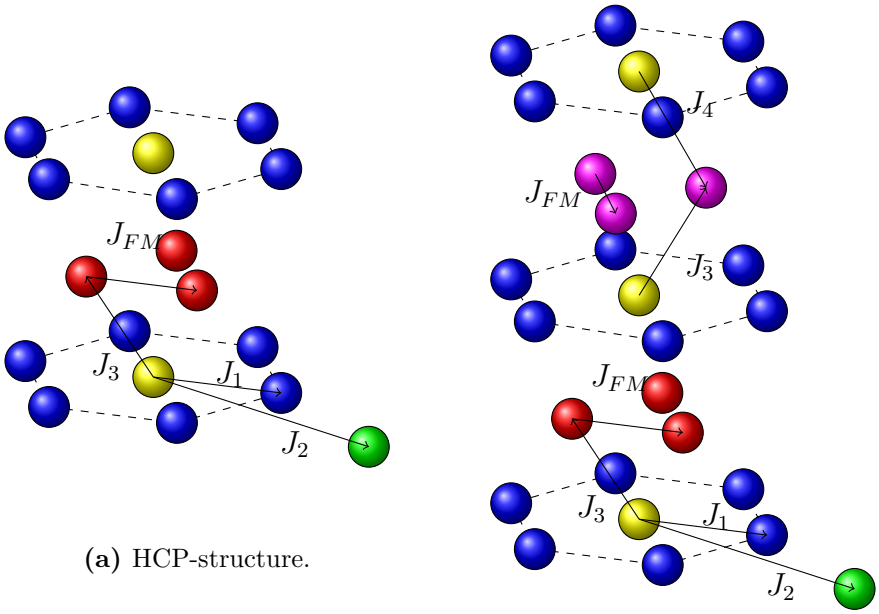
Figure 7: Neighbor interactions of the HEX structure. The blue particles are the nearest neighbors and green represents the next nearest. All couplings follow hexagonal symmetry giving a total of 12 couplings within the cell.

5.2.2 Hexagonal compact packing

A system of $20 \times 20 \times 20$ unit cell was simulated using the HCP lattice configuration. Two new interactions are introduced from the HEX structure, as seen in Figure 8a. The offset layer in red was chosen to be in a ferromagnetic configuration with $J_{FM} = 1$ (mRy) and the interplanar spin interaction J_3 was varied in the range of $J_3 \in [-1, 1]$ with a step size of $h = 0.1$. The blue layer in figure 8a follow the same exchange scheme as in the HEX-system with the J_2 parameter fixed to a constant value depending on the result from the earlier hexagonal simulation. From these sweeps, four different exchange combinations J_1, J_2, J_3 was chosen as candidates for SGs. These were then run for a longer simulation time, using a total of $5 \cdot 10^6$ time steps with the step-size $dt = 10^{-15}$ (s) corresponding to a total simulation time of 5 (ns).

5.2.3 Double hexagonal compact packing

A DHCP system of $20 \times 20 \times 20$ unit cells is investigated, following largely the same configuration as the HCP one. The exchanges J_1, J_2, J_3 and J_{FE}, as seen in figure 8b, have values defined from the longer HCP sweep and the new interaction J_4 is varied through the range $J_4 \in [-1, 1]$ with a step size of $h = 0.1$. Two sets of parameters are selected from the sweep and run again with the longer simulation time 5 (ns), similar to the HCP case.



(a) HCP-structure.

(b) DHCP-structure.

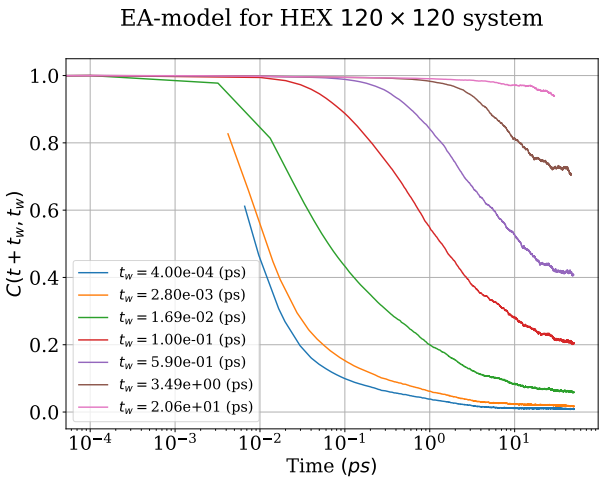
Figure 8: Interaction in HCP and DHCP structures following hexagonal symmetry.

6 Results

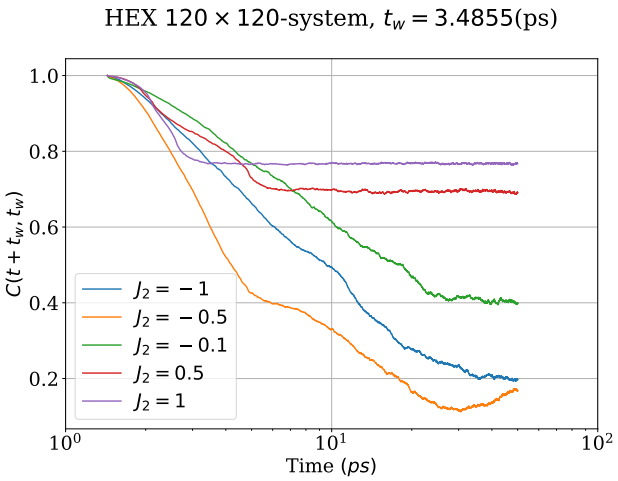
Each parameter sweep results in 21 different parameter combinations, these were studied by plotting the correlation function against logarithmic time for each waiting time. An example of a resulting plot can be seen in Figure 10a. To examine if the systems exhibit some sort of self-induced, glassy behavior each plot was then compared using ocular inspection to the corresponding EA-model simulation, see for example Figure 12a. The coupling-sets that showed similarity to the random exchange model were then chosen for further analysis, either by adding more interactions, as for the hexagonal plane, or by increasing the simulation time. The process was then repeated after this final analysis and these are the results presented in this section. Due to the nature of simulation studies a huge amount of data is generated and all cannot be presented in this report, we will however include all files necessary to run the simulation in the Appendix for anyone interested. To show some of the comparative process, each section also includes a Figure presenting $C(t + t_w, t_w)$ for a fixed waiting time and instead varying exchange, as seen in Figure 9b for example. The choice of waiting time is arbitrary and chosen to reflect the different dynamics for varying exchange. Furthermore, all spin interactions are given in (mRy) if not otherwise stated and all timescales are logarithmic.

6.1 HEX - systems

Two different hexagonal systems were investigated with different sizes. Here we present the 120×120 case. In Figure 9a the random-exchange system can be seen with correlation function $C(t + t_w, t_w)$ for different waiting times. Beside it, in Figure 9b, we present the correlation function for a specific waiting time, t_w, for different J_2 within the plane. Note the early stabilization of the function for more positive values of the parameter. Lastly, in Figure 10 one finds autocorrelation plots for two specific exchange interactions, $J_2 = -0.1$ in 10a and $J_2 = -1$ in Figure 10b.

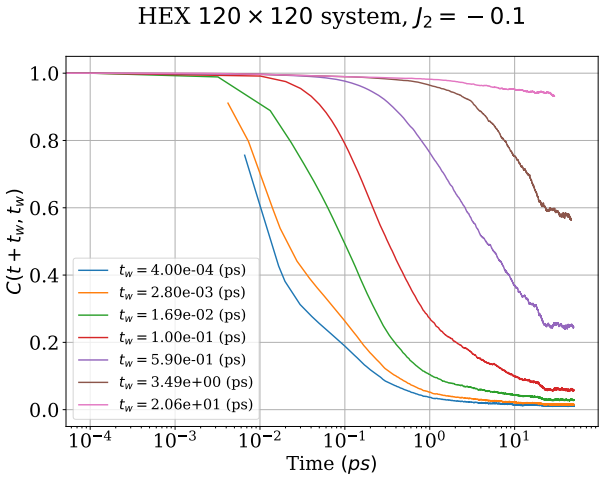


(a) Hexagonal plane, EA-model autocorrelation for different waiting times.

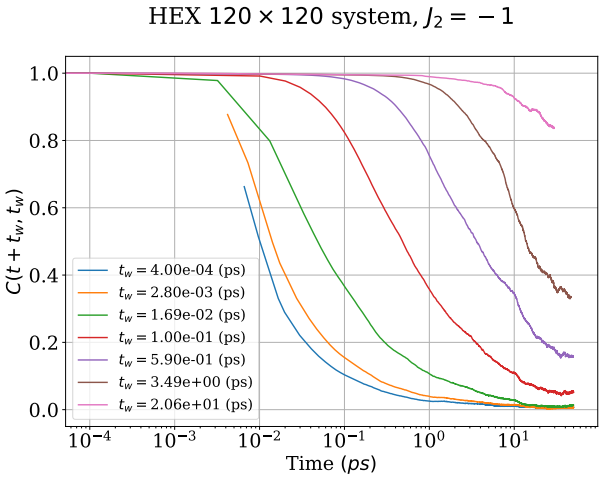


(b) Correlation function for varying in-plane interaction and $J_1 = -1$.

Figure 9: To the left, autocorrelation for EA-model for hexagonal structure. To the right, autocorrelation for specific waiting time t_w for a collection of J_2 parameters.



(a) Same as Figure 9a but now with specific exchange parameters, $J_1 = -1$ and $J_2 = -0.1$.



(b) Same as Figure 9a but now with specific exchange parameters, $J_1 = -1$ and $J_2 = -1$.

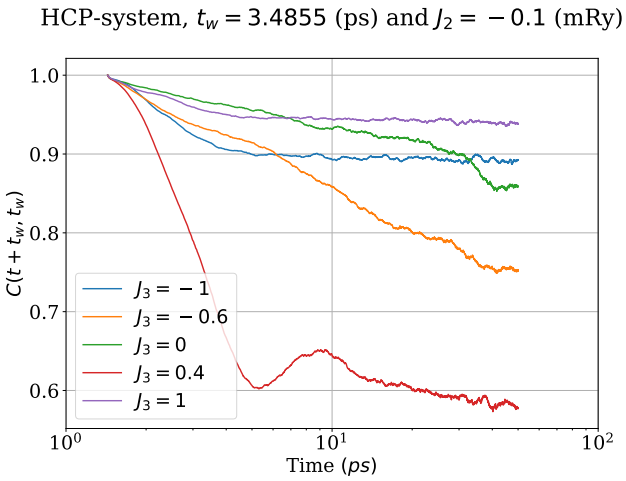
Figure 10: Autocorrelation $C(t + t_w, t_w)$ plotted against logarithmic time for specific exchange parameters.

6.2 HCP - systems

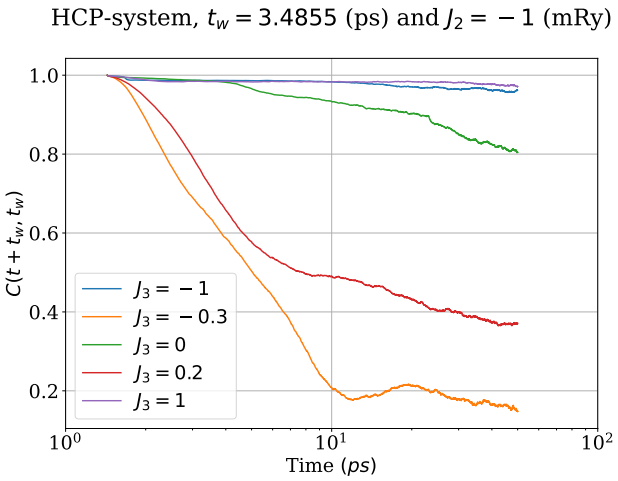
Based on results from the larger hexagonal system, two different parameters were chosen to continue investigations for larger structures, namely $J_2 = -0.1$ and $J_2 = -1$. The former case is presented in Figure 11a, for different interplanar exchange, and the latter case is shown in Figure 11b. As before, a random exchange run was made using the Edwards-Andersson model and the resulting behavior can be seen in Figure 12a. From the results of the sweeps in Figure 11, four pairs of exchange (J_2, J_3) were chosen as subjects for further investigation, see Table 1. These were run again but now with a much longer simulation time. The results for the most prominent case can be seen in Figure 12b.

J_2 (mRy)	-0.1	-0.1	-1	-1
J_3 (mRy)	-0.6	0.4	-0.3	0.2

Table 1: Table of the chosen coupling parameters. The pairs are presented column wise.

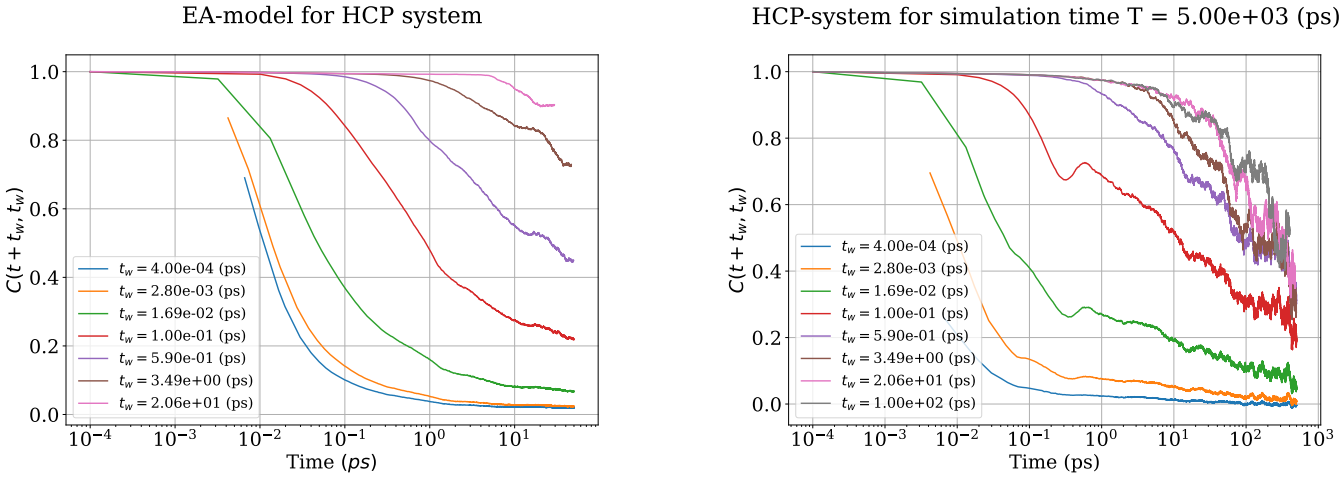


(a) Correlation function for the HCP-structure for different interplanar couplings and $J_2 = -0.1$, J_3 given in units of mRy. Notable cases are when $J_3 = -0.6$ and $J_3 = 0.4$ as the decay throughout the whole simulation.



(b) Same as for Figure 11a but $J_2 = -1$. Pay closer attention to the interplanar couplings $J_3 = -0.3$ and $J_3 = 0.2$ as they behave similarly to the random coupling case seen in Figure 12a.

Figure 11: Same as for Figure 10 but for hexagonal compact packing.



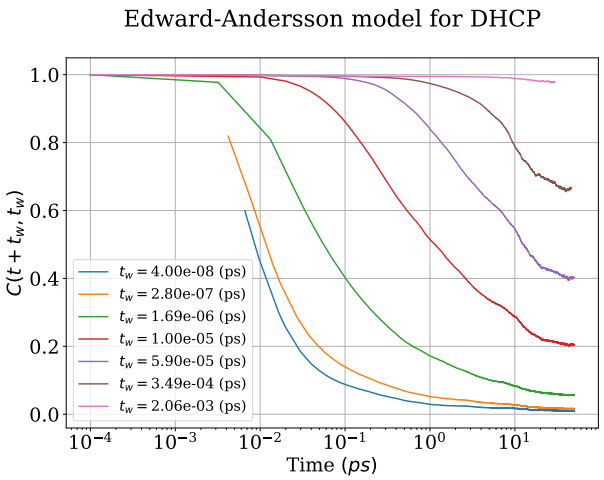
(a) HCP-system subject to the Edwards-Andersson model. The correlation function $C(t + t_w, t_w)$ is presented for different waiting times t_w. Note the exponential decrease for $t_w \approx 0$, as predicted in Eq. (4.2).

(b) HCP-system with couplings $J_2 = -1$ and $J_3 = 0.2$ and a simulation time of 5000 (ps).

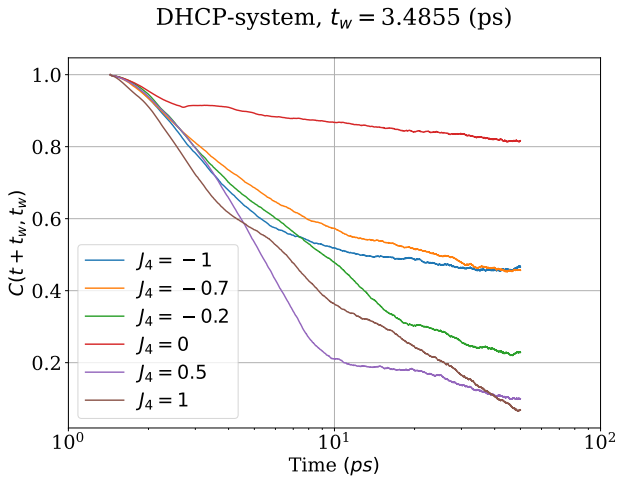
Figure 12: Autocorrelator $C(t + t_w, t_w)$ as a function of logarithmic time for EA-model (left) and a definitive set of coupling for a longer simulation time (right).

6.3 DHCP - systems

A single sweep on the double hexagonal structure was made and the result for a fixed waiting time can be seen in Figure 13b. As previously, a simulation was run with the EA-model and the resulting correlation behaviour can be seen in Figure 13a. Based on comparisons of the Edwards-Andersson system with the simulations from the sweep a couple sets of parameters was chosen for a final, longer simulation. These were $J_1 = J_2 = -1$, $J_3 = 0.2$ and $J_4 = -0.2$ or $J_4 = -0.9$. These runs are presented in Figure 14a and 14b respectively.

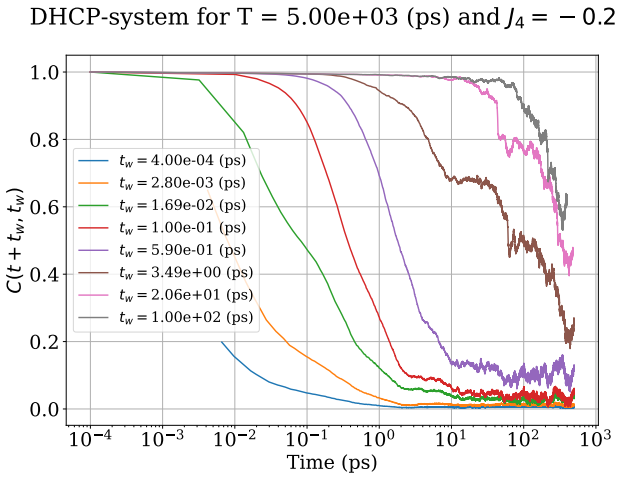


(a) EA-model simulation for the DHCP-structure. The correlation function is sampled for different waiting times t_w.

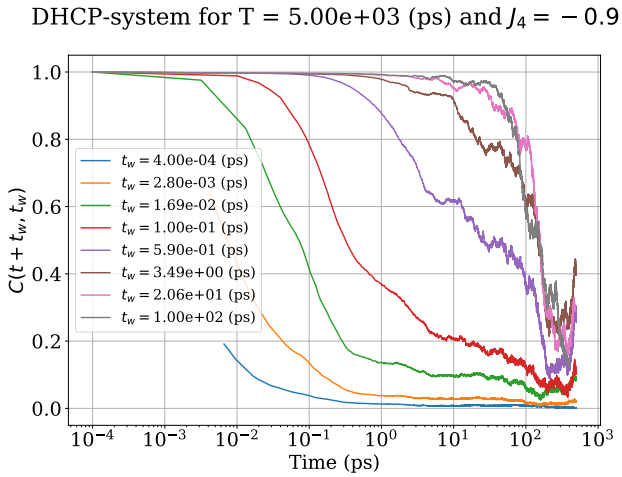


(b) Correlation function $C(t + t_w, t_w)$ for different parameters in the double hexagonal compact packing structure. The values $J_4 = -0.2$, $J_4 = 0.2$ were chosen as a candidates for longer simulation times.

Figure 13: EA-model (left) and parameter sweep (right) for the DHCP-structure.



(a) DHCP-system for $J_2 = -1$, $J_3 = 0.2$ and $J_4 = -0.2$ for a simulation time of 5 (ns).



(b) Same as Figure 14a but with $J_4 = -0.9$

Figure 14: Longer simulation runs for two cases of J_4 in the DHCP-structure.

7 Discussion

7.1 Investigated systems

We start from the bottom up by addressing the hexagonal plane. It should be stated that we do not expect glass-like behavior from these classes of systems, constituting of only nearest and next-nearest neighbour exchange in the plane. The reasoning behind this statement can be explained by the two cases when $J_2 > 0$ and $J_2 < 0$. The former will result in some sort of anti-ferromagnetic ordering, and the latter will cause frustration but not result in a SG. However, we still observe the long relaxation times associated with SGs as seen in Figure 9a. An explanation for this behaviour could be other frustrated systems such as spin-liquids [19]. This special class of magnetic systems share the degenerate properties of the ground states in SGs, resulting in similar behaviour of the correlation function due to the continuous reconfiguration of the spins.

To further emphasize this statement, the results presented in this report match the findings of [19] by showing a heavily frustrated system in the range $0.08 \lesssim J_2/J_1 \lesssim 0.16$. As discussed in section 6.2, one of the studied systems have $J_2/J_1 = 0.1$ and the measured correlation function can be seen in Figure 10a. This results is particularly interesting due to the striking resemblance between the EA-model and the finite exchange in this case, which raises the question if there is some correlation between spin-liquids and the EA-model. It is however important to state that the two findings are not directly comparable as they do not share the same lattice structure. This report only considers hexagonal structures, whereas the article [19] studies other triangular lattices. Furthermore, we also need to state that the spin-liquid and spin-glass are fundamentally different magnetic phases, as discussed in [20].

Moving on to the higher dimensional systems of hexagonal compact packing and double hexagonal compact packing. In the case of the EA-model, both structures experience the long relaxation time associated with SGs, as seen by the steady decrease of $C(t + t_w, t_w)$ for all t_w in Figures 12a and 13a. They furthermore show a resemblance of a plateau, which is associated with SGs, making them well-suited to exhibit self-induced SG behavior. The DHCP-system, in particular, has strong resemblance with investigated SG-systems from other sources, see Figure4 in [17] and Figure2 in [18]. In the finite exchange simulations seen in figures 12b, 14a and 14a both configurations show behaviour somewhat resembling the EA-model. They exhibit slow relaxation times but are missing a clearly marked plateau, instead showing some sort of oscillatory behaviour for $t > 10^2$ (ps). This could be due to a lack of time resolution, as the plot is logarithmic, or due to some other interesting dynamics. For example, the system may oscillate between some quasi-equilibrium states,

similar to that of a spin-liquid, and therefore exhibiting the jagged behaviour of the autocorrelator. It is also important to note that these systems have a much greater simulation time, making a direct comparison more difficult, they should however keep their glassy behaviour for longer simulation times. Lastly, we want to stress that it is hard to definitively state if these system are spin-glasses or not, indicating the difficulty of classifying these kinds of systems, and we will discuss improvements to the analysis in the coming paragraph.

7.2 Improvements to analysis

Even though the correlation function $C(t + t_w, t_w)$ is an interesting quantity when studying SGs, it is prominent from this report that it is not sufficient to classify these magnetic systems completely. The comparison to the Edwards-Andersson system is only enough to get a rough idea of the spin behaviour but it is lacking quantitative properties. It is thus suggested to complement this method with further analyses. These could be chosen from already existing experimental methods, such as performing a temperature sweep to study the phase transition. However, such methods require extensive computation considering the number of simulations needed to study the behaviour. We should therefore consider other, numerical approaches. An example is the average exchange within the system $\langle J_{ij} \rangle$ as mentioned in [3]. For a system to choose a dominant spin direction, the average has to reflect this with a value largely different from zero. In contrast, a value close to zero will result in a system exhibiting some sort of frustration. Using this quantity together with the correlation function could result in a more thorough analysis of the SG candidates, additional investigations are however needed.

Furthermore, the average $\langle J_{ij} \rangle$ could also be used in the parameter variation process to exclude parameters which would not exhibit this frustration. Using this method, it enables the investigation to cover a wider variety of interaction sets by considering all possible configurations and only then running the few that lie within a given range. Hence, resulting in a more systematic approach to choosing the parameter values. We however also need to mention a caveat of the average J_{ij} method, anti-ferromagnetic systems. These configurations will naturally satisfy the conditions on $\langle J_{ij} \rangle$ and therefore be included in the sweep. These cases would require some extra care when generating the exchange parameters.

Another method to quantify the spin-glass behaviour would be that of a simple fit to a given function, such as Eq. (4.2), and then use the different fitted parameters for comparison. It is however evident from the drastically

different behaviour of the autocorrelator for different systems that a reliable fit would difficult to achieve. We therefore suggest to study other approaches for classifying the behaviour.

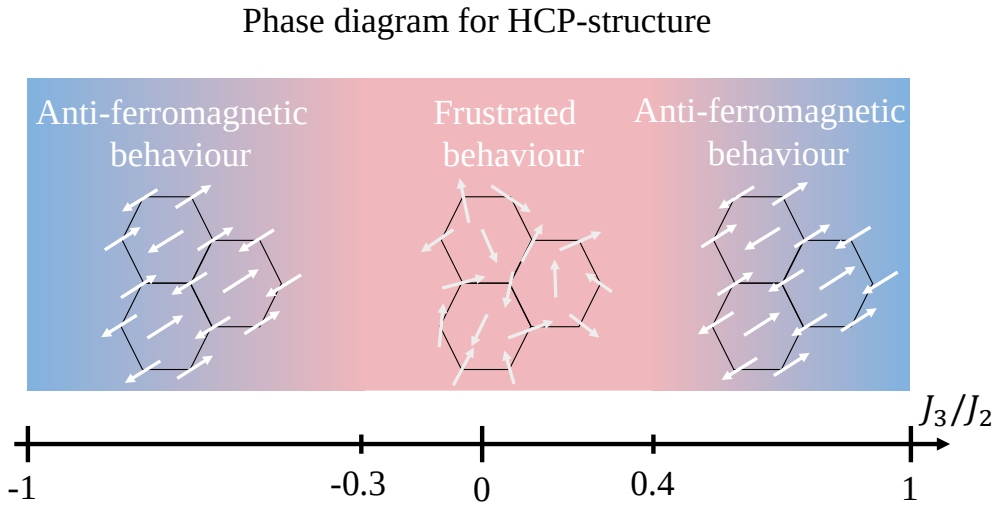


Figure 15: Phase diagram for one case of the HCP-system in terms of the ratio J_3/J_2 with varying J_3 and $J_2 = -1$.

Finally, to summarize the behaviour of the spin-dynamics one can construct a phase diagram. This is done by classifying the magnetic ordering from the correlation function for different values of the parameters. In the case of the $J_2 = -1$ HCP system, the behaviour is seen in Figure15. Similar diagrams can be constructed for the other cases; this is however quite troublesome, due to the difficulty in classifying magnetic ordering as mentioned previously. Hence, the reason why the method was not employed in this project. The use of phase diagrams could however be useful in the choice of simulation parameters for composite systems. They enable a deeper, visual, understanding how the different magnetic phases are related and provide some much needed intuition. Phase diagram should therefore be considered as an improvement to the analysis for further studies.

8 Conclusion

This report has presented the investigation of self-induced spin-glasses in Heisenberg systems using a variational approach. The results show the associated slow relaxation time of SGs in stacked hexagonal lattices. It is however difficult to draw any conclusions regarding other defining properties such as a plateau in the autocorrelation function. As a result, it can not be stated that a self-induced spin-glass was found using the methods presented in this report. It is therefore suggested that the analysis should be complemented with other, more quantitative studies to more effectively classify these magnetic systems. A possible contribution to the analysis would be the study of an exchange average, which could be used to more systematically choose parameter sets for the simulations. Nevertheless, it is clear that the method of varying exchange parameters in the Heisenberg Hamiltonian is promising and is able to produce spin-systems exhibiting spin-frustration using only nearest and next-nearest neighbor interactions.

References

1. Fischer, K. H. *Spin glasses* ISBN: 0521342961 (Cambridge Univ. Press, Cambridge, 1991).
2. Kamber, U. *et al.* Self-induced spin glass state in elemental and crystalline neodymium. *Science* **368,** 966+ (May 2020).
3. Nordblad, P. Disordered magnetic systems (2015).
4. Kirkpatrick, S. Frustration and ground-state degeneracy in spin glasses. *Physical Review. B, Solid state* **16,** 4630–4641 (1977).
5. Baity Jesi, M. *Spin Glasses: Criticality and Energy Landscapes* ISBN: 9783319412306 (Springer International Publishing, Cham).
6. Binder, K. & Young, A. Spin-Glasses - Experimental Facts, Theoretical Concepts, and Open Questions. *Reviews of Modern Physics* **58,** 801–976 (Oct. 1986).
7. Edwards, S. F. & Anderson, P. W. Theory of spin glasses. *Journal of Physics F: Metal Physics* **5,** 965 (1975).
8. Nishimori, H. Spin glasses and information. *Physica A* **384,** 94–99 (2007).
9. Baity-Jesi, M. *et al.* Comparing dynamics: deep neural networks versus glassy systems*. *Journal of Statistical Mechanics: Theory and Experiment* **2019,** 124013 (Dec. 2019).
10. Eriksson, O., Bergman, A., Bergqvist, L. & Hellsvik, J. *Atomistic spin dynamics : foundations and applications* ISBN: 9780198788669 (Oxford University Press, Oxford, 2017).
11. Division of Materials Theory at Uppsala University. *Uppsala Atomistic Spin Dynamics* 2023. `https://github.com/UppASD`.
12. Kohn, W. Nobel Lecture: Electronic structure of matter—wave functions and density functionals. *Reviews of Modern Physics* **71,** 1253–1266 (1999).
13. Gilbert, T. A phenomenological theory of damping in ferromagnetic materials. *IEEE Transactions on Magnetics* **40,** 3443–3449 (2004).
14. Boyce, W. E. *Elementary Differential Equations and Boundary Value Problems, 11th Edition* ISBN: 9781119382874 (John Wiley & Sons, 2017).
15. Kloeden, P. E. *Numerical solution of stochastic differential equations* ISBN: 3540540628 (Springer-Vlg, Berlin ; 1992).
16. Ibach, H. & Lüth, H. *Solid-State Physics: An Introduction to Principles of Materials Science* ISBN: 9783540938033 (Springer Berlin Heidelberg, Berlin, Heidelberg).

17. Skubic, B. *et al.* Atomistic spin dynamics of the Cu-Mn spin-glass alloy. *Physical Review. B, Condensed matter* **79,** 024411– (2009).

18. Berthier, L. & Young, A. P. Aging dynamics of the Heisenberg spin glass. *Physical Review. B, Condensed matter and materials physics* **69,** 184423.1–184423.15 (2004).

19. Iqbal, Y., Hu, W.-J., Thomale, R., Poilblanc, D. & Becca, F. Spin liquid nature in the Heisenberg J1-J2 triangular antiferromagnet. *Physical Review B* **93** (2016).

20. Wen, J., Yu, S.-L., Li, S., Yu, W. & Li, J.-X. Experimental identification of quantum spin liquids. *NPJ Quantum Materials* **4** (Apr. 2019).

Appendix

A UppASD

All the behind the scenes machinery of the UppASD software have already been presented in the theoretical background and this will be a purely practical introduction to how to handle the program. As mentioned earlier, UppASD uses a few input files to set up the simulation. These come in variety of forms and the main ones of interest for this project are the `inpsd.dat`, `posfile`, `momfile` and the `jfile`. All except the latter file serve to set up the system of interest, they contain information about the lattice structure, the temperature of the system and initial configurations and will most often be constant throughout a parameter sweep. What is actually then varied are the `jfile` which change for each simulation. This file contains the coupling of the spins, the J_{ij} as seen in Eq. (3.1). Furthermore, to choose which quantities one would like to study a few keywords has to be added to the `inpsd.dat`-file. For the purposes of this report these were, `do_autocorr T` and `acfile .\twfile` were the `twfile` contains the different waiting times for the correlation function $C(t + t_w, t_w)$ in time steps. For a more detailed documentation of all the different inputs see [ref].

B Code

Below is the major script for running simulations automatically in UppASD with varying exchange.

```
import numpy as np
from string import Template
import os
import itertools

# ---------------------------------------------------------------------------------------------
# This script is a tool used to run multiple simulations of UppASD with varying Heisenberg exchange, J.
# The script generates a inputfile, posfile and momfile given a set of initial conditions. It then
# then takes a range of exchange, how many parameters you want to vary and then runs the simulation
# for those.
# ---------------------------------------------------------------------------------------------

def gen_input(simid = '', system = '10   10   10', temp = 1, struct = '', initmag = 1, posfiletype = 'C
    '):

    # ---------------------------------------------------------------------------------------------
    # Genereatas inputfile for UppASD based on size, structure and temperrature. Supported
    # structures are bcc, fcc, hex, hcp and dhcp.
    # For further documentation on initmag and posfiiletype see UppASD documentation.
    # ---------------------------------------------------------------------------------------------

    if struct == 'bcc':
        sym = 1
        cell =  f'1.000   0.000   0.000 \n'\
                f'0.000   1.000   0.000 \n'\
                f'0.000   0.000   1.000'

    elif struct == 'fcc':
        sym = 1
        cell =  f'0.500   0.500   0.000 \n'\
                f'0.500   0.000   0.500 \n'\
                f'0.000   0.500   0.500'

    elif struct == 'hcp':
        sym = 3
```

```
        posfiletype = 'D'

        cell =  f'1.000   0.000   0.000 \n'\
                f'-0.500  {np.sqrt(3)/2}  0.000 \n'\
                f'0.000   0.000   {2*np.sqrt(2/3)}'

    elif struct == 'dhcp':
        sym = 3
        posfiletype = 'D'
        cell =  f'1.000   0.000   0.000 \n'\
                f'-0.500   {np.sqrt(3)/2}  0.000 \n'\
                f' 0.000   0.000   {4*np.sqrt(2/3)}'

    elif struct == 'hex':
        sym = 3
        cell =  f'1.000   0.000   0.000 \n'\
                f'-0.500   {np.sqrt(3)/2}  0.000 \n'\
                f' 0.000   0.000   {2*np.sqrt(2/3)}'

    else:
        raise 'Structure not defined!'

    template = Template(''.join(open('input_temp.txt').readlines()))
    inputfile  = template.substitute({'simid': simid, 'system': system,
                                      'cell': cell, 'sym': sym, 'temp': temp, 'initmag': initmag,
                                      'posfiletype' : posfiletype})

    gen_file(inputfile, 'inpsd.dat')

def gen_posfile(struct):

    # ------------------------------------------------------------------------------------------
    # Generate posfile for given structure. Here all particles within the unit cell
    # are considered different.
    # ------------------------------------------------------------------------------------------

    if struct == 'bcc':
        pos =   f'1 1     0.000   0.000   0.000 \n2 1     0.500   0.500   0.500'
    elif struct == 'fcc':
        pos =   f'1 1     0.000   0.000   0.000'
    elif struct == 'hcp':
        pos =   f'1 1     0.000   0.000   0.000 \n2 2     {1/3}   {2/3}   0.500'
    elif struct == 'dhcp':
        pos =   f'1 1     0.000   0.000   0.000 \n'\
                f'2 2     {1/3}   {2/3}   0.250 \n'\
                f'3 3     0.000   0.000   0.500 \n'\
                f'4 4   {-1/3}   {-2/3}   0.750'
    elif struct == 'hex':
        pos =   f'1 1     0.000   0.000   0.000'

    gen_file(pos, 'posfile')

def gen_jfile(struct, J):

    # ------------------------------------------------------------------------------------------
    # Generate jfile for ONE parameter, if more parameters are to be varied they have to be
    # specified within the code. J is read as a list, more couplings can therefor be added by just
    # substituting an exchange for {J[i]} with i being the index of the exchange in the list.
    # ------------------------------------------------------------------------------------------

    if struct == 'bcc':
        exc =   f'1 1  0.5    0.5     0.5     {J[0]} \n'\
                f'1 1  1.0      0.0     0.0     {J[1]} \n'\
                f'1 1  1.0      1.0     0.0     {J[2]}'
    elif struct == 'fcc':
        exc =   f'1 1  0.5    0.5     0.0     {J[0]} \n'\
                f'1 1  0.5      0.0     0.5     {J[1]} \n'\
                f'1 1  0.0      0.5     0.5     {J[2]}'
    elif struct == 'hcp':
        exc =   f'1 1 1.0000    0.0000    0.0000    1 \n'\
                f'2 2 1.0000    0.0000    0.0000    -1 \n'\
                f'2 2 2.0000    1.0000    0.0000    1   \n'\
                f'1 2 0.0000    0.0000    0.0000    {J[0]}\n'\
                f'2 1 0.0000    0.0000    0.0000    {J[0]}\n'

    elif struct == 'dhcp':
        exc =   f'1 1    1  0  0     1.0   \n'\
                f'2 2    1  0  0     -1.0  \n'\
                f'2 2    2  1  0     -1.0  \n'\
                f'3 3    1  0  0     1.0   \n'\
                f'4 4    1  0  0     -1.0  \n'\
                f'4 4    2  1  0     -1.0  \n'\
                f'1 2    0  0  0     0.2  \n'\
                f'2 1    0  0  0     0.2  \n'\
```

```
                f'2 3    0  0  0     0.2  \n'\
                f'3 2    0  0  0     0.2  \n'\
                f'3 4    0  0  0    {J[0]} \n'\
                f'4 3    0  0  0    {J[0]} \n'\
                f'4 1    0  0  1    {J[0]} \n'\
                f'1 4    0  0 -1    {J[0]} \n'

    elif struct == 'hex':
        exc =   f'1 1  1.000   0.000   0.000   -1.00 \n'\
                f'1 1  2.000   1.000   0.000  {J[0]} \n'


    gen_file(exc, 'jfile')


def gen_momfile(struct, mom):

    # ------------------------------------------------------------------------------------------
    # Generate momfile given structure and an initial moment "mom". If "mom" is an empty string
    # all initial moment strengths are considered random. They all however start in an aligned
    # configuration.
    # ------------------------------------------------------------------------------------------

    if mom == '':
        mom = np.random.rand(4)

    if struct == 'bcc':
        momline = f'1 1 {mom[0]}    1.000   0.000 0.000 \n2 1 {mom[1]}    1.0   0.0   0.0'
    elif struct == 'fcc':
        momline = f'1 1 {mom[0]}    1.000   0.000   0.000'
    elif struct == 'hcp':
        momline = f'1 1 {mom[0]}    1.000   0.000   0.000 \n2 1 {mom[0]}    1.0     0.00    0.00'
    elif struct == 'dhcp':
        momline =   f'1 1 {mom[0]}   1.0   0.0   0.0 \n'\
                    f'2 1 {mom[0]}   1.0   0.0   0.0 \n'\
                    f'3 1 {mom[0]}   1.0   0.0   0.0 \n'\
                    f'4 1 {mom[0]}   1.0   0.0   0.0'
    elif struct == 'hex':
        momline = f'1 1 {mom[0]}    0.000   0.000   0.000'

    gen_file(momline, 'momfile')

def MeshLoop(Jlen, jrange, simid, struct, h):

    # ------------------------------------------------------------------------------------------
    # Major loop for running the simulation. The code generates a list of tuples containing
    # all possible variation of the parameter given the desired range. If only one parameter
    # is varied this is equivalent to a np.arange() array. The code saves all exchanges for
    # the single simulations in a txt-file and saves all outputs from UppASD (*out) in
    # individual folders.
    # ------------------------------------------------------------------------------------------

    i = 1

    #  Create iteration mesh
    iter = [list(np.arange(jrange[0], jrange[1] +h, h)) for i in range(Jlen)]
    J = [t for t in itertools.product(*iter)]

    for j in J: # Loop over all possible combinations

        gen_jfile(struct, j)

        os.system('~/UppASD/bin/sd.gfortran')   # Run simulation

        # Save Js
        jlines = open('jfile', 'r')
        lines = jlines.readlines()
        jlines.close()

        J_res = open(f'j_res_{simid}.txt', 'a')
        J_res.write(f'Run {i}:\n'+str(''.join(lines).replace(',',''))+'\n')
        J_res.close()

        # Save output files
        os.system(f'mkdir {simid+str(i)}')
        os.system(f'mv *out {simid+str(i)}')

        i += 1 # New step

def MonteCarloLoop(N, jrange, simid, struct, Jlen):

    # ------------------------------------------------------------------------------------------
    # Same as the above but with randomly generated couplings withing the specified range. Takes
    # an integer as the max number of simulations. Not thoroughly tested.
```

```
    # ------------------------------------------------------------------------------------------

    for i in range(N):

        J = abs(jrange[0] - jrange[-1])*np.random.rand(Jlen) - abs(jrange[-1])

        gen_jfile(struct, J)

        os.system('~/UppASD/bin/sd.gfortran')   # Run simulation

        # Save Js
        jlines = open('jfile', 'r')
        lines = jlines.readlines()
        jlines.close()

        J_res = open(f'j_res_{simid}.txt', 'a')
        J_res.write(f'Run {i}:\n'+str('\n'.join(lines).replace(',',''))+'\n')
        J_res.close()

        # Save results in folder
        os.system(f'rm moments.{simid}.out')

        os.system(f'mkdir {simid+str(i)}')
        os.system(f'mv *out {simid+str(i)}')

def single_sim(struct, j2): # Code for just running one simulation and saving the output.

        gen_jfile(struct, j2)

        os.system('~/UppASD/bin/sd.gfortran')   # Run simulation

        # Save output files
        os.system(f'mkdir {simid}')
        os.system(f'mv *out {simid}')

def gen_file(string, filename):
    file = open(filename, 'w')
    file.write(string)
    file.close()
    print(f'{filename} generated!')

# ------------------------------------------------------------------------------------------
#   Below is a quick way of running a simulation sweep by just running this script. The code asks
#   for all nessessary inputs and runs the simulation using the functions above. It is however
#   recomended that another python script is used to set up the simulation.
# ------------------------------------------------------------------------------------------

if __name__ == "__main__":
    # Check for inputs
    simid = input('Specify simid: ').replace(' ','')
    system = input('Specify system size: ').replace(',', '  ')
    temp = input('Specify temp: ')
    struct = input('Specify lattice strcuture: ')
    mom = [float(x) for x in input('Specify intital moment, leave empty for random: ').split(',')]
    jrange = [float(x) for x in input('Specify range of couplings: ').split(',')]

    if mom != '': # Check for random inital moments
        initmag = 3
    else:
        initmag = 1

    # Generate sim-files
    gen_input(simid, system, temp, struct, initmag)
    gen_posfile(struct)
    gen_momfile(mom)

    Looptype = input('Do you want meshloop (M) or montecarlo (R) [M/R]? ').lower().replace(' ','')

    if Looptype == 'm': # MeshLoop
        h = float(input('Specify stepsize: '))
        Jlen = int(input('Specify number of Js: '))

        check = input('Run simulation [Y/N] ? ').lower()
        if check == 'y':
            MeshLoop(Jlen, jrange, simid, struct, h) # Run loop
        else:
            exit()

    if Looptype == 'r': # MonteCarlo loop
        N = int(input('Specify number of iterations: '))
        Jlen = int(input('Specify number of Js: '))

        check = input('Run simulation [Y/N] ? ').lower()
        if check == 'y':
```

```
287             MonteCarloLoop(N, jrange, simid, struct, Jlen) # Run loop
288         else:
289             exit()
290 
291     # Clean folder
292     clean = input('Simulation done! Clean folder [Y/N]? ').lower()
293     if clean == 'y':
294         os.system(f'rm posfile momfile jfile inpsd.dat inp.{simid}.json uppasd.{simid}.yaml')
```

C Input files

Although most of the input files are automatically generated using the code in B, some are simply to big to be included in the syntax. It is therefore easier to make a template which is read by the script and edited to create the desired inputs. The input for the `inpsd.dat`-file was generated using the following template. The $ indicate a parameter read and inputed by the Python code.

```
simid $simid
ncell     $system                          System size
BC        P         P         P            Boundary conditions
                                           (0=vacuum, P=periodic)

cell      $cell
Sym       $sym                             Symmetry of lattice
                                           (0 no, 1 cubic, 2 2d cubic,
                                                        3 hexagonal)
do_prnstruct 1
maptype 2                                  Specify coordinates for jfile
                                           (1 cartesian, 2 lattice)
posfile    ./posfile
posfiletype $posfiletype
momfile    ./momfile
exchange   ./jfile

SDEalgh   5                                SDE solver
                                             (1=midpoint, 2=heun, 3=heun3,
                                               4=Heun_proper, 5=Depondt)

Initmag   $initmag                         Initial config of moments
                                           (1=random, 2=cone, 3=spec., 4=file)

ip_mode   N
ip_mcanneal 1
          10000 100.0 1.00e-16 0.95
          100 100.0 1.00e-16 0.95
          100 100.0 1.00e-16 0.95

mode      S
Temp      $temp          K               Temperature of the system
hfield    0.00000   0.00000   0.00000     Static H field

damping   0.50                           Damping parameter (gamma)
nstep     500000                         Number of time-steps
timestep  1.000e-16       s              The time step-size for
                                           the SDE-solver

do_avrg   Y                                Measure averages

do_autocorr Y
acfile ./twfile
```