

# Exercise 2, TFY4235 Computational physics

Martin Johnsrud

## Introduction

This report documents the simulation of magnons, as described in [1]. Using Heun's method, the Landau-Lifshitz-Gilbert equations for a classical, interacting ensemble of spins are integrated in time. The algorithm is first investigated by simulating one spin, for which the exact solution is known. The algorithm is then used to investigate the formation of collective modes, magnons, the difference between ferromagnetic and anti-ferromagnetic materials and magnetization.

## Theory

### Units

The Hamiltonian, as well as the equations of motion in [1] defines some natural units for the problem:

- Energy:  $[\mathcal{H}] = [J\hbar^2 s^2]$
- Magnetic field and magnetization:  $[\vec{B}] = [\vec{M}] = [\mu\vec{S}]$
- Anisotropy:  $[d_z] = [J]$
- Time :  $[t] = [\gamma J\hbar s]^{-1}$

where  $s$  is the spin of the particles. ( $s = 1/2$  for electrons) This is included so that  $|\vec{S}| \in [0, 1]$ . The defining dimensionfull constants of the system is thus the spin  $\hbar s$ , the coupling  $J$ , the magnetic moment  $\mu$  and the gyromagnetic ratio  $\gamma$ . These units are used throughout the exercise, including in all figures. Physical values for a given, actual system is then dependent on its values for  $s, J, \mu$  and  $\gamma$ .

### Indices

For easy implementation, the Hamiltonian can be written on index form and using the units as described above

$$\mathcal{H}(S; d_z, a, B) = -\frac{1}{2}J \sum_{\langle i,j \rangle, a} S_{i,a} S_{j,a} - d_z \sum_j (S_{j,3})^2 - \sum_{j,a} B_{j,a} S_{j,a}.$$

Here,  $J \in \{-1, 0, 1\}$ ,  $i \in \{1, \dots, N\}$  is the site index,  $a$  is vector component index. The effective field can be written

$$H_{k,b} = -\frac{\partial \mathcal{H}}{\partial S_{k,b}} = \frac{1}{2}J \sum_{\langle i,j \rangle, a} (S_{i,a} \delta_{j,k} \delta_{a,b} + S_{j,a} \delta_{i,k} \delta_{a,b}) + 2d_z \sum_j S_{j,3} \delta_{b,3} \delta_{j,k} + \sum_{j,a} B_{j,a} \delta_{k,b},$$

using the vector triple product identity  $\vec{A} \times (\vec{B} \times \vec{C}) = (\vec{A} \cdot \vec{B})\vec{C} - (\vec{A} \cdot \vec{C})\vec{B}$ . The first sum becomes

$$\frac{1}{2} \sum_{\langle i,j \rangle, a} (S_{i,a} \delta_{j,k} \delta_{a,b} + S_{j,a} \delta_{i,k} \delta_{a,b}) = \frac{1}{2} \sum_{\langle i,j \rangle} (S_{i,b} \delta_{j,k} + S_{j,b} \delta_{i,k}) = \frac{1}{2} \sum_{\langle j,i \rangle} 2S_{i,b} \delta_{j,k} = \sum_{j \in \text{NN}_k} S_{j,b},$$

where  $\text{NN}_k$  are the set of nearest neighbours of lattice point  $k$ . The Landua-Lifshitz-Gilbert equation for the time evolution of the system is then

$$\frac{d}{dt}S_{j,a} = -\frac{1}{(1+\alpha^2)} \left[ \sum_{bc} \varepsilon_{abc} S_{j,b} H_{j,c} + \alpha \sum_b (S_{j,b} S_{j,b} H_{j,a} - S_{j,b} H_{j,b} S_{j,a}) \right], \quad (1)$$

$$H_{k,b} = J \sum_{j \in \text{NN}_k} S_{j,b} + 2d_z S_{k,3} \delta_{k,3} + B_{k,b}. \quad (2)$$

## Implementation

The main object of the simulation is a NumPy-array  $\mathbf{S}$  of shape  $(T, N, 3)$ . This contains the components of each of the  $N$  spins at each timestep. The function `integrate` then runs a loop, calling the implementation of Heun's method `heun_step`. The index notation laid out in the Theory section allows for straight forward implementation of the LLG equation using NumPy's `einsum`-function. For example, using a array `eijk[a, b, c]` for the Levi Civita symbol,

$$\sum_{bc} \varepsilon_{abc} S_{j,b} H_{j,c}$$

becomes

```
np.einsum("...ac, ...c-> ...a", np.einsum("abc, ...b -> ...ac", eijk, S), H).
```

`LLG` takes as arguments  $\mathbf{S}$  and the needed parameters. Then, it first evaluates the first sum of (1) as shown above. If  $\alpha \neq 0$ , it then evaluates the second sum. `LLG` calls `get_H`. This functions implements (2), using NumPy's `roll`-function to sum over all nearest neighbours. `LLG` then returns `dtS`, a NumPy-array containing the time derivative of  $\mathbf{S}$ . This implementation makes it easy to generalize to spins distributed in 2 or 3 dimensions, as is done in `3D.py`. The only change needed is to extend the shape of  $\mathbf{S}$  to  $(T, N, N, N, 3)$ , and change `get_H` to sum over nearest neighbours in all dimensions.

The library `mayavi` is used to make 3D visualization of the spins, as in figure 5 and to make videos of the time evolution of the spins.

## Results

### Single spin

The first test of the simulation is to initialize a single spin, in a magnetic field  $B = (0, 0, 1)$ . This spin is given a slight tilt, with initial conditions  $(\theta, \phi) = (0.5, 0)$ . The expectation is that the spin will precess in a circle around the  $z$ -axis, with a Larmor frequency  $\omega = -\gamma B$ , (REFERANSE) due to the units as described in the subsection on units. Figure 1 shows the  $x, y$ -components of this spin, as a function time, together with the expected analytical result.

To analyze the error, the simulation is run with different step lengths  $h$ , for the same simulation time  $t_0 = 5$ . The result is shown in figure 2. As Heun's method is of higher order than Euler's method, it converges faster. It is, however, necessary to make 2 function calls when using Heun's method, while Euler's method only require one. This should make Euler's method twice as fast, which was observed. Euler's method ran at around 16000 iterations per second, while Heun's method only ran at 8000. The large gain in precision, however, makes Heun's method preferred for this application.

When including  $\alpha > 0$ , one should expect the the oscillations to die away, with a lifetime given by

$$\tau = \frac{1}{\alpha\omega}$$

Larger  $\alpha$  should give a shorter lifetimes, and thus faster decay. Figure 3 shows this. Furthermore, we see that the amplitude is proportional to  $\exp(-t/\tau)$ . We should expect this, not only is this a common form for decay, but as no time is special, the decay should be proportional to the amplitude, which gives exponential decay.

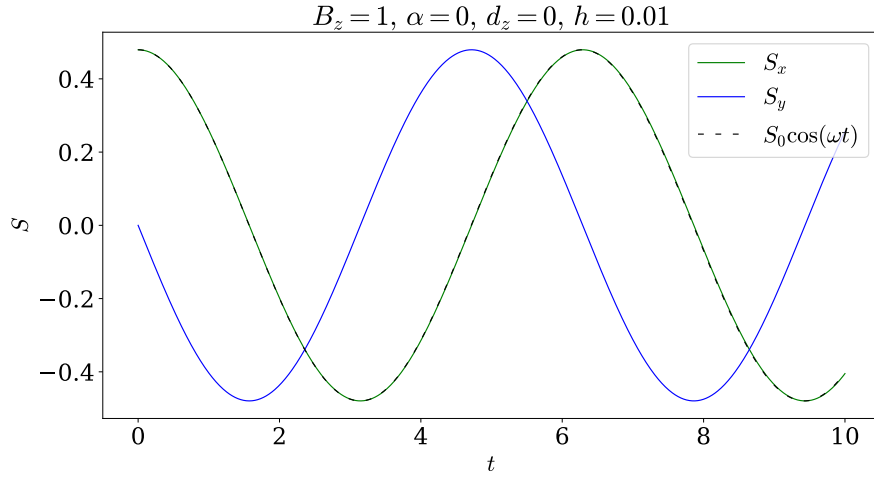


Figure 1: caption

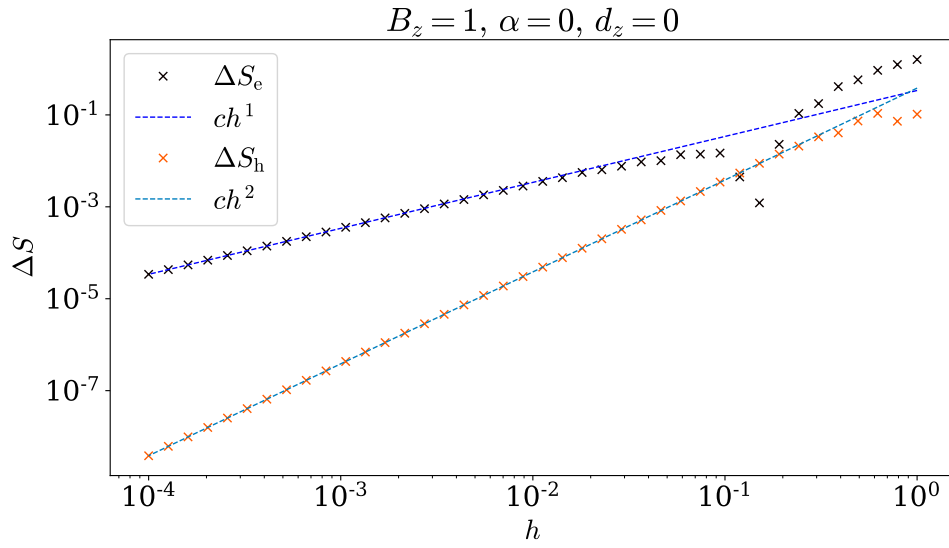


Figure 2: caption

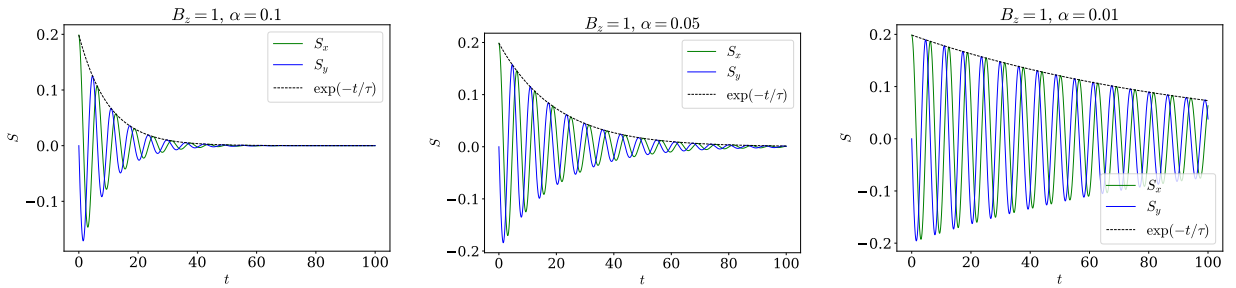


Figure 3: caption

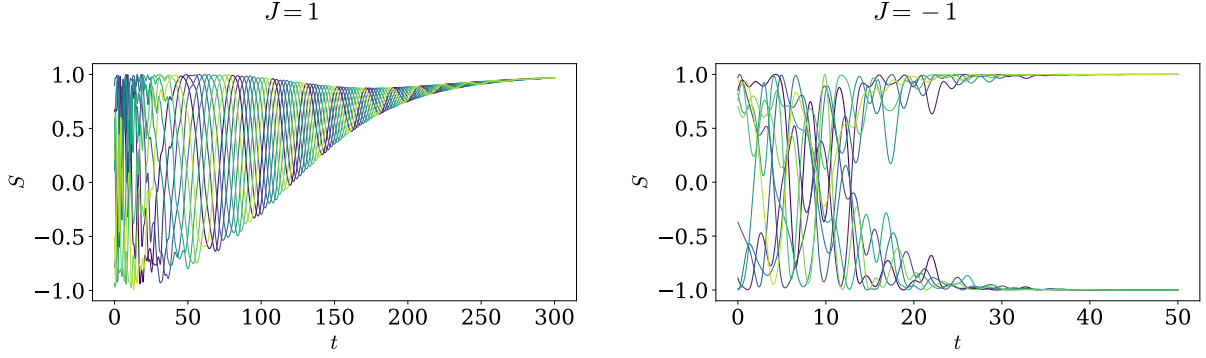


Figure 4: caption

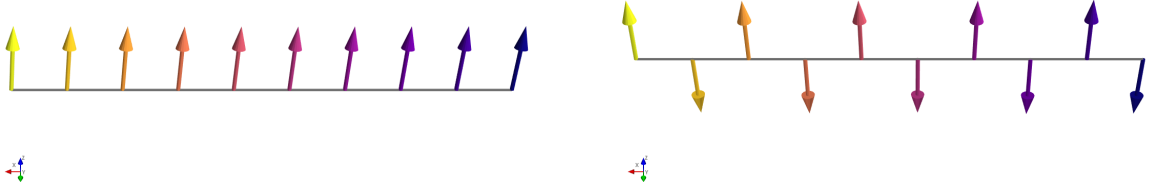


Figure 5: caption

## Spin chain

The simulation now includes several spins, in a ferromagnetic or anti-ferromagnetic coupling, depending on if  $J = 1$  or  $J = -1$ , respectively. When including damping  $\alpha > 0$ , both these settle into the ground state of the system, after some time. This is shown in figure 4. However, the two systems have different ground states. In the ferromagnetic case, the lowest energy configuration is the alignment of all the spins, while in the anti-ferromagnetic case the spins are oppositely aligned. The final configuration is shown in figure 5.

When the coupling is turned off,  $J = 0$ , but  $d_z > 0$  and initializes one spin tilted, it will precess alone, as shown in the left in figure 6. However, if the coupling is turned on again, as shown on the right, the disturbance will ripple through the chain, in a wave. This happens as the tilted spin makes it energetically advantageous for its neighbours to tilt towards it, starting a chain reaction that propagates through the chain.

We can see that the vibrations in the chain are dominated by high frequency oscillations. By looking back on the damping, as shown in figure 7 on the left, the energetically costly high frequencies die out fast, and we are left with the fundamental frequencies of the system. The plot on the right illustrates the sum of the  $x$ -component of all the spins. This follows the long-term pattern immediately. The dotted line is curve-fitted to the function  $f(t; A, \alpha, \tau) = A \cos(\omega t) e^{-t/\tau}$ .

The left plot in figure 8 shows the same situation, but with an antiferromagnetic coupling  $J = -1$ . This system starts in a much higher energy state, and thus becomes highly excited. However, the excitation dies out much faster than in the ferromagnetic case. The right plot shows the same system, but starting in the ground state, except one tilted spin. Here too, the excitation dies out much faster, without a surviving oscillation.

The magnetization of the state is, in the units previously described, given by

$$M_a = \frac{1}{N} \sum_j S_{j,a}.$$

Thus, the magnetization of the ground state is in the ferromagnetic case  $\vec{M} = (0, 0, 1)$ , while in the anti-ferromagnetic case  $\vec{M} = (0, 0, 0)$ . This means that a disturbance in the ground state of the ferromagnet, like

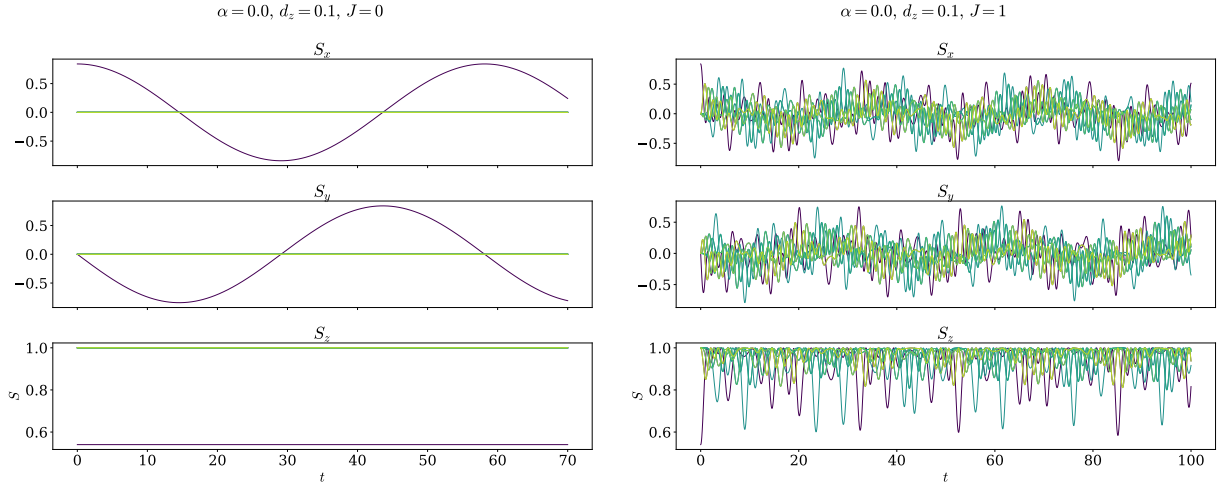


Figure 6: caption

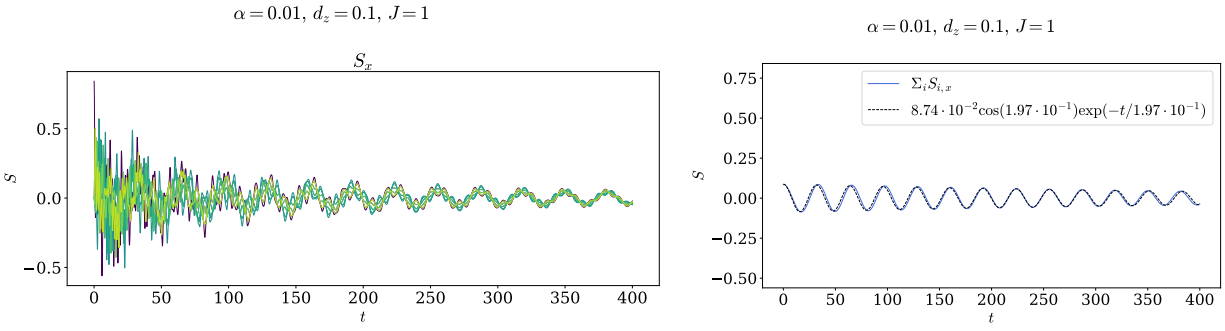


Figure 7: caption

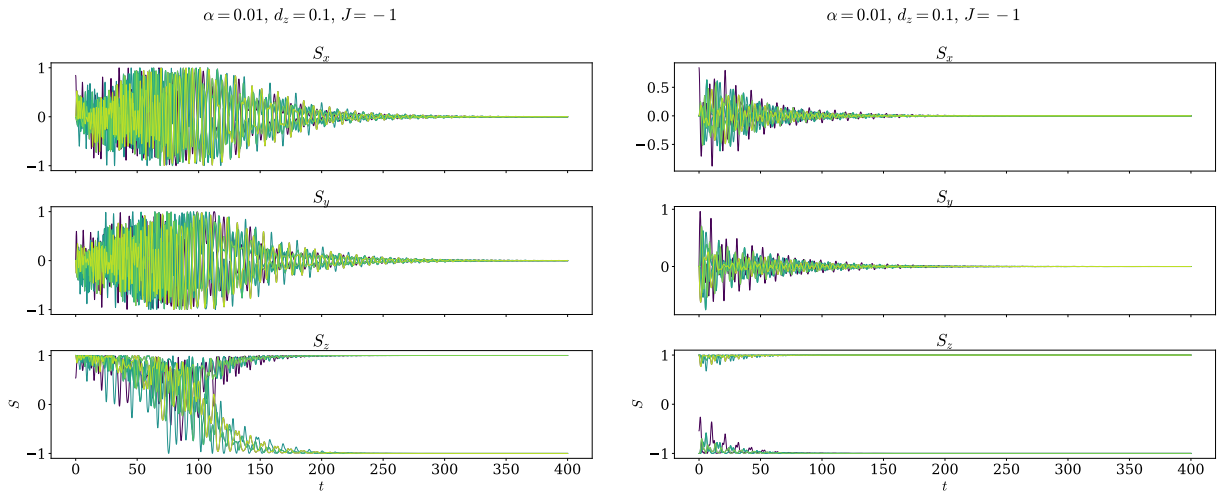


Figure 8: caption

$$\alpha = 0.01, d_z = 0.1, J = 1$$

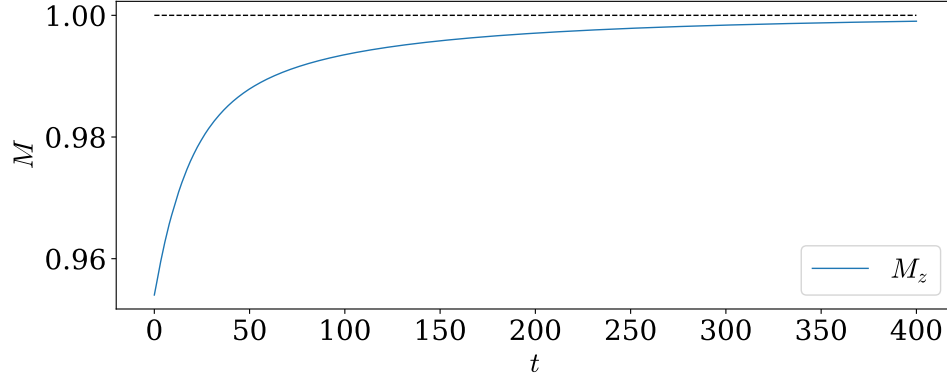


Figure 9: caption

the sustained magnon oscillation we showed earlier, will result in a sustained loss of magnetization. This is due to the fact that it is the state of maximum magnetization. This is shown in figure 9. Here, the magnetization quickly approaches 1 as the energy is dissipated. However, as the system settles in to the mode of sustained oscillation, as noted earlier, it retains some loss in magnetization.

## Conclusion

Using Heun's method, and the LLG equations materials of interacting spins have been investigated. Ferromagnetic and antiferromagnetic states have different ground states, and anti-ferromagnets reach their groundstate almost 10 times faster through dissipation than their ferromagnetic counterpart. Collective modes, magnons, is shown to exist in ferromagnetic materials. These modes are clearly visible in systems which are disturbed from the ground state, and have some dissipation. They affect the magnetization of the material, as they disturb it away from its highest magnetization state, the ground state. The simulation confirms this.

## References

- [1] Exercise 2, 2021, tfy4235 computational physics.