Exercise 2, TFY4235 Computational physics

Martin a. Johnsrud

Introduction

This report documents the simluation of magnons, as described in [1].

Theory

Units

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

- Energy: $[\mathcal{H}] = [J\hbar^2 s^2]$
- Magnetic field: $[\vec{B}] = [\mu \vec{S}]$
- 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 cupling J, the magnetic moment μ and the gyromacnetic ratio γ.

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

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

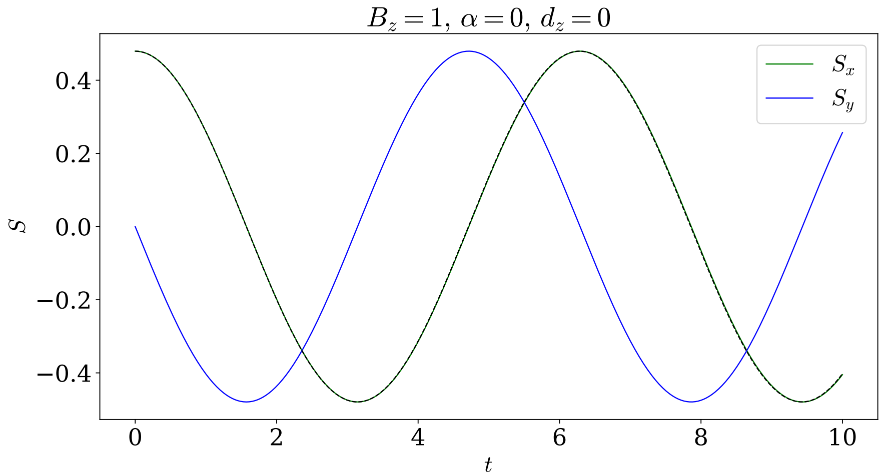
using the vector triple product identity $\vec{A} \times (\vec{B} \times \vec{C}) = (\vec{A} \cdot \vec{B})\vec{C} - (\vec{A} \cdot \vec{C})\vec{B}$. The first sum becomes

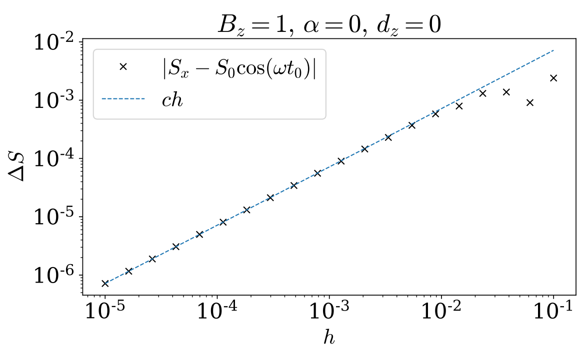
$$\frac{1}{2} \sum_{\langle i,j \rangle, a} (S_{i,a} \delta_{j,k} \delta_{a,b} + S_{j,a} \delta_{i,k} \delta_{a,b}) = \frac{1}{2} \sum_{\langle i,j \rangle} (S_{i,b} \delta_{j,k} + S_{j,b} \delta_{i,k}) = \frac{1}{2} \sum_{\langle j,i \rangle} 2 S_{i,b} \delta_{j,k} = \sum_{j \in \mathrm{NN}_k} S_{j,b},$$

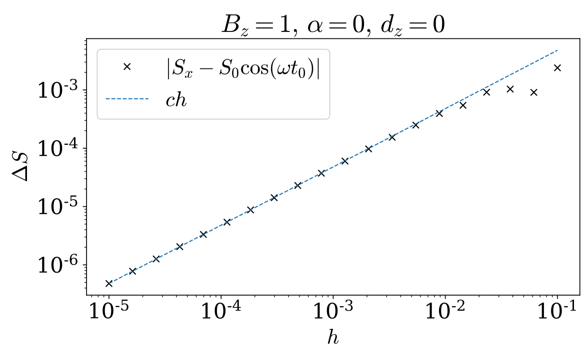
where NN_k are the set of nearest negihbours of lattice point k. The Landua-Lifshitz-Gilbert equation for the time evolution of the system is then

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

$$H_{k,b} = J \sum_{j \in \mathrm{NN}_k} S_{j,b} + 2 d_z S_{k,3} \delta_{k,3} + B_{k,b}. \tag{2}$$







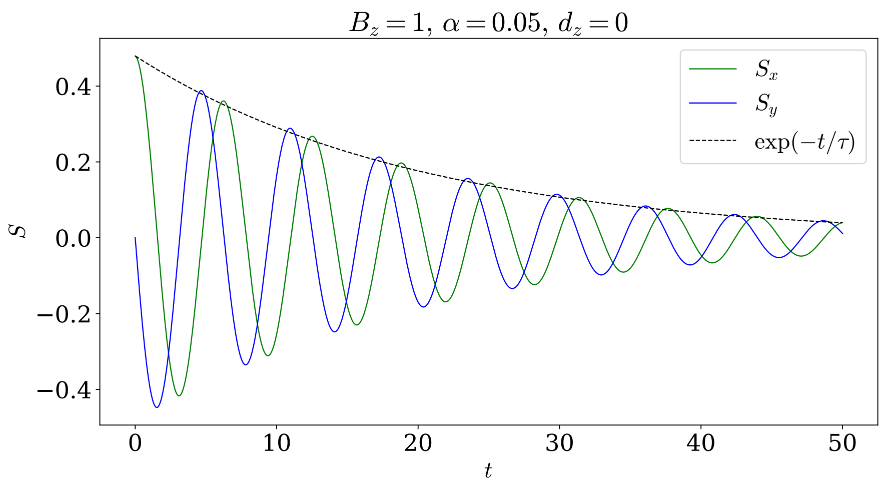
Implementation

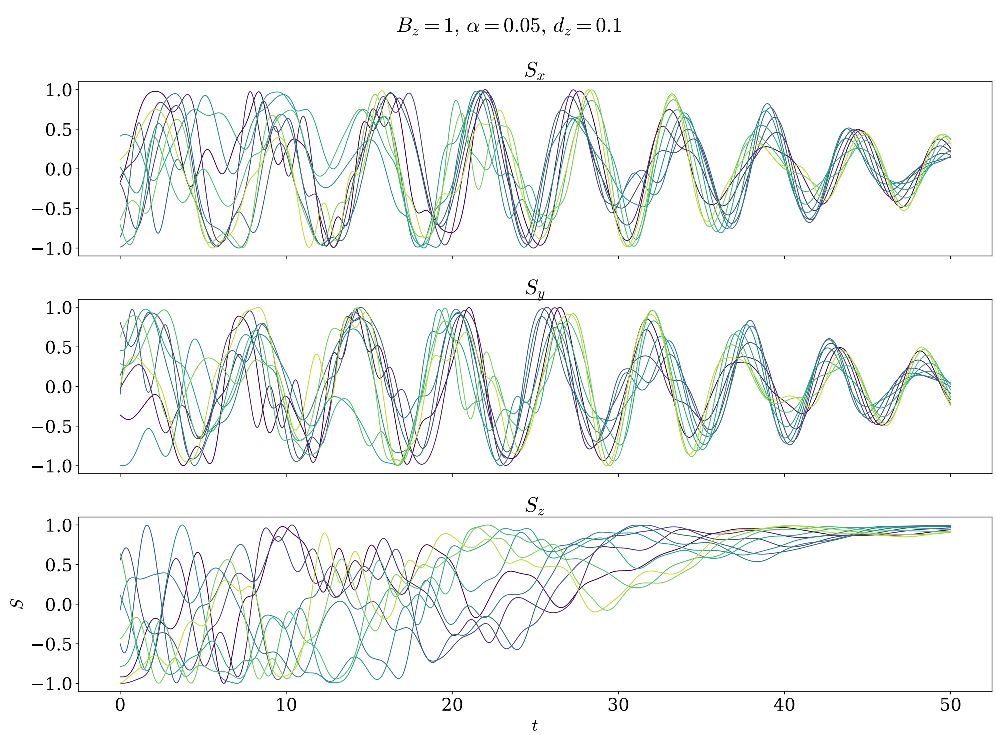
The main object of the simulation is a NumPy-array `S` of shape `(T, N, 3)`. This contains the components of each of the N spins at each timestep. The function `integrate` then runs a loop, calling the implementation of the Heun method `heun_step`. The index notation laid out in the Theory section allows for straight forward implementation of the LLG equation using NumPy's `einsum`-function. `LLG` takes as arguments `S` and the needed parameters. Then, it first evaluates the first sum of (1) using two nested `einsum`-functions, as well as an implementation of the Levi-Civita tensor. 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 negihbours. `LLG` then returns `dtS`, a NumPy-array containing the time derivative of S.

Results

References

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


$B_z = 1,\ \alpha = 0.05,\ d_z = 0$
S_x
S_y
$\exp(-t/\tau)$
S
t


$B_z = 1,\ \alpha = 0.05,\ d_z = 0.1$
S_x
S_y
S_z
S
t