Mid-Term Report

PH19B011: Vismay Churiwala

October 2023

1 Introduction

1.1 Sparse Identification of Non-Linear Dynamics(SINDy)

- The detection of governing equations is a central problem across many areas of research. A large number of systems that are of interest are differential equations with a relatively few number of dominant terms, i.e., their governing dynamics is determined by equations that are sparse in the space of all possible functions.
- SINDy illustrates how we identify the dynamics of a system using a data-driven approach, using sparse-regression to determine the governing equations using the least number of terms, balancing accuracy with model-complexity to avoid over-fitting [1].

1.2 Magnonics

- Spin waves are collective excitations in magnetically ordered materials. The study of spin waves in called magnonics. In essence, spin waves are a propagating re-ordering of the magnetisation in a material and arise from the precession of magnetic moments [2].
- The dynamics of the spins of the system is determined by the Landau-Lifshitz-Gilbert Equation(LLG-equation), which is as follows:

$$\frac{\partial m}{\partial t} = -\gamma m \times H_{eff} + \alpha m \times \frac{\partial m}{\partial t} \tag{1}$$

  The constant α is the Gilbert phenomenological damping parameter and depends on the solid, and γ is the electron gyromagnetic ratio. Here $m = M/M_s$. The second term represents the damping that causes the amplitude of the spin waves to reduce over distances.
- The LLG-equation can be approximated to one or more orders and this provides an instance of a dynamical system from which governing equations can be extracted using SINDy. The time-series can be obtained from software like OOMPH or NMAG, and fed into the SINDy with the proper basis functions to extract a sparse model.

2 Literature Review

2.1 SINDy [1]

2.1.1 Background

- A new approach to the long sought-after problem of discovering dynamical systems from data is from the perspective of sparse regression and compressed sensing. The fact that most physical systems have only a few relevant terms that define the dynamics can be leveraged, making the governing equations sparse in a high-dimensional nonlinear function space[1].
- Here, we consider dynamical systems of the form:

$$\frac{d}{dt}x(t) = f(x(t)) \tag{2}$$

  where $x(t)$ is the state vector at time t, and $f(x(t))$ defines the dynamic constraints that define the equations of motion of the equation.
- Recent advances in compressed sensing and sparse regression make the viewpoint of sparsity favorable, because it is now possible to determine which right-hand-side terms are nonzero without performing a combinatorially intractable brute-force search.

2.1.2 Methodology

- To determine the function f from data, we collect a time history of the state x(t) and either measure the derivative $\dot{x}(t)$ or approximate it numerically from x(t). The data are sampled at several times $t1, t2, ..., tm$ and arranged into two matrices:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}^T(t_1) \\ \mathbf{x}^T(t_2) \\ \vdots \\ \mathbf{x}^T(t_m) \end{bmatrix} = \overrightarrow{\text{state}} \begin{bmatrix} x_1(t_1) & x_2(t_1) & \cdots & x_n(t_1) \\ x_1(t_2) & x_2(t_2) & \cdots & x_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(t_m) & x_2(t_m) & \cdots & x_n(t_m) \end{bmatrix} \downarrow \text{time}$$

$$\dot{\mathbf{X}} = \begin{bmatrix} \dot{\mathbf{x}}^T(t_1) \\ \dot{\mathbf{x}}^T(t_2) \\ \vdots \\ \dot{\mathbf{x}}^T(t_m) \end{bmatrix} = \begin{bmatrix} \dot{x}_1(t_1) & \dot{x}_2(t_1) & \cdots & \dot{x}_n(t_1) \\ \dot{x}_1(t_2) & \dot{x}_2(t_2) & \cdots & \dot{x}_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ \dot{x}_1(t_m) & \dot{x}_2(t_m) & \cdots & \dot{x}_n(t_m) \end{bmatrix}.$$

Figure 1: State Vectors[1]

- Next, we construct a library $\Theta(X)$ consisting of candidate nonlinear functions of the columns of X. For example,$\Theta(X)$ may consist of constant,

polynomial, and trigonometric terms:

$$\Theta(\mathbf{X}) = \begin{bmatrix} | & | & | & | & | & | & | & | \\ 1 & \mathbf{X} & \mathbf{X}^{P_2} & \mathbf{X}^{P_3} & \cdots & \sin(\mathbf{X}) & \cos(\mathbf{X}) & \cdots \\ | & | & | & | & | & | & | & | \end{bmatrix}$$

Figure 2: Library functions [1]

where X^{P_2}, X^{P_3}... are higher order polynomials in the state $\mathbf{x}$

- This sets up a sparse regression problem where we need to determine the sparse vectors of coefficients $\Xi = \xi_1 \xi_2 .. \xi_n$ to determine which non-linearities are active:

$$\dot{\mathbf{X}} = \Theta(\mathbf{X})\Xi \tag{3}$$

  Once the coefficients Ξ are determined, a model of each row of the governing equations can be constructed in terms of the symbolic functions of elements of $\mathbf{x}$

- Each column of Equation (3) requires a distinct optimization to find the sparse row of coefficients ξ_k for the kth equation.

2.1.3 Solivng PDEs of High-Dimensionality:

- Often, the physical system of interest may be naturally represented by a partial differential equation (PDE) in a few spatial variables. But for a few cases like fluid simulations, the number of variables may be extremely large(billions), and the optimization of each variable would be ill-suited for this type of sparse regression.

- Most of such systems evolve on a low-dimensional manifold or attractor that is well approximated using a low-rank basis Ψ. We can obtain such a low-rank approximation using dimensionality reduction techniques, such as Proper Orthogonal Decomposition(POD).

- Simply put, obtaining sparse dynamics requires the right coordinate transformation and basis functions, and it is not always clear what these are a priory. These may be informed by machine learning, data mining or partial knowledge of physics or human intuition.

- Identification of Sparse dynamics in many complex systems have been demonstrated using SINDy, including chaotic lorentz system, PDE for vortex shredding behind an obstacle etc.

- It is a versatile tool, and several tools can be used to extend functionality of the main algorithm, such as dimensionality reduction, denoising of the data and derivatives, change of coordinates, discovery of normal forms in case of bifurcations, different optimizers etc.

3 Identifying the LLG Equation using SINDy

3.1 Setting up the problem

- The 3 dimensional LLG equation can be solved using odeint package in python, provided we know the H_{eff} and the magnetic properties of the material.
- The 3 components of magnetisation are determined by the LLG equation, where we can approximate it by considering terms upto the 2nd order. The resulting equation is:

$$\frac{\partial m}{\partial t} = -\gamma m \times H_{eff} - \alpha m \times (\gamma m \times H_{eff}) \tag{4}$$

- Now, we consider a point-system, where H_{eff} can be determined by the demagnetisation tensor(N) as follows:

$$H_{eff} = H_{ext} - N.M \tag{5}$$

  The demagnetisation tensor depends on the geometry of the magnetic body

3.2 Simulation of LLG equation

- We now simulate the LLG equation for different materials and different geometries, and obtain time-series data of the 3 components of the magnetisation and feed it to SINDy.
- Here, we have used values of $\gamma = 1.760$, and starting from the magnetisation (1,0,0), we apply an external magnetic field of (0,0,1) at t=0, and evaluate the LLG equation using the scipy library odeint to obtain the time-series of 1000 time-steps, from t=0 to t=100.

3.2.1 No Demagnetisation field

$\alpha = 0.02, \mathbb{N} = (0, 0, 0)$

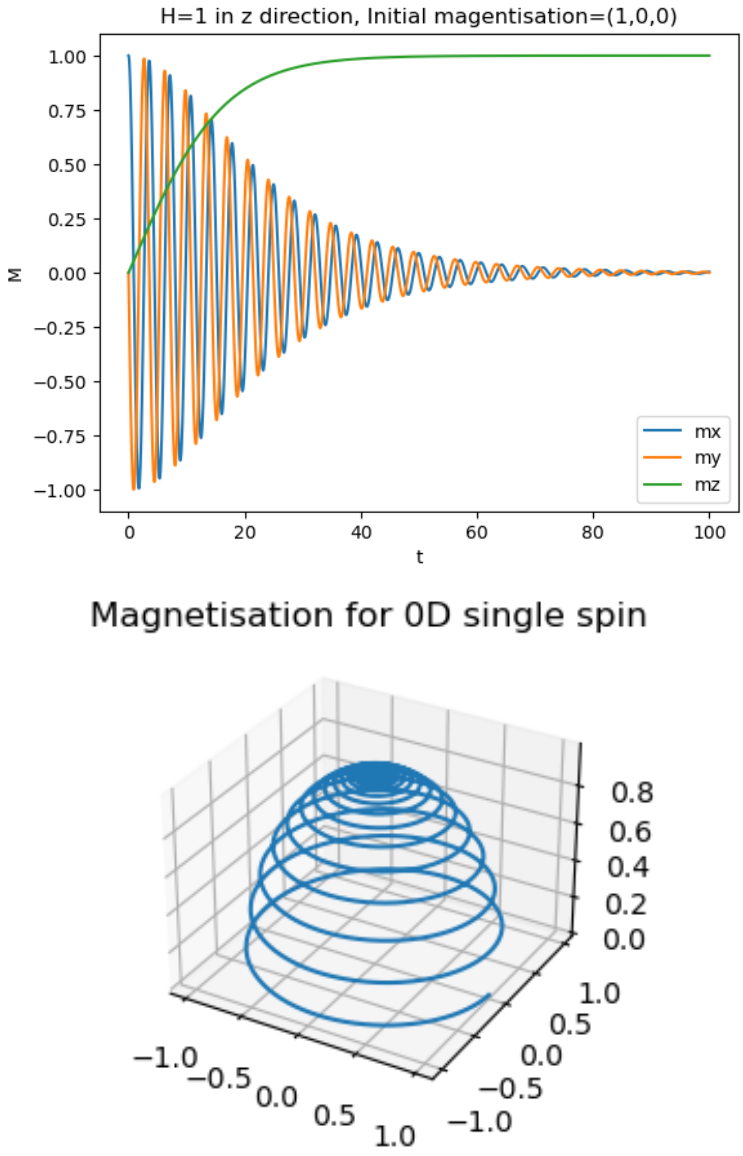


Figure 3: odeint simulation

SINDy was fed the time series data, and the sparse regression was done with the polynomial library of degree 2 and using STLSQ as the optimizer to obtain the model:

```
model.print()
#Expectation:

# mx'= 1.76 (my - 0.0352 mx mz),
# my'= 1.76 (-mx - 0.0352 my mz),
# mz'=-0.0352 (-1.76 mx^2 - 1.76 my^2)
```

```
(mx)' = 1.760 my + -0.062 mx mz
(my)' = -1.760 mx + -0.062 my mz
(mz)' = -0.017 1 + 0.079 mx^2 + 0.079 my^2 + 0.017 mz^2
```

Figure 4: SINDy model

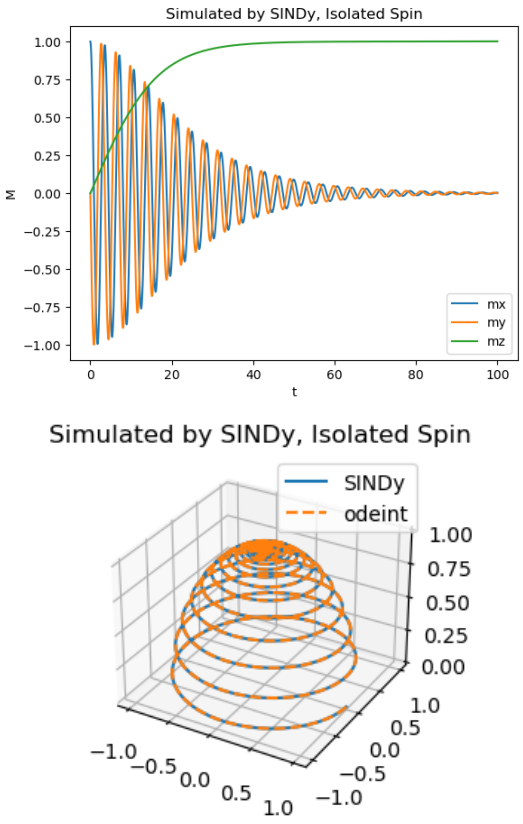
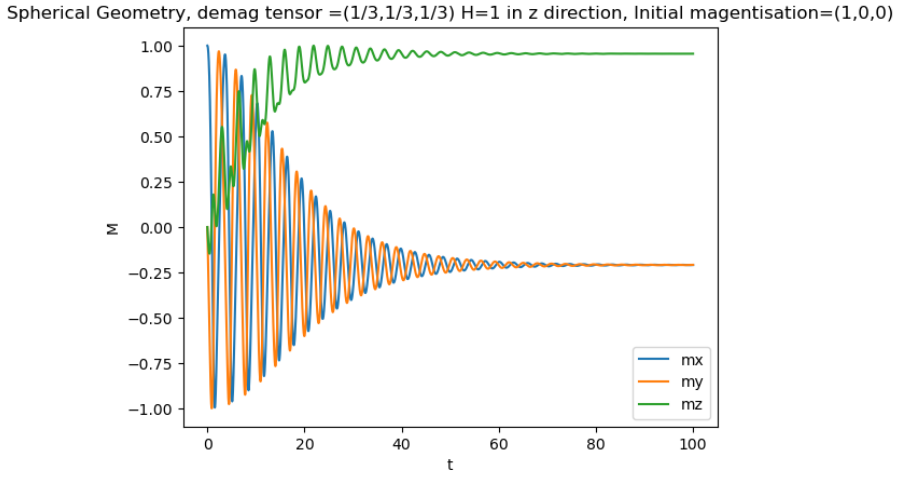


Figure 5: SINDy model simulation, Isolated Spin, E_{RMS} =7.084093e-06

3.2.2 Spherical Geometry

$\alpha = 0.02$, $\mathbb{N} = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$



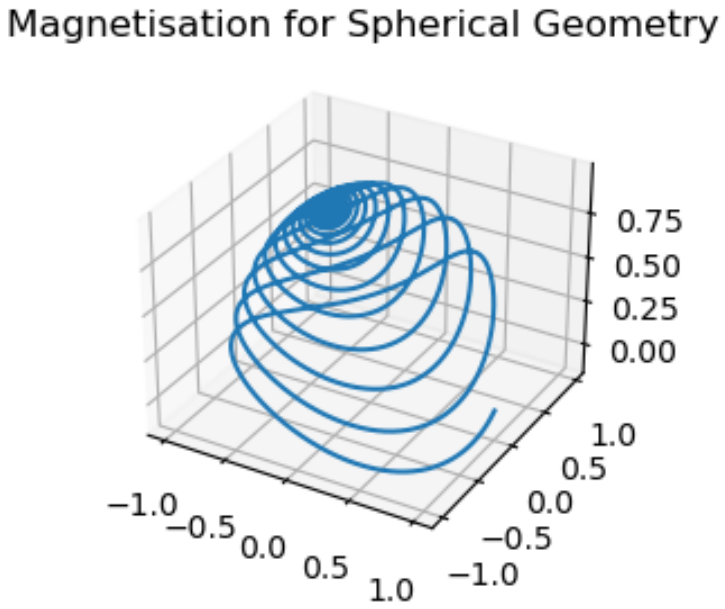
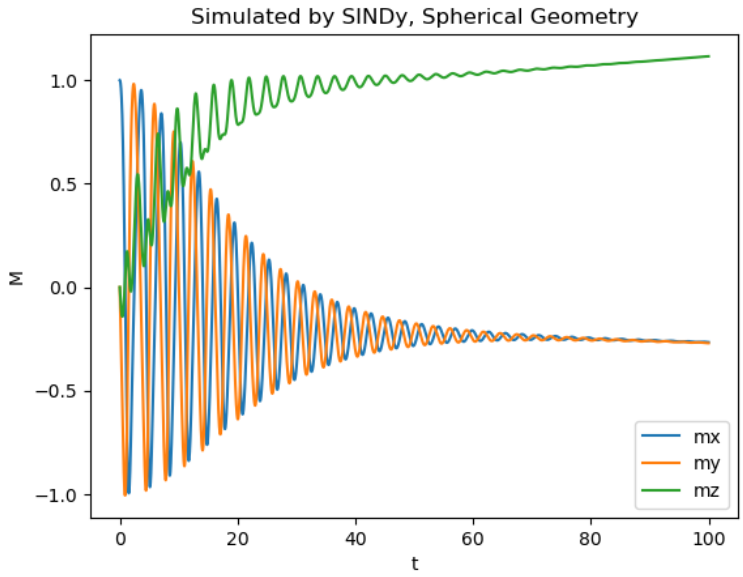


Figure 6: odeint simulation

Using the same optimizer and polynomial basis functions as in the previous case, the SINDy model so obtained is:

```
(mx)' = 1.746 my + -0.575 mx my + 0.520 mx mz + -0.589 my^2 + 0.568 mz^2
(my)' = -1.773 mx + 0.585 mx^2 + 0.597 mx my + -0.653 my mz + -0.605 mz^2
(mz)' = -0.524 mx^2 + -0.577 mx mz + 0.651 my^2 + 0.596 my mz
```

Figure 7: SINDy model, Spherical Geometry



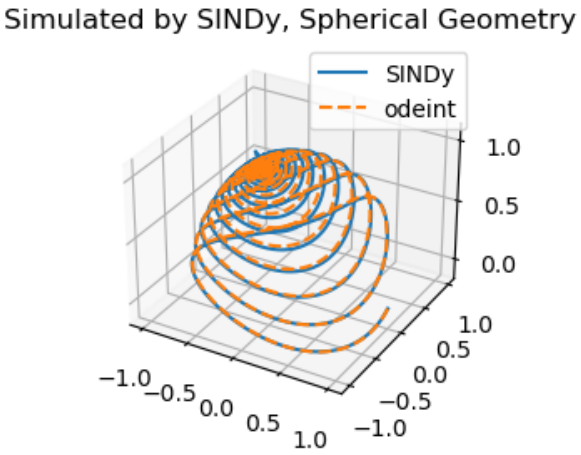


Figure 8: SINDy Model, Spherical Geometry, E_{RMS} =0.00097328

3.2.3 Errors in SINDy models

We can see that for low damping(α=0.02), we get remarkably accurate models using SINDy. Note that the algorithm is not provided with any parameters or form of the solutions to the LLG equation except the basis functions, and it is still able to come up with the PDE and simulate it, given initial conditions. Now let us see how SINDy performs for various degrees of damping.

Isolated Spin:

Alpha	E_{RMS}
0.02	0.000311
0.04	0.00012
0.06	7.390e-05
0.08	5.882e-05
0.10	5.254e-05
0.12	4.961e-05
0.14	4.817e-05
0.16	4.746e-05
0.18	4.713e-05
0.20	4.703e-05

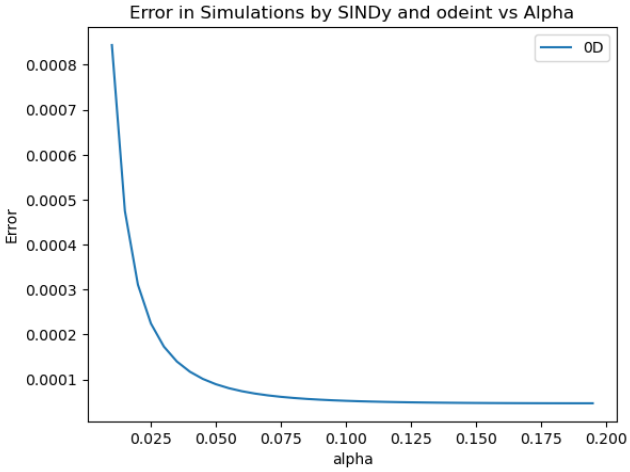


Figure 9: E_{RMS} vs α, Isolated Spin

We see that the model is performing reasonably well across values of α

Spherical Geometry:

Alpha	E_{RMS}
0.02	0.07843
0.04	0.05782
0.06	0.05758
0.08	0.05099
0.10	0.04868
0.12	0.02422
0.14	0.01222
0.16	0.00662
0.18	0.0109
0.20	0.0143

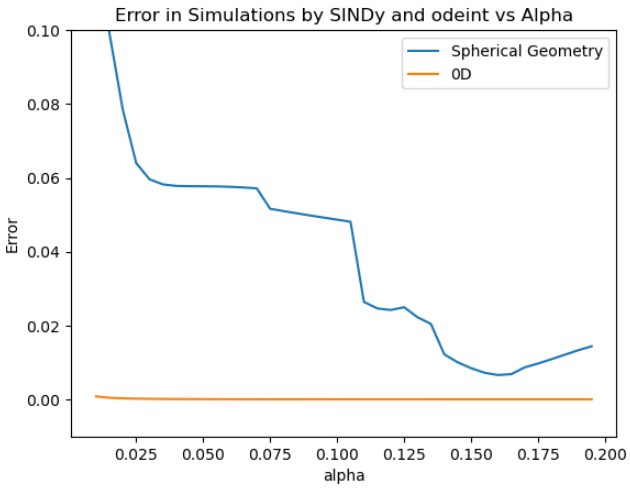


Figure 10: E_{RMS} vs α, Spherical Geometry

Note that for spherical geometry, the errors are higher as the model is more complex, but they are nevertheless reasonably small.

3.2.4 A more complex Geometry

We have encountered relatively simple geometries until now, with demagnetisation tensors $\mathbb{N} = (0, 0, 0)$ and $\mathbb{N} = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$. Now, let us consider a point inside a flat rectangular prism, where the demagnetisation tensor need not be diagonal.

The symmetric demagnetisation tensor $\mathbb{N}$ at (x,y,z) inside a flat rectangular prism of dimensions (a,b,c) has the components[3]:

$$\begin{aligned} &N_{ii}\left(\mathbf{r}\right) \\ &= \tfrac{1}{4\pi}[\arctan f_i\left(x,y,z\right) + \arctan f_i\left(-x,y,z\right) + \arctan f_i\left(x,-y,z\right) \\ &+ \arctan f_i\left(x,y,-z\right) + \arctan f_i\left(-x,-y,z\right) + \arctan f_i\left(x,-y,-z\right) \\ &+ \arctan f_i\left(-x,y,-z\right) + \arctan f_i\left(-x,-y,-z\right)] \end{aligned}$$

where,

$$f_x(x,y,z) = \frac{(b-y)(c-z)}{(a-x)[(a-x)^2+(b-y)^2+(c-z)^2]^{\frac{1}{2}}} \tag{6}$$

$$f_y(x,y,z) = \frac{(b-y)(c-z)}{(a-x)[(a-x)^2+(b-y)^2+(c-z)^2]^{\frac{1}{2}}} \tag{7}$$

$$f_z(x,y,z) = \frac{(c-z)(a-x)}{(a-x)[(a-x)^2+(b-y)^2+(c-z)^2]^{\frac{1}{2}}} \tag{8}$$

The off-diagonal terms are:

$$N_{ij}\left(\mathbf{r}\right) = -\tfrac{1}{4\pi}\ln\left[\frac{F_{ij}(\mathbf{r},a,b,c)F_{ij}(\mathbf{r},-a,-b,c)F_{ij}(\mathbf{r},a,-b,-c)F_{ij}(\mathbf{r},-a,b,-c)}{F_{ij}(\mathbf{r},a,-b,c)F_{ij}(\mathbf{r},-a,b,c)F_{ij}(\mathbf{r},a,b,-c)F_{ij}(\mathbf{r},-a,-b,-c)}\right], \quad i \neq j$$

Where,

$$F_{xy}(r,a,b,c) = (c-z) + [(a-x)^2+(b-y)^2+(c-z)^2]^{1/2} \tag{9}$$

$$F_{yz}(r,a,b,c) = (a-x) + [(a-x)^2+(b-y)^2+(c-z)^2]^{1/2} \tag{10}$$

$$F_{zx}(r,a,b,c) = (b-y) + [(a-x)^2+(b-y)^2+(c-z)^2]^{1/2} \tag{11}$$

$\alpha = 0.02$, Dimensions of Prism= $1 \times 1 \times 0.1$

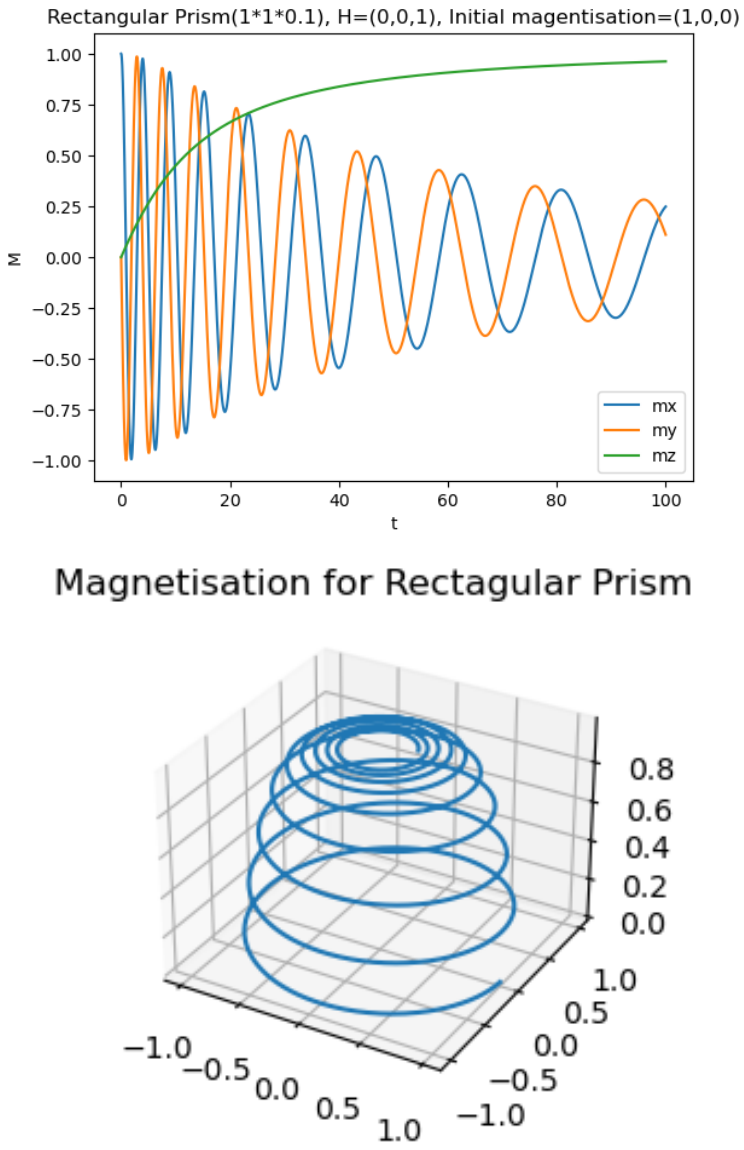
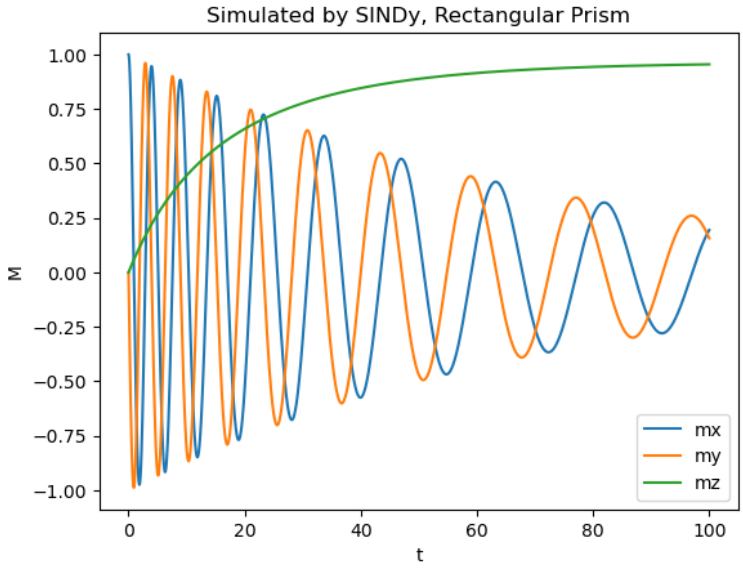


Figure 11: Magnetisation at (0,0,0) for a Rectangular Prism

We feed this time series data into SINDy to get the model:

```
(mx)' = -0.014 mx + 1.759 my + -1.522 my mz
(my)' = -1.761 mx + -0.014 my + 1.524 mx mz
(mz)' = 0.066 1 + -0.096 mz + 0.028 mz^2
```

Figure 12: SINDy Model



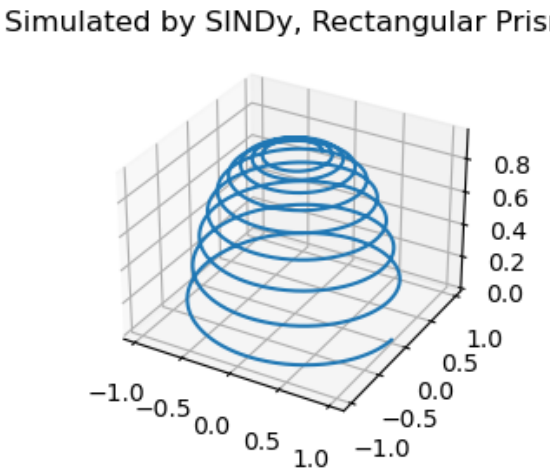


Figure 13: SINDy Model, Rectangular Prism, E_{RMS} =0.0008188

Here, we again see that the SINDy model is remarkably close to the actual equations. Let us now see how the RMS error changes as we vary the thickness of the prism.

Thickness	E_{RMS}
0.08	0.00169
0.16	0.0040
0.24	0.00449
0.32	0.00317
0.40	0.0.0020
0.48	0.00120
0.56	0.000691
0.64	0.00038
0.72	0.00020
0.80	0.00027

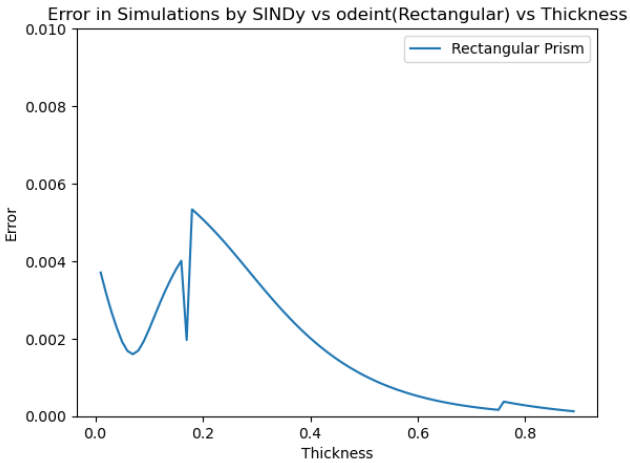


Figure 14: E_{RMS} vs thickness, Rectangular Prism

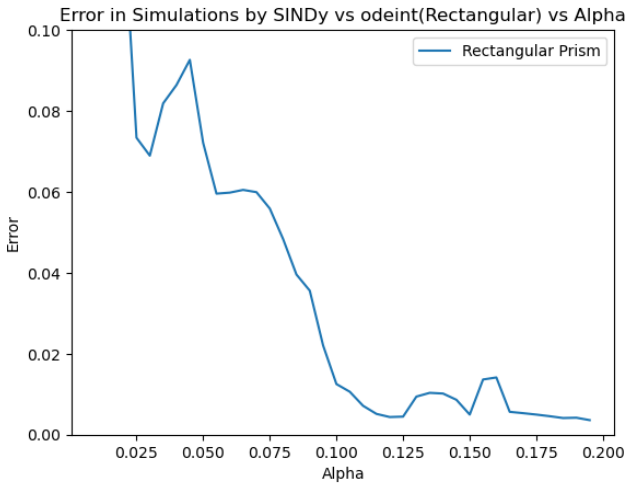


Figure 15: E_{RMS} vs alpha, Rectangular Prism

3.3 Reducing Dimensions

- The reduction of dimensions in the solutions to the LLG comes out naturally, without the use of techniques such as POD or PCA.
- This is due to the fact that the magnetization is normalized, and the state can be defined as a function of just (θ,ϕ).
- The LLG equation in this case can be represented as functions of θ and ϕ as:[4]

$$\dot{\theta} = \frac{d\theta}{dt} = \frac{\gamma\alpha}{\alpha^2+1}\left(H\sin\theta - H_k\sin\theta\cos\theta\right)$$

$$\frac{-d\varphi}{dt} = \left(\frac{\gamma}{\alpha^2+1}\right)\left(H - H_A\cos\theta\right).$$

- Here, I changed to the appropriate basis functions in SINDy by defining a custom function Library. Also, the derivates were fed directly to SINDy to compensate for the discontinuities in the angles at 0 and 2Π
- Now, we are considering switching times, i.e: The spin is initially oriented 10°from the z axis, and we need to determine the time it takes for it to reach 170°from the z axis. That is, $\theta = \frac{170}{180}\Pi$

3.3.1 Simulating

$\alpha = 0.1, \mathbb{N} = (0, 0, 0)$

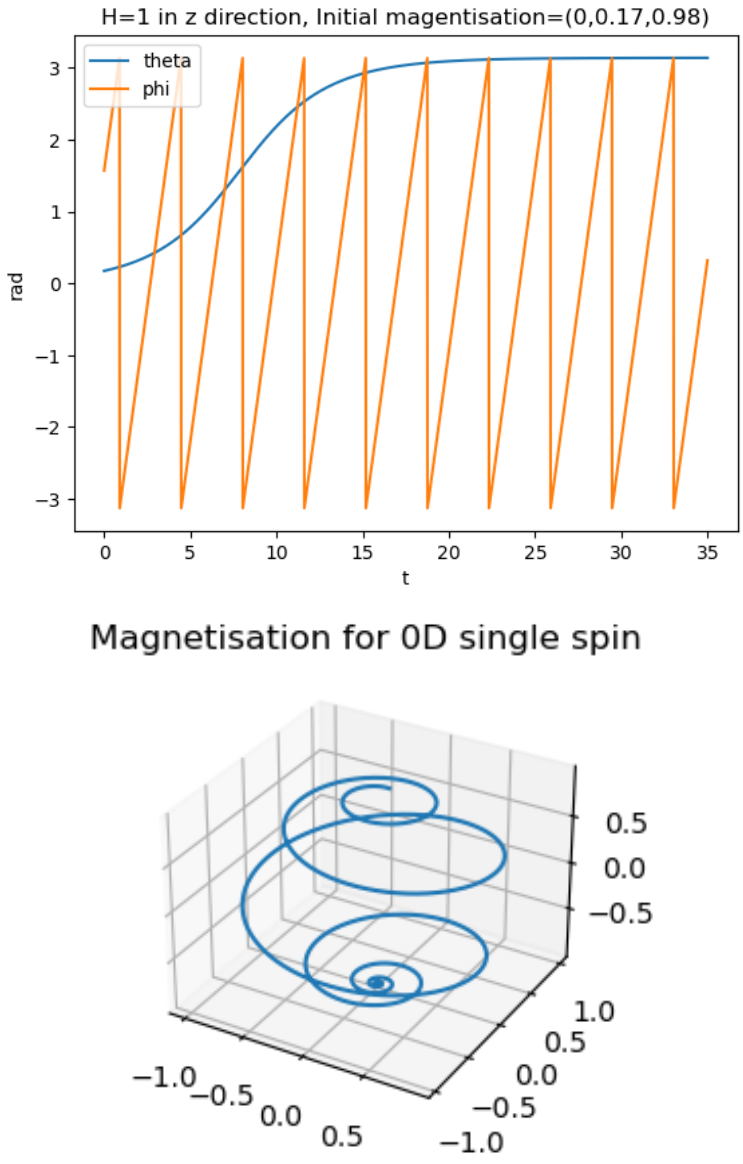


Figure 16: odeint simulation

SINDy was fed the time series data, and the sparse regression was done with the polynomial library of degree 2 and using STLSQ as the optimizer to obtain the model:

```
(theta)' = 0.310 sin(theta)
(phi)' = 1.760
```

Figure 17: SINDy model

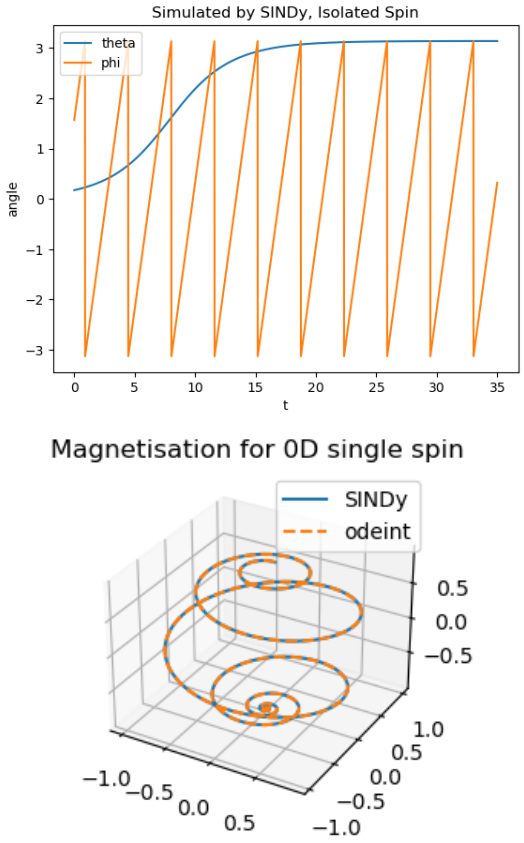


Figure 18: SINDy model simulation, Isolated Spin, E_{RMS} =7.084093e-06

3.3.2 Switching times

The switching times can be theoretically calculated as:[4]

$$\tau = \frac{\alpha^2+1}{\gamma\alpha}\frac{1}{H^2-H_k^2}\left[H\ln\left(\frac{\tan\theta_2/2}{\tan\theta_1/2}\right)\right.$$
$$\left. + H_k\ln\left(\frac{H-H_k\cos\theta_1}{H-H_k\cos\theta_2}\right) + H_k\ln\left(\frac{\sin\theta_2}{\sin\theta_1}\right)\right]$$

Where H_k is the anisotropy field, which is zero for a point spin. Now let us compare this to our theoretic results.

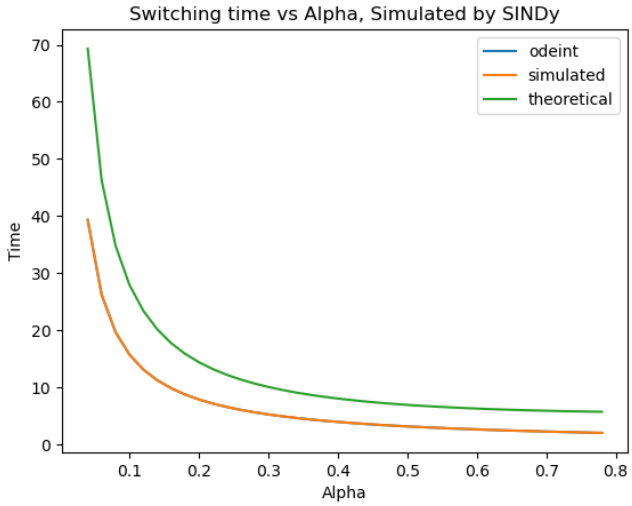


Figure 19: Switching times

3.3.3 Errors

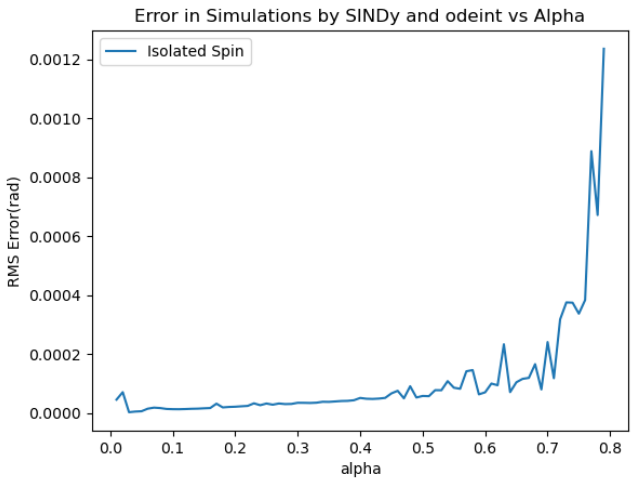


Figure 20: Errors in SINDy vs alpha

We see that the errors are acceptable over a wider range of α in the 2D case, when compared to the 3D case.

4 Pulse Programming

- We have demonstrated the robustness of SINDy for identifying the dynamics of a non-linear system such as the LLG equation.
- We now look to possible avenues where this might be of use. One such avenue is pulse programming.
- Given that we can control pulses in the z and now the x-direction, our aim is to identify the optimal pulse lengths, timings and amplitudes to manipulate the spin so that it reaches the desired state in the least amount of time.
- More specifically, we can start with finding the optimal pulses to reduce the switching time.

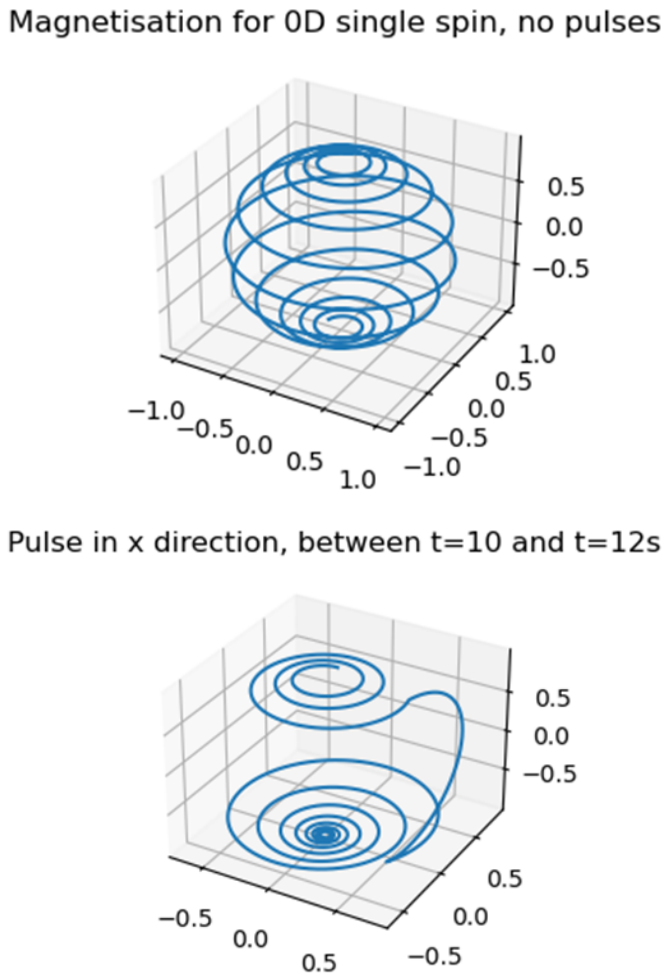


Figure 21: Effect of x-Pulse on switching times

The Switching time was shortened from τ=39.33s to τ =25.81s. Notice how the spin took a shorter path along the Bloch sphere to reach the -ve z-axis quicker.

4.1 Optimizing Pulses

As a first step, we try to optimize the parameters separately.

4.1.1 Optimizing time to start pulse, keeping pulse length constant

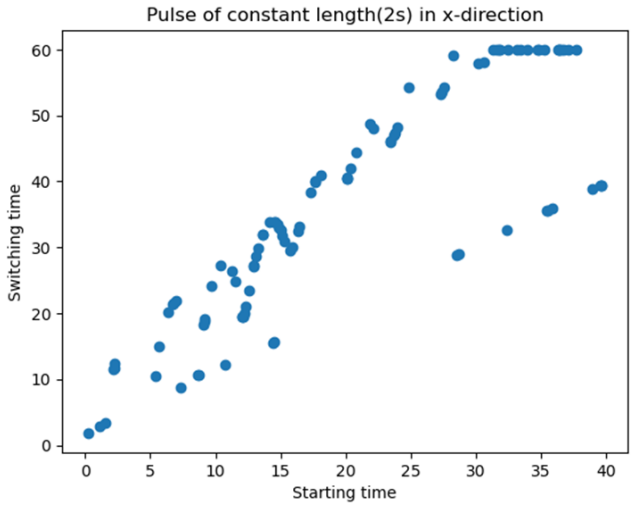


Figure 22: Switching time vs pulse starting time

We find the optimal time to be t0=0.19s.

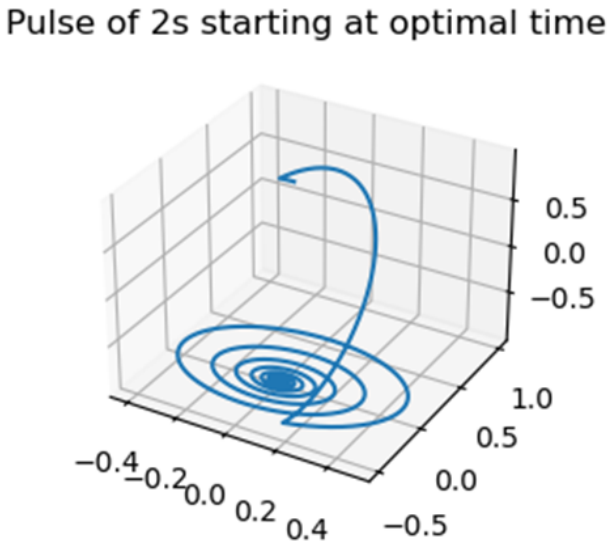


Figure 23: Evolution at optimal pulse time=0.19s

4.1.2 Optimizing pulse length

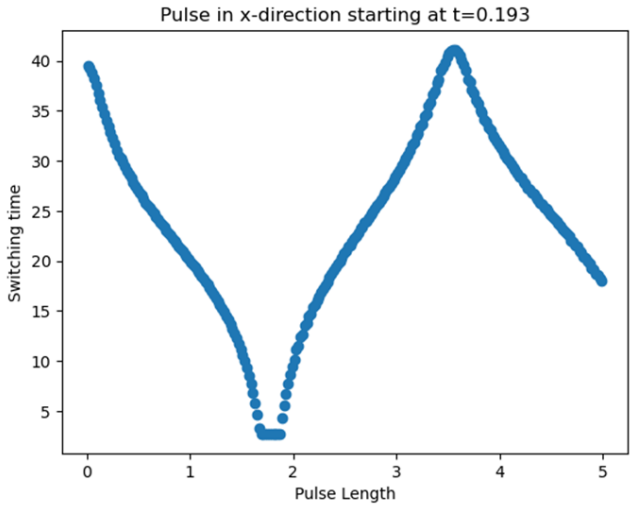


Figure 24: Switching time vs pulse length

We find the optimal time to be t=1.81s. Note,if the spin goes to 170°and comes back, before settling down, this behaviour is undesirable, we can account for this by redefining the switching time function, by requiring that switching is done when 25 consecutive values of θ are at $\geq 170°$.

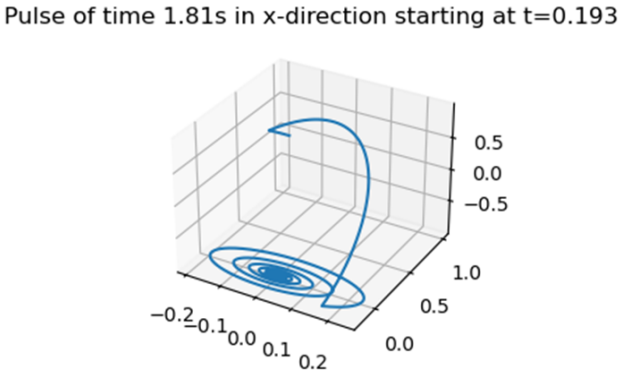


Figure 25: Evolution at optimal pulse time=0.19s, and length=1.81s

5 Next Steps

We have successfully demonstrated the use of SINDy for extracting the dynamical equations of the LLG equation using the time-series data for a single point system under a wide range of conditions. But, we have seen that for some extreme cases of high damping, SINDy is unable to identify the model successfully, and need very specific hyperparameters and optimizers to identify the equation. I have tweaked these hyperparameters as a function of damping by observing general trends in the success of the algoithm to obtain a more robust algorithm that can identify the dynamics of the LLG equation better, and even use various hyperparamters to come up with smaller errors with being too computationally demanding. I have also converted the timeseries data to polar coordinates, since the radial component is normalized for the LLG equation, we have just 2 components that change with time. I have shown the LLG identifies the 2D- LLG equation in polar coordinates [4] using this coordinate transformation, for a point isolated charge, and I am working on extending it to other geometries and adding anisotropy fields to see if we can still identify the dynamics successfully. A next step might also be to increase the dimensionality and provide SINDy with a spatial grid of points that all interact with each other to extract the dominant PDE. Here, the underlying equation(LLG) would be the same, but the data provided would be of a much higher dimension, and the task would be more challenging.
Further, I have started looking into Pulse programming and I have taken the first step, that is to find the optimal pulse parameters separately. The parameters can be determined in a similar manner by brute force, but this is a very computationally intensive task. I am exploring some quantum algorithms to see if it can be done more efficiently. The final goal would be to optimize pulses from time-series data to replace logical quantum gates with simpler operations to simplify quantum circuits. Machine learning can help us with these kinds of tasks without replying on intractable exhaustive algorithms.
Also, we can add noise to the data and see how SINDy, along with denoising techniques, deals with it, and whether the equations are still able to be identified.

References

[1] Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, mar 2016.

[2] V V Kruglyak, S O Demokritov, and D Grundler. Magnonics. *Journal of Physics D: Applied Physics*, 43(26):260301, jul 2010.

[3] A. Smith, K. K. Nielsen, D. V. Christensen, C. R. H. Bahl, R. Bjørk, and J. Hattel. The demagnetizing field of a nonuniform rectangular prism. *Journal of Applied Physics*, 107(10):103910, 05 2010.

[4] J.C. Mallinson. Damped gyromagnetic switching. *IEEE Transactions on Magnetics*, 36(4):1976–1981, 2000.