Physics Informed Machine Learning in Micromagnetism

Sebastian Schaffer
Advisors: Norbert J. Mauser, Lukas Exl

Research Platform MMM Mathematics - Magnetism - Materials, Univ. of Vienna
Wolfgang Pauli Institute (WPI) c/o Fak. Math, Vienna

30.11.2022

Topics

- Physics Informed Machine Learning for Micromagnetism
- Physics Informed Neural Networks
- Stray Field Problem
- Micromagnetic Energy Minimization

Advisors

- Supervised by Prof. Norbert J. Mauser and co-supervised by Lukas Exl at the Wolfgang Pauli Institute (WPI) and the Research Platform MMM Mathematics - Magnetism - Materials at University of Vienna.
- Austrian Science Fund (FWF) project "Reduced Order Approaches for Micromagnetics (ROAM)" (PI L. Exl) and FWF project "Design of Nanocomposite Magnets by Machine Learning (DeNaMML)" at MMM c/o Univ. Wien (National research partner: L. Exl).

Stray field problem

We want to solve the stray field problem using machine learning

$$\begin{aligned}
-\Delta\phi &= -\nabla\cdot m \quad \text{in } \Omega\subset\mathbb{R}^3,\\
-\Delta\phi &= 0 \quad \text{in } \overline{\Omega}^c,\\
[\phi] &= 0 \quad \text{in } \partial\Omega,\\
\left[\frac{\partial\phi}{\partial n}\right] &= -m\cdot n \quad \text{in } \partial\Omega,\\
\phi(x) &= \mathcal{O}\left(\frac{1}{\|x\|}\right) \quad \text{as } \|x\|\to\infty.
\end{aligned} \tag{1}$$

The solution is given by

$$\phi(x) = -\frac{1}{4\pi}\left(\int_\Omega \frac{\nabla\cdot m(y)}{\|x-y\|}dy - \int_{\partial\Omega}\frac{m(y)\cdot n(y)}{\|x-y\|}d\sigma(y)\right). \tag{2}$$

Ansatz of García-Cervera and Roma

Split the potential $\phi = \phi_1 + \phi_2$ and solve the following problems [1]:

$$\begin{aligned} -\Delta\phi_1 &= -\nabla \cdot m \quad \text{in } \Omega, \\ \phi_1 &= 0 \quad \text{on } \partial\Omega, \end{aligned} \tag{3}$$

and

$$-\Delta\phi_2 = 0 \quad \text{in } \Omega,$$

with the boundary values computed by the single layer potential

$$\phi_2^*(x) = \int_{\partial\Omega} \left(\frac{(m \cdot n - \frac{\partial\phi_1}{\partial n})(y)}{4\pi\|x - y\|} \right) d\sigma(y). \tag{4}$$

Neural Network

$$\begin{aligned} a^{(0)} &= \sigma^{(0)}(x) \\ a^{(t)} &= \sigma^{(t)}(W^{(t)}a^{(t-1)} + b^{(t)}) \quad \text{for } t = 1, \dots, T \end{aligned} \tag{5}$$

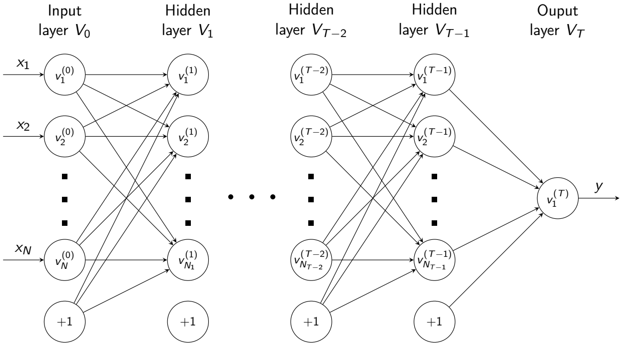

Figure 1: A multilayer perceptron with N inputs and scalar output and $T-1$ hidden layers.

Why Physics Informed Machine Learning?

Can we build a model which can predict the solution of this problem for all parameter ranges?

$$\frac{d^2u(x)}{dx^2} = 1 - 2x^2 \tag{6}$$

with mixed boundary conditions

$$\alpha_l u(x) + \beta_l \frac{d^2u(x)}{dx^2} = \gamma_l \quad \text{at } x = 0,$$
$$\alpha_h u(x) + \beta_h \frac{d^2u(x)}{dx^2} = \gamma_h \quad \text{at } x = 1,$$

for $x \in [0, 1]$, $\alpha_l \in [0.8, 1]$, $\beta_l \in [-1, 0]$, $\gamma_l \in [0, 1]$, $\alpha_h \in [0.8, 1]$, $\beta_h \in [0, 1]$ and $\gamma_h \in [0, 1]$

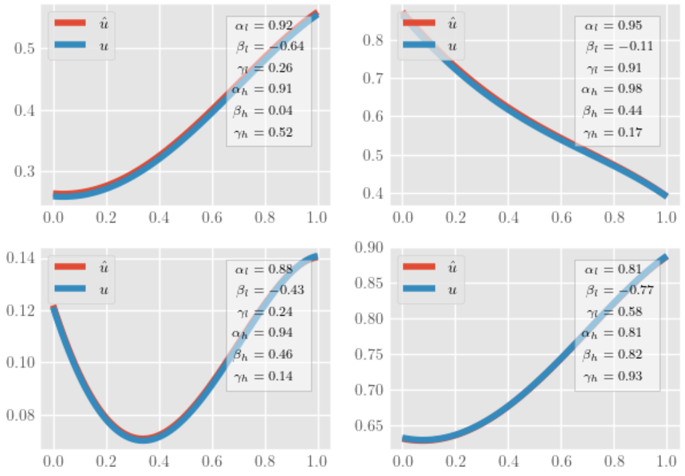

Figure 2: Solution to problem (6) using a single PINN with only 2131 parameters.

Why Physics Informed Machine Learning?

Table 1: Different NN models for problem (6) for different input dimensions together with the required number of model parameters. All models had one hidden layer with a hyperbolic tangent activation function.

dimension	#parameters	d.o.f. FEM	MAE PINN
1	9 (3)	$\mathcal{O}(10^1)$	2.75e-4
2	12 (3)	$\mathcal{O}(10^2)$	6.74e-4
3	30 (6)	$\mathcal{O}(10^3)$	3.92e-4
4	48 (8)	$\mathcal{O}(10^4)$	1.74e-3
5	63 (9)	$\mathcal{O}(10^5)$	2.27e-3
6	88 (11)	$\mathcal{O}(10^6)$	3.12e-3

Here the # of PINN parameters grows about linearly only.

PINNs

Given the problem

$$\begin{aligned} -\Delta u = f \qquad & \text{in } \Omega \\ u = u^* \qquad & \text{on } \partial\Omega \end{aligned}$$

we can use a neural network $\hat{u}$ as ansatz function. For N sampling points $\{x_1, \dots, x_N\} \in \Omega$ and M points $\{y_1, \dots, y_M\} \in \partial\Omega$ we can write the objective as

$$\sum_{i=1}^{N} \left(\Delta \hat{u}(x_i) + f(x_i)\right)^2 + \alpha \sum_{j=1}^{M} \left(\hat{u}(y_j) - u^*(y_j)\right)^2 .$$

This can be solved using SGD or other optimization techniques.

PINN stray field solution

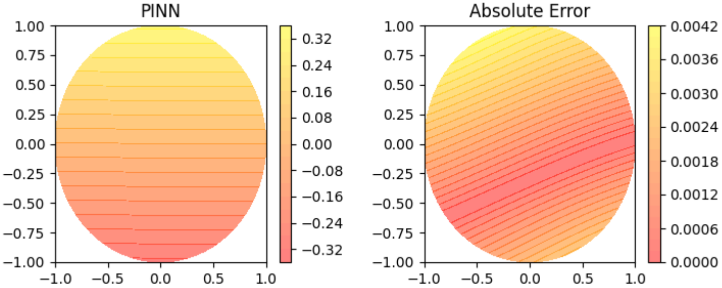

Figure 3: PINN solution of stray field problem in 3d uniformly magnetized sphere.

Hard Constraints

We can use the following penalty free ansatz [2] to remove the penalty term

$$\hat{u}(x) = \ell(x)\hat{u}(x) + g(x),$$
$$\ell(x) = \begin{cases} 0 & \text{if} \quad x \in \partial\Omega \\ > 0 & \text{else.} \end{cases} \tag{7}$$

We can learn the parameters b for the boundary data and de-couple model and constraints

$$b^* = \arg\min_{b \in \mathbb{R}^B} \sum_{i=1}^{M} \|g(y_i) - u^*\|^2.$$

Weak Formulation

Instead of solving the Poisson/Laplace equation directly, we can use the Ritz–Galerkin formulation and minimize the functional

$$\mathcal{I}(u) = \int_\Omega \left(\frac{1}{2} |\nabla u(x)|^2 - f(x)u(x) \right) dx,$$

using a penalty framework, this leads to the Deep Ritz method [3]:

$$\sum_{i=1}^{N} \left(\frac{1}{2} |\nabla \hat{u}(x_i)|^2 - f(x_i)\hat{u}(x_i) \right) + \alpha \sum_{j=1}^{M} \left(\hat{u}(y_j) - u^*(y_j) \right)^2 .$$

Weak Formulation

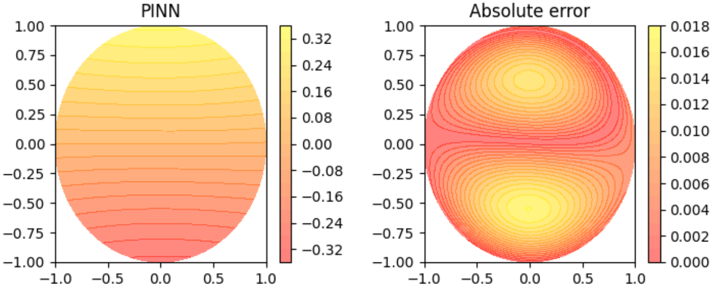

Figure 4: PINN solution of stray field problem using the Deep Ritz method in 3d uniformly magnetized sphere.

Pros and Cons

− Using soft constraints often yield very complicated stiff optimization problems [4].

− Soft constraints require an additional penalty parameter.

\+ The weak formulation can reduce the required smoothness of the model.

\+ Parameters can be integrated as additional inputs to the model to learn a whole family of solutions.

Extreme Learning Machine

$$y = a^T \sigma(Wx + b) \tag{8}$$

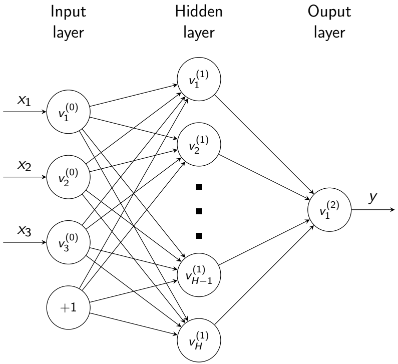

Figure 5: An extreme learning machine with 3-dimensional input x, H hidden nodes and scalar output y where only a is trainable [5].

Unconstrained Quadratic Optimization

Alternatively, we can use the following penalty free ansatz [2] to turn the problem into an unconstrained quadratic program

$$\hat{u}(x) = \ell(x) \sum_{i=1}^{D} a_i k_i(x) + \sum_{j=1}^{B} b_j g_j(x),$$
$$\ell(x) = \begin{cases} 0 & \text{if} \quad x \in \partial\Omega \\ > 0 & \text{else.} \end{cases} \tag{9}$$

We can learn the parameters b for the boundary data and de-couple model and constraints

$$b^* = \arg\min_{b \in \mathbb{R}^B} \sum_{i=1}^{M} \|b^T g(y_i) - u^*\|^2.$$

Unconstrained Quadratic Optimization

With the solution b^* we can solve the following unconstrained quadratic program:

$$\mathcal{I}_Q(a) = a^T Q a - b^T a, \tag{10}$$

with $Q \in \mathbb{R}^{D \times D}$, $b \in \mathbb{R}^D$ and

$$\begin{aligned} Q_{ij} &= \frac{1}{2} \int_\Omega \nabla(\ell(x) k_i(x))^T \nabla(\ell(x) k_j(x)) dx, \\ b_i &= \int_\Omega f(x) \ell(x) k_i(x) + \nabla(\ell(x) k_i(x))^T \nabla({b^*}^T g(x)) dx. \end{aligned}$$

The indicator function ℓ can be learned with spline interpolation [2] and instead of a neural network we can use an Extreme Learning Machine (ELM).

Unconstrained Quadratic Optimization

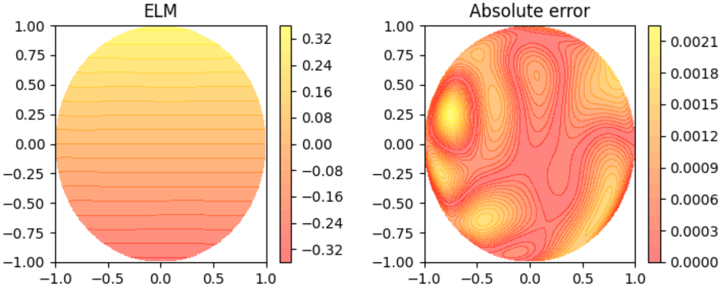

Figure 6: ELM solution of stray field problem in 3d uniformly magnetized sphere using an unconstrained quadratic programming approach.

Unconstrained Quadratic Optimization

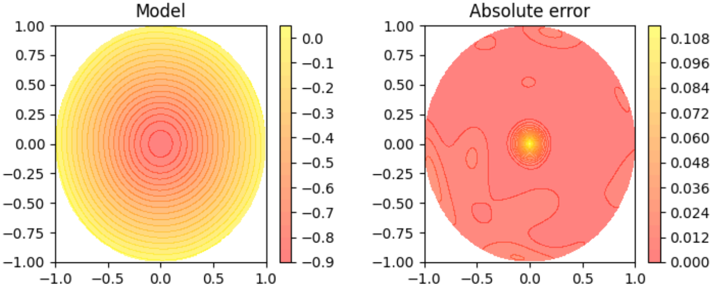

Figure 7: ELM solution of stray field problem in 3d outward magnetized sphere using an unconstrained quadratic programming approach.

Flower State

Table 2: Comparison of PINN and FEM for the flower state in a unit cube w.r.t. DM method with $40 \times 40 \times 40$ grid [6].

	FEM ($N = 7.2e04$)	PINN
Error in Energy	2.0e-04	2.0e-04

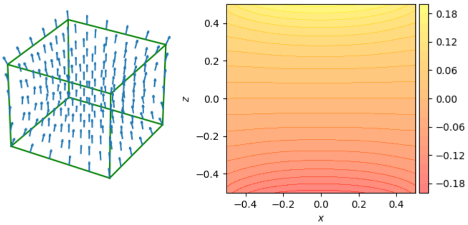

Figure 8: ELM solution of stray field problem in 3d cube with flower state using an unconstrained quadratic programming approach. **The model only had** 3×256 **trainable parameters.**

Vortex

Table 3: Comparison of PINN and FEM for the vortex state in a unit cube w.r.t. DM method with $40 \times 40 \times 40$ grid [6].

	FEM ($N = 7.2e04$)	PINN
Error in Energy	2.9e-04	7.2e-04

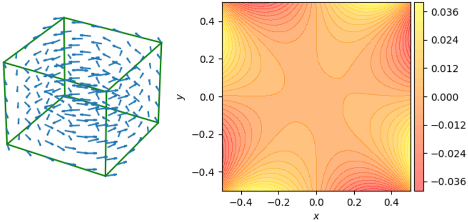

Figure 9: ELM solution of stray field problem in 3d cube with vortex magnetization using an unconstrained quadratic programming approach. **The model only had** 3×256 **trainable parameters.**

Uniform magnetized plate

Table 4: Comparison of PINN and FEM for the uniform state in a plate wrt. DM method with $40 \times 40 \times 8$ grid [6].

	s	FEM ($N = 4.9e04$)	PINN
Error in Energy	1.0	8.6e-04	7.6e-04
	0.1	4.2e-04	3.3e-04

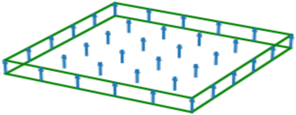

Figure 10: Uniformly magnetized plate with thickness s.**The model only had** 3×256 **trainable parameters.**

Advantage over conventional methods

- We can learn the solution to a whole family of functions, because neural networks can approximate functions in very high dimensional spaces, breaking the curse of dimensionality.
- There are less parameters (d.o.f.) required to achieve satisfactory accuracy compared to conventional mesh-based methods.
- PINNs are inherently mesh free methods.
- There is no (additional) interpolation required and the derivative can be evaluated at every point.
- ELM are fast and easy to train and require little hyper-parameter adjustment. By using hard constraints we can achieve similar accuracy as conventional methods.

Energy Minimization

The magnetostatic energy $e_m = -\frac{1}{2}\int_\Omega m \cdot h \, dx$ can bound by

$$e_m \geq -\|h\|^2_{L^2(\mathbb{R}^3,\mathbb{R}^3)} - 2(h,m)_{L^2(\Omega,\mathbb{R}^3)}, \tag{11}$$

$$e_m \leq \|b\|^2_{L^2(\mathbb{R}^3,\mathbb{R}^3)} - 2(b,m)_{L^2(\Omega,\mathbb{R}^3)}, \tag{12}$$

with $m = b - h$ [7].

Energy Minimization

By using a NN ansatz for h and m, we can minimize the functional

$$L = L_{mag} + L_{zee} + L_{ani} + L_{ex}, \tag{13}$$

with

$$\begin{aligned}
L_{mag} &= \|b\|^2_{L^2(\mathbb{R}^3,\mathbb{R}^3)} - 2(b, m)_{L^2(\Omega,\mathbb{R}^3)}, \\
L_{zee} &= -(m, h_{ext})_{L^2(\Omega,\mathbb{R}^3)}, \\
L_{ani} &= -(K_1 m_z^2)^2_{L^2(\Omega)} + (K_2 m_z^2)^4_{L^2(\Omega)}, \\
L_{ex} &= \|A_{ex} \nabla m\|^2_{L^2(\Omega,\mathbb{R}^{3\times 3})},
\end{aligned}$$

where m_z is the component of m aligned with the easy axis and $A_{ex} \in \mathbb{R}^{3\times 3}$ is the matrix of exchange coefficients.

Energy Minimization

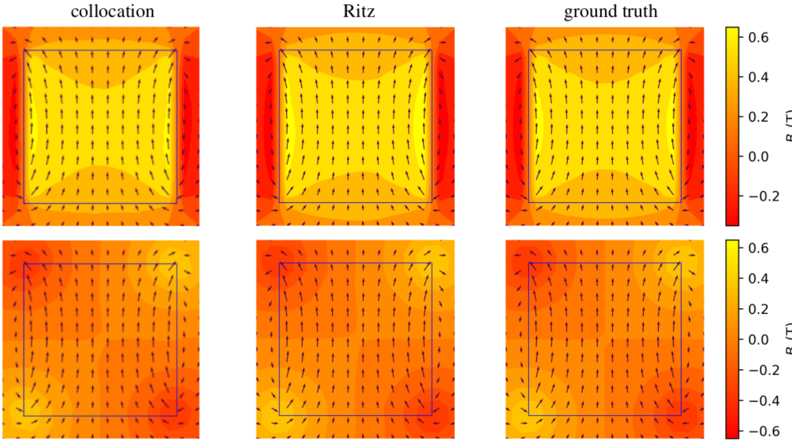

Figure 11: Comparison of collocation method and energy minimization for uniform magnetized domain in 2d [8].

NIST μMAG Standard Problem 3

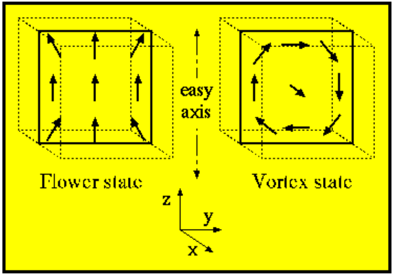

Figure 12: μMAG Standard Problem 3.

Conclusion and Outlook

- We want to establish models for non-uniform magnetization on complex geometries.
- Domain decomposition and parallel computing can be used to scale PINNs to larger domains.
- Applying PINNs in full micromagnetic setups e.g. hysteresis computation.
- Parameter dependent PINNs can learn a whole family of solutions. This can e.g. be used as surrogate model for inverse design.

Bibliography I

[1] Lukas Exl and Thomas Schrefl. "Non-uniform FFT for the finite element computation of the micromagnetic scalar potential". In: *Journal of Computational Physics* 270 (2014), pp. 490–505.

[2] Hailong Sheng and Chao Yang. "PFNN: A penalty-free neural network method for solving a class of second-order boundary-value problems on complex geometries". In: *Journal of Computational Physics* 428 (2021), p. 110085.

[3] Bing Yu et al. "The deep Ritz method: a deep learning-based numerical algorithm for solving variational problems". In: *Communications in Mathematics and Statistics* 6.1 (2018), pp. 1–12.

[4] Sifan Wang, Yujun Teng, and Paris Perdikaris. "Understanding and mitigating gradient flow pathologies in physics-informed neural networks". In: *SIAM Journal on Scientific Computing* 43.5 (2021), A3055–A3081.

[5] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. "Extreme learning machine: theory and applications". In: *Neurocomputing* 70.1-3 (2006), pp. 489–501.

[6] Claas Abert, Lukas Exl, Gunnar Selke, et al. "Numerical methods for the stray-field calculation: A comparison of recently developed algorithms". In: *Journal of Magnetism and Magnetic Materials* 326 (2013), pp. 176–185.

Bibliography II

[7] Lukas Exl. "A magnetostatic energy formula arising from the L2-orthogonal decomposition of the stray field". In: *Journal of Mathematical Analysis and Applications* 467.1 (2018), pp. 230–237.

[8] Alexander Kovacs, Lukas Exl, Alexander Kornell, et al. "Magnetostatics and micromagnetics with physics informed neural networks". In: *Journal of Magnetism and Magnetic Materials* 548 (2022), p. 168951.