Solar Physics (2024) 299:108
https://doi.org/10.1007/s11207-024-02352-5

RESEARCH

Machine-Learning-Based Numerical Solution for Low and Lou's Nonlinear Force-Free Field Equilibria

Yao Zhang[1,2] · **Long Xu**[3] · **Yihua Yan**[1,2]

Received: 11 February 2023 / Accepted: 19 July 2024 / Published online: 9 August 2024

Abstract
Low and Lou (*Astrophys. J.* **352**, 343, 1990) presented a family of nonlinear force-free magnetic fields that have established themselves as the gold standard for extrapolating force-free magnetic fields in solar physics. Building upon this important work, our study introduces a novel grid-free machine-learning-based method to effectively solve the equilibria proposed by Low and Lou. Through extensive numerical experiments, our results unequivocally demonstrate the efficient capability of the machine-learning algorithm in deriving numerical solutions for Low and Lou's equilibria. Furthermore, we explore the opportunities and challenges of applying artificial-intelligence technology to real observed solar active regions.

Keywords Nonlinear · Force free · Magnetic fields · Machine learning

1. Introduction

Knowledge of the magnetic-field structure is significant for understanding solar phenomena on the Sun, for example coronal mass ejections, flares, and filaments. However, it is difficult to measure the solar magnetic field with high accuracy except in the photosphere. Therefore, there are several models of magnetic-field extrapolation that are proposed, such as potential fields (PF) (Schmidt 1964; Altschuler and Newkirk 1969), linear force-free fields (LFFF) (Nakagawa and Raadu 1972; Chiu and Hilton 1977), and nonlinear force-free fields (NLFFF) (Sakurai 1989; Neukirch 2005; Wiegelmann and Sakurai 2021).

✉ L. Xu
lxu@nao.cas.cn

Y. Zhang
jaafar_zhang@163.com

Y. Yan
yyh@nao.cas.cn

1 State Key Laboratory of Space Weather, National Space Science Center, Chinese Academy of Sciences, Beijing, China

2 University of Chinese Academy of Sciences, Beijing, China

3 Ningbo University, Ningbo, Zhejiang, China

The PF model is charaterized by

$$\begin{aligned} \Delta\phi &= 0 \ \text{ in } \Omega \\ \frac{\partial\phi}{\partial n} &= B_{z_0} \ \text{ on } \partial\Omega \end{aligned}, \tag{1}$$

where $\boldsymbol{B} = \boldsymbol{B}(x, y, z) = \left(B_x, B_y, B_z\right)$ is the field, Ω is the open space above the solar surface, and $\partial\Omega$ is its boundary. Finally, the magnetic field can be formulated as $\boldsymbol{B} = \nabla\phi$.

The solution of Equation 1 is unique. The solution is a good approximation of the large-scale topology of a solar active region, but it is not suitable for a small-scale topology such as that discussed by Aulanier, Pariat, and Démoulin (2005).

The force-free field model can be written as

$$\begin{aligned} \nabla \times \boldsymbol{B} &= \alpha \boldsymbol{B} \ \text{ in } \Omega \\ \nabla \cdot \boldsymbol{B} &= 0 \ \text{ in } \Omega \\ \boldsymbol{B} &= \boldsymbol{B_0} \ \text{ on } \partial\Omega \end{aligned}, \tag{2}$$

where $\boldsymbol{B_0}$ is the observed magnetic field in the photosphere. When α is a constant, Equation 2 represents the LFFF model, which can be solved analytically using the Green's function method or the Fourier method, as discussed by Wiegelmann and Sakurai (2021). In particular, when $\alpha = 0$, Equation 2 describes the PF model, which can also be solved using these analytical methods.

In the LFFF model, α is a global free parameter. As observed by Démoulin et al. (2002) and Valori et al. (2010), a large α may lead to overestimation, while a small α may lead to under-estimation. When the assumption of a constant α is relaxed, Equation 2 becomes the NLFFF model, where α is a spatially dependent scalar function. With a variable α, a closed-form solution to Equation 2 is no longer feasible. This has prompted the development of various numerical algorithms to solve Equation 2. These numerical algorithms include the works by Grad and Rubin (1958), Nakagawa (1974), Mikić and McClymont (1994), Amari et al. (1997), Wheatland, Sturrock, and Roumeliotis (2000), Yan and Sakurai (2000), Régnier, Amari, and Kersalé (2002), Wiegelmann and Neukirch (2003), Wiegelmann, Inhester, and Sakurai (2006), and Yan and Li (2006).

However, without ground-truth 3D magnetic fields, the performance, stability, and accuracy of these algorithms cannot be objectively evaluated. Fortunately, assuming an axially symmetric configuration of the magnetic field, i.e. $\frac{\partial}{\partial\phi} = 0$ in the spherical coordinate system (r, θ, ϕ), Low and Lou provided a set of separable and semi-analytical solutions to Equation 2:

$$\boldsymbol{B} = \frac{B_0}{R^{n+2}} \left[-\frac{\mathrm{d}F}{\mathrm{d}\mu}\widehat{\boldsymbol{r}} + \frac{nF}{\left(1-\mu^2\right)^{\frac{1}{2}}}\widehat{\boldsymbol{\theta}} + \frac{\eta A}{\left(1-\mu^2\right)^{\frac{1}{2}}}\widehat{\boldsymbol{\phi}} \right], \tag{3}$$

where

$$A = \begin{cases} F^{1+\frac{1}{n}}, & n = 1 \\ F|F|^{\frac{1}{n}}, & n \geqslant 3 \end{cases} \tag{4}$$

and F satisfies

$$\left(1-\mu^2\right)\frac{\mathrm{d}^2F}{\mathrm{d}\mu^2} + n(n+1)F + C = 0, \tag{5}$$

where

$$C = \begin{cases} \left(1+\frac{1}{n}\right) a^2 F^{1+\frac{2}{n}}, & n = 1 \\ \left(1+\frac{1}{n}\right) a^2 F\left(F^2\right)^{\frac{1}{n}}, & n \geqslant 3. \end{cases} \tag{6}$$

For a comprehensive and in-depth analysis of the mathematical framework of Low and Lou's equilibria, please consult Appendix A for detailed discussions.

In fact, $n = 1$ and $n \geq 3$ can be generalized, as $\frac{2}{n}$ is positive and not necessarily an integer in both Equations 4 and 6. For example, $n = \frac{4}{3}$, i.e. $\frac{2}{n} = \frac{3}{2}$ is not a positive integer, but $n = \frac{2}{9}$, i.e. $\frac{2}{n} = 9$ is a positive integer.

It is to be observed that $F^{1+\frac{2}{n}}$ is not always equal to $F\left(F^2\right)^{\frac{1}{n}}$. For example, $F^{\left(1+\frac{2}{n}\right)} > 0$, but $F\left(F^2\right)^{\frac{1}{n}} < 0$ when $F = -0.1$, $n = \frac{2}{9}$. In this article, we take $n = 5$, 3, 1.5, 1, 0.9, 0.7, 0.5, 0.3, and 0.1, respectively.

Equation 5 is a second-order, nonlinear, ordinary differential equation (ODE). Solving ODEs is an important topic in mathematics and engineering. In general, most of the existing research for finding solutions to ODEs falls into two main categories: analytical techniques and numerical methods. The analytical techniques include, e.g., separation of variables and the method of integrating factors. The numerical methods include, e.g., Euler's method and the Runge–Kutta (RK) method. The numerical method is usually expressed in terms of the discretization parameters. Artificial-intelligence-based methods are also currently being used in solving ODEs, such as that discussed by Raissi, Perdikaris, and Karniadakis (2019), Dufera (2021), and Cuomo et al. (2022).

Several works have extended Low and Lou's equilibria in the last 30 years, such as Low and Flyer (2007), Lerche and Low (2014), and Prasad, Mangalam, and Ravindra (2014) redefined Equations 6, 7, and 9 of Low and Lou (1990) in different ways. To the best of our knowledge, there is no discussion about the existence and uniqueness of analytical solutions to Equation 5. The most commonly used numerical method to solve Equation 5 is the RK fourth-order method (RK4). RK4 provides the approximate value of $F(\mu_i)$ at the discrete sampling points μ_is. If μ_j is not included in the set of μ_i, we will not be able to directly determine the value of $F\left(\mu_j\right)$. In this article, a neural network can compute $F\left(\mu_j\right)$ and $F'\left(\mu_j\right)$ at any μ_j in $[-1, 1]$ directly.

The remainder of this article is organized as follows: Section 2 provides a RK-based method for solving for the parameter a in Equation 6. Section 3 develops the corresponding numerical algorithm. The data-driven numerical method for the parameters n and a is presented in Section 4. The conclusion is given in Section 5.

2. RK-Based Method for the Parameter a in Equation 5

Returning to Equations 3 and 5, there is one parameter a and two unknowns $\left(F, \frac{\mathrm{d}F}{\mathrm{d}\mu}\right)$ with $B_0 = 1$ and $r_0 = 1$ in Low and Lou's equilibria. In this section, we study numerical methods for solving for a.

The basic idea for solving Equation 5 with initial conditions is to rewrite it as a system of first-order ODEs. Introducing the variables

$$f_1 = F(\mu) \quad \text{and} \quad f_2 = F'(\mu),$$

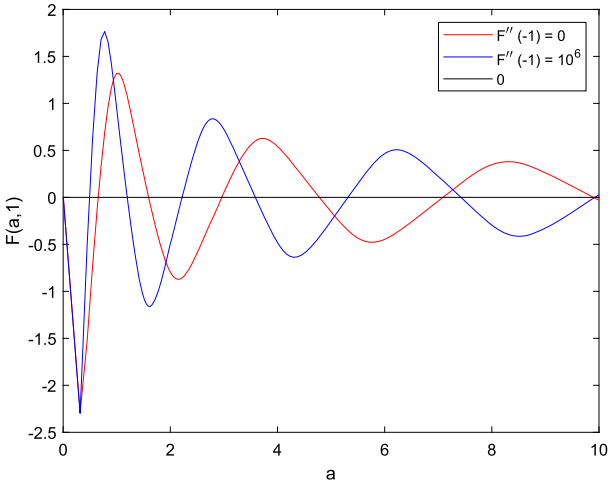


Figure 1 The curves of $F(a,1)$ with the variable a for $F''(-1)=0$ and $F''(-1)=10^6$, where the points of intersection between $F(a,1)$ and the horizontal axis can determine the values of a for $F(a,1)=0$.

we obtain a system of two first-order ODEs:

$$\begin{cases} f_1' = f_2, \\ f_2' = -\frac{n(n+1)f_1+D}{1-\mu^2}, \end{cases}$$

where

$$D = \begin{cases} \left(1+\frac{1}{n}\right)a^2 f_1^{1+\frac{2}{n}}, & \text{if } \frac{2}{n} \text{ is an integer,} \\ \left(1+\frac{1}{n}\right)a^2 f_1 \left(f_1^2\right)^{\frac{1}{n}}, & \text{otherwise,} \end{cases}$$

with initial values at $\mu=-1$

$$f_1(-1)=0 \quad \text{and} \quad f_2(-1)=10.$$

The RK method is an effective method for solving the initial-value problem of Equation 5 with an unknown parameter a if $f_1(1)=0$.

We divide the interval $[a_0, a_{\max}]$ into N equal parts, choosing a step $h=\frac{a_{\max}-a_0}{N}$. Then, $a_i=a_0+(i-1)h\,(1\leqslant i\leqslant N)$. Without loss of generality, we choose $a_0=10^{-5}$, $a_{\max}=10$, $N=999$.

Note that $\mu=-1$ is a singular point of f_2' in Equation 5. The value of $f_2'(-1)$ depends on the parameter a, as shown Figure 1, for $n=1$, $f_2'(-1)=0$, and 10^6.

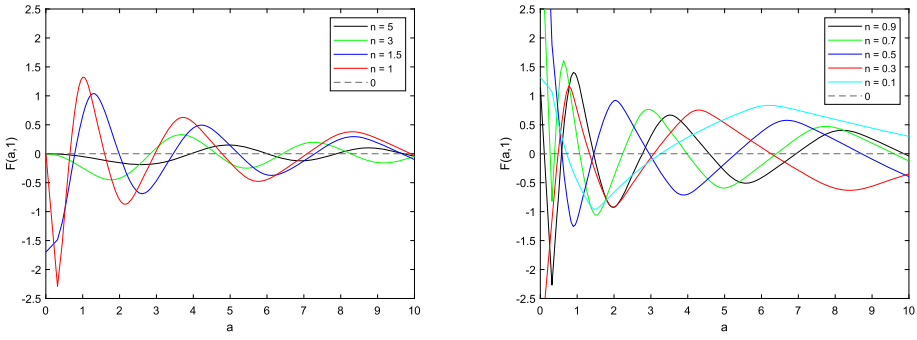
We take $f_2'(-1)=0$ in this section. It helps us to find the interval that contains zeros when we plot $F(a_i,\mu=1)$ against a, as shown in Figure 2. Finally, we find $F(a_i,\mu=1)=0$ by the bisection method, and we find $a_{n,m}$s $(1\leq m\leq 3)$ as presented in Table 1.

3. Machine-Learning-Based Method for F and $\frac{\mathbf{d}F}{\mathbf{d}\mu}$

Most of the classical numerical methods compute an approximate value for the solution at discrete sampling points. In this section, we propose a grid-free method based on a neural network to obtain a numerical solution at any point in $[-1,1]$.

Table 1 $a_{n,m}$ for the mth-order root of $F(a_i, \mu = 1)$ for a given n.

m	n								
	5	3	1.5	1	0.9	0.7	0.5	0.3	0.1
1	3.9341	2.8348	0.7846	0.6538	0.0719	0.0589	0.5865	0.4819	0.7162
2	6.0052	4.6243	1.9865	1.5947	0.5760	1.0678	1.4512	1.3332	3.2644
3	7.8851	6.4147	3.4555	2.9597	1.4474	2.2332	2.9533	3.0769	12.4581



(a) $n \geq 1$ (b) $n < 1$

Figure 2 $F(a, 1)$ with the variable a for different n.

The following theorem provides a solid theoretical basis for using multi-layer perceptrons (MLP) in scientific computing.

Theorem 1 (Universal approximation theorem (Cybenko 1989 and Hornik 1991)) *Let* $K \subseteq \mathbb{R}^d$ *be compact,* $f : K \to \mathbb{R}$ *be continuous,* $\rho : \mathbb{R} \to \mathbb{R}$ *be continuous and not a polynomial. Then, for* $\forall \epsilon > 0$, *there exists* $N \in \mathbb{N}$, $a_k, b_k \in \mathbb{R}$, $w_k \in \mathbb{R}^d$ *with*

$$\left\| f - \sum_{k=1}^{N} a_k \rho\left(\langle w_k, \cdot \rangle - b_k\right) \right\|_{\infty} < \epsilon.$$

Remark 1 The universal approximation theorem also exists when $f : K \to \mathbb{R}^m$.

Therefore, we use an MLP architecture with three fully connected operations to solve Equation 5 as shown in Figure 3. The first fully connected operation has one input channel corresponding to the inputs μ. The second fully connected operation has N hidden neurons. The last fully connected operation has two outputs F and $\frac{\mathrm{d}F}{\mathrm{d}\mu}$. It can be formulated as

$$\boldsymbol{f}(\mu; \theta) = [f_1, f_2]^T = W_2 Y + b_2 = W_2 [\sigma(W_1 \mu + b_1)] + b_2, \tag{7}$$

where $W_1, b_1 \in \mathbb{R}^{N \times 1}$, $W_2 \in \mathbb{R}^{2 \times N}$, $b_2 \in \mathbb{R}^{2 \times 1}$, $[\cdot]^T$ is the transpose operator in linear algebra, $Y = [y_1, \ldots, y_N]^T$, and σ is the hyperbolic tangent sigmoid (tanh) elementwise operator

$$\tanh(x) = \frac{1 - \mathrm{e}^{-x}}{1 + \mathrm{e}^{-x}}, \tag{8}$$

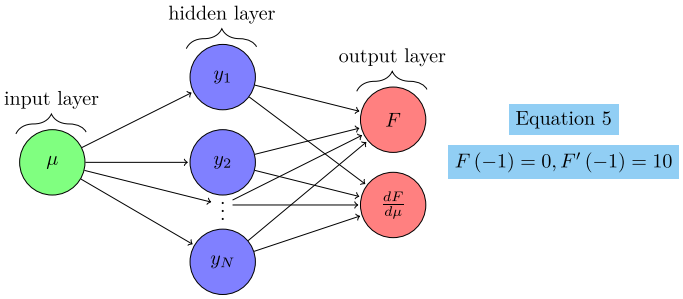


Figure 3 The MLP has an input layer, one hidden layer, and an output layer. A layer consists of small individual units called neurons. The letter N is used for the number of neurons in the hidden layer. The loss function contains information from the differential equation and the initial conditions.

where x is an element of the column vector $W_1\mu + b_1$.

Theorem 1 ensures the existence of a neural-network solution $\boldsymbol{f}(\mu;\theta)$ that can approximate the solution of Equation 5 arbitrarily closely. Then, we find $\boldsymbol{f}(\mu;\theta)$ by minimizing the loss function

$$\mathcal{L}(\mu;\theta) = \lambda_1\mathcal{L}_1(\mu;\theta) + \lambda_2\mathcal{L}_2(\mu;\theta) + \lambda_3\mathcal{L}_3(\mu;\theta), \tag{9}$$

where

$$\mathcal{L}_1(\mu;\theta) = \frac{1}{M}\sum_{\mu=\mu_j,j=1}^{j=M}\left(\left(1-\mu^2\right)\frac{\mathrm{d}f_2}{\mathrm{d}\mu} + n(n+1)f_1 + E\right)^2$$

$$\mathcal{L}_2(\mu;\theta) = \frac{1}{M}\sum_{\mu=\mu_j,j=1}^{M}\left(\frac{\mathrm{d}f_1}{\mathrm{d}\mu} - f_2\right)^2$$

$$\mathcal{L}_3(\mu;\theta) = (f_1(-1;\theta))^2 + (f_2(-1;\theta) - 10)^2,$$

where M is the sample size and where

$$E = \begin{cases} a^2\left(1+\frac{1}{n}\right)f_1^{\left(1+\frac{2}{n}\right)}, & \text{if } \frac{2}{n} \text{ is a positive integer,} \\ a^2\left(1+\frac{1}{n}\right)f_1\left(f_1^2\right)^{\frac{1}{n}}, & \text{otherwise} \end{cases}$$

and where $\boldsymbol{\theta} \in \mathbb{R}^{(4N+2)\times 1}$ is a learnable parameter, which is reshaped by the collecting set $\{W_1, b_1, W_2, b_2\}$. Clearly, if Equation 9 does reduce to zero, then Equation 5 holds.

In order to compute $\frac{\mathrm{d}f_1}{\mathrm{d}\mu}$, $\frac{\mathrm{d}f_2}{\mathrm{d}\mu}$, and $\frac{\partial\mathcal{L}}{\partial\theta}$ we use automatic differentiation (AD) (Baydin et al. 2018), rather than numerical differentiation or integration (Kincaid and Cheney 2002, Chapter 7) based on the assigned grid. AD is a set of techniques for evaluating the derivatives numerically. AD uses symbolic rules for differentiation, however AD evaluates derivatives at particular numeric values, and it does not construct symbolic expressions for derivatives. Automatic differentiation is a powerful tool to automate the calculation of derivatives and is preferable to more traditional methods, especially when differentiating complex algorithms and mathematical functions (Baydin et al. 2018). In Matlab, a dlgradient command takes derivatives with respect to the input or to the parameters.

Table 2 Hyperparameters Setting of the MLP.

	n								
	5	3	1.5	1	0.9	0.7	0.5	0.3	0.1
λ_1	1	1	1	1	1	1	1	1	0.005
λ_2	1	1	1	1	1	1	1	1	0.005
λ_3	1	1	1	1	1	1	1	1	1
N	64	128	50	50	64	64	81	50	81
Nb	50	50	50	100	50	50	50	80	50
η	0.01	0.01	0.005	0.01	0.005	0.005	0.01	0.001	0.001

Table 3 Test performance of the numerical solutions generated by the MLP when $n > 1$.

	n			
	5	3	1.5	1
MSE(F)	2.99×10^{-5}	5.14×10^{-5}	1.22×10^{-6}	1.25×10^{-6}
MSE($\frac{\mathrm{d}F}{\mathrm{d}\mu}$)	2.70×10^{-3}	2.58×10^{-4}	2.61×10^{-5}	1.39×10^{-5}

Table 4 Test performance of the numerical solutions generated by the MLP when $n < 1$.

	n				
	0.9	0.7	0.5	0.3	0.1
MSE(F)	5.78×10^{-6}	8.24×10^{-6}	1.91×10^{-7}	2.43×10^{-6}	6.01×10^{-5}
MSE($\frac{\mathrm{d}F}{\mathrm{d}\mu}$)	5.91×10^{-6}	2.84×10^{-6}	2.22×10^{-6}	4.34×10^{-5}	2.82×10^{-2}

In practice, we use the mini-batch ADAM (Chen et al. 2022) that is a batch of Nb randomly sampled points at every training iteration to minimize $\mathcal{L}$. The M data points are randomly divided into $\frac{M}{Nb}$ batches of size Nb. When all $\frac{M}{Nb}$ batches of data are used for optimization once an epoch is completed. The mini-batch ADAM is an optimization algorithm that can minimize the loss function $\mathcal{L}$.

Select $M = 100{,}000$ points from -1 to 1 at random to train the MLP. Set the $m = 1$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$ for all ns. Set λ_1, λ_2, λ_3, N, Nb, and η, as Table 2, for different n at the same time. Then, update $\boldsymbol{\theta}$ using the mini-batch ADAM algorithm, which is shown as Algorithm 1.

Finally,

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} Loss.$$

For values of $\mu_i = -1 + (i-1)\frac{2}{9999}$ $(1 \leqslant i \leqslant 10{,}000)$, compare the predicted values (F_{MLP}) of the MLP with the numerical solutions (F_{RK}) of Equation 2 using the classical RK4.

Algorithm 1 Mini-batch ADAM Algorithm.

Input: Initialize time step $t = 1$.

Initialize 1st and 2nd moment variables: $m_1 = 0$, $v_1 = 0$.

Set the exponential decay rates for moment estimates β_1, β_2.

i: **while** not converged **do**

ii: Sample a mini-batch of Nb examples from $\{\widetilde{\mu}_1, \widetilde{\mu}_2, \ldots, \widetilde{\mu}_{Nb}\}$

iii: Compute gradient estimate: $\left.\frac{\partial \widetilde{\mathcal{L}}}{\partial \theta}\right|_{\theta=\theta_t}$,

where $\widetilde{\mathcal{L}} = \lambda_1 \widetilde{\mathcal{L}}_1 + \lambda_2 \widetilde{\mathcal{L}}_2 + \lambda_3 \widetilde{\mathcal{L}}_3$

and where

$$\widetilde{\mathcal{L}}_1 = \frac{1}{Nb} \sum_{\widetilde{\mu}=\widetilde{\mu}_j, j=1}^{Nb} \left[\left(1 - \widetilde{\mu}^2\right) \frac{\mathrm{d} f_2}{\mathrm{d}\mu} + n\,(n+1)\, f_1 + E \right]^2$$

$$\widetilde{\mathcal{L}}_2 = \frac{1}{Nb} \sum_{\widetilde{\mu}=\widetilde{\mu}_j, j=1}^{Nb} \left(\frac{\mathrm{d} f_1}{\mathrm{d}\mu} - f_2 \right)^2$$

$$\widetilde{\mathcal{L}}_3 = L_3$$

iv: Update biased first moment estimate: $m_{t+1} = \beta_1 m_t + (1 - \beta_1) \left.\frac{\partial \widetilde{\mathcal{L}}}{\partial \theta}\right|_{\theta=\theta_t}$

v: Update biased second moment estimate:

$v_{t+1} = \beta_2 v_t + (1 - \beta_2) \left(\left.\frac{\partial \widetilde{\mathcal{L}}}{\partial \theta}\right|_{\theta=\theta_t} \right)^2$

vi: Correct bias in first moment: $\widetilde{m}_{t+1} = \frac{m_{t+1}}{1-\beta_1^t}$

vii: Correct bias in second moment: $\widetilde{v}_{t+1} = \frac{v_{t+1}}{1-\beta_2^t}$

viii: Update $\theta_{t+1} = \theta_t - \eta \frac{\widetilde{m}_{t+1}}{\sqrt{\widetilde{v}_{t+1}} + \varepsilon}$

ix: update $t \leftarrow t + 1$

x: **end while**

Output: $\boldsymbol{\theta_{t+1}}$

Define the mean square error (MSE):

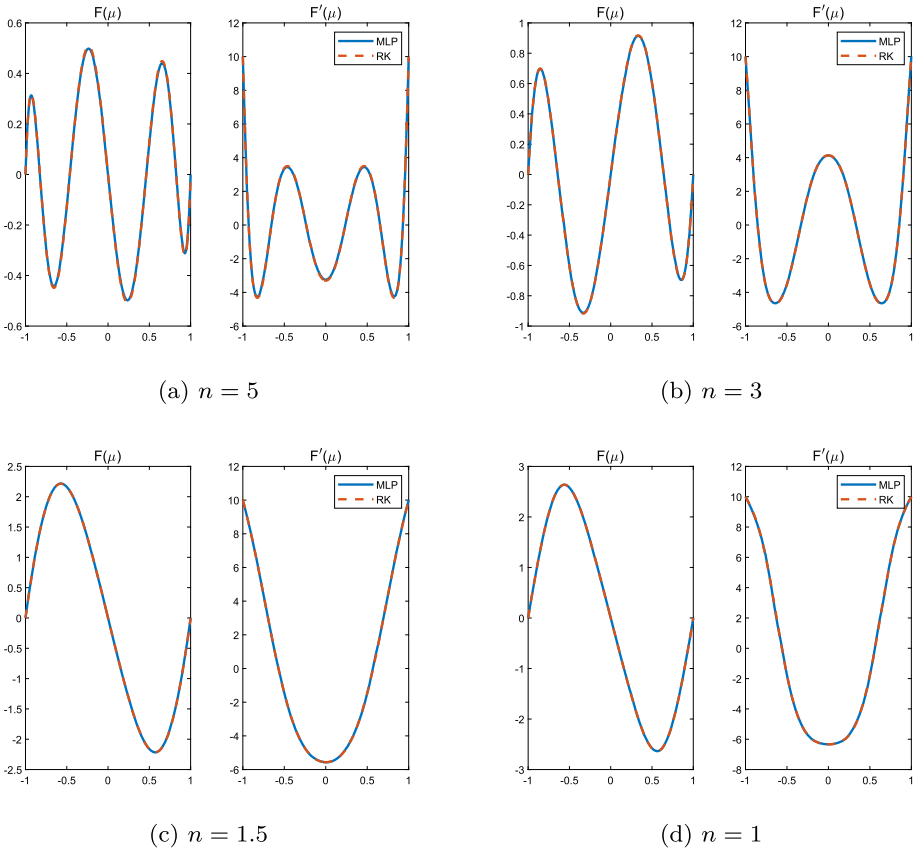
$$\mathrm{MSE}(F) = \frac{1}{10{,}000} \sum_{i=1}^{10{,}000} \left[F_{\mathrm{MLP}}(\mu_i) - F_{\mathrm{RK}}(\mu_i) \right]^2,$$

$$\mathrm{MSE}\left(\frac{\mathrm{d}F}{\mathrm{d}\mu}\right) = \frac{1}{10{,}000} \sum_{i=1}^{10{,}000} \left[\frac{\mathrm{d}F}{\mathrm{d}\mu}_{\mathrm{MLP}}(\mu_i) - \frac{\mathrm{d}F}{\mathrm{d}\mu}_{\mathrm{RK}}(\mu_i) \right]^2,$$

to measure how close the $F_{\mathrm{MLP}}(\mu_i)$ is to the $F_{\mathrm{RK}}(\mu_i)$ and the $\left(\frac{\mathrm{d}F}{\mathrm{d}\mu}_{\mathrm{MLP}}\right)(\mu_i)$ is to the $\left(\frac{\mathrm{d}F}{\mathrm{d}\mu}\right)_{\mathrm{RK}}(\mu_i)$, respectively. The MSEs are shown in Tables 3 and 4. This shows that the MLP approach is efficient, compared with the RK4.

Figures 4 and 5 show how close the solutions generated by MLPs are to the RK method. Therefore, they illustrate that MLP works reasonably well.

We developed a numerical algorithm that can effectively solve a specific class of ODEs, particularly those derived from $\frac{\partial}{\partial \phi} = 0$. One notable aspect of our algorithm is its remarkable



(a) $n = 5$ (b) $n = 3$

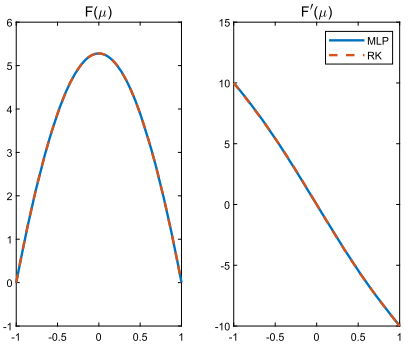
(c) $n = 1.5$ (d) $n = 1$

Figure 4 F, $\frac{\mathrm{d}F}{\mathrm{d}\mu}$ generated by MLP and RK methods when $n \geq 1$.

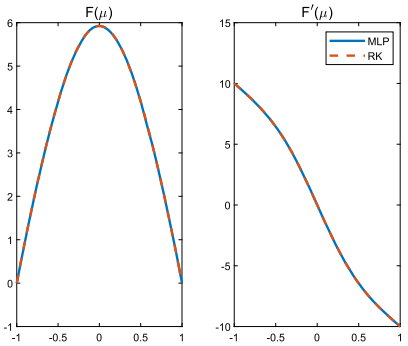
adaptability, which facilitates effortless adjustments to a wide range of initial and boundary conditions through simple modifications of the loss function in our proposed artificial-intelligence (AI) neural-network model. Lerche and Low (2014) generalized the equilibria proposed by Low and Lou (1990) and made modifications to the initial conditions. In this scenario, our numerical algorithm remains valid and applicable. To avoid unnecessary repetition, we have included it as Appendix B.

4. Data-Driven Approach for Identifying the Parameters n and a

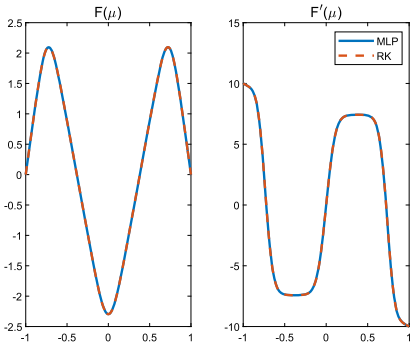
The objective is to identify the optimum parameters of Low and Lou's equilibria to match the observations of an active region at the photosphere. This process concerns an inverse problem: given a measured magnetic field $\boldsymbol{B}$ at the photosphere, or its value $P\boldsymbol{B}$ under a measurement operator P, determine a corresponding parameter set $\{a, n\}$ such that the neural-network solution $f\,(r_i, \mu_i; a, n)$ can approximate the field $\boldsymbol{B}$.



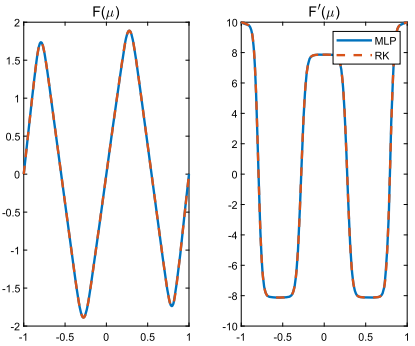
(a) $n = 0.9$



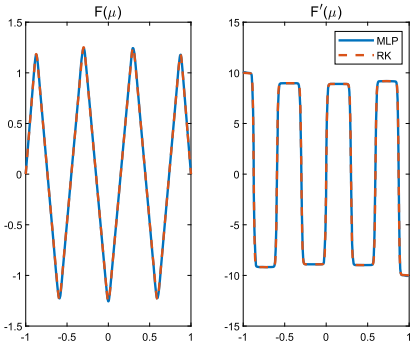
(b) $n = 0.7$



(c) $n = 0.5$

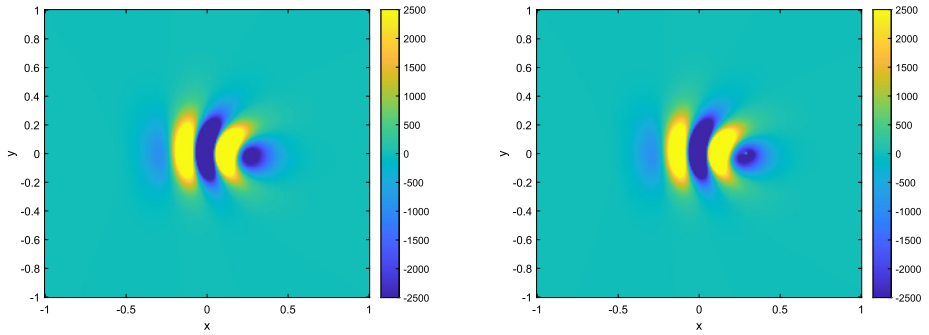


(d) $n = 0.3$



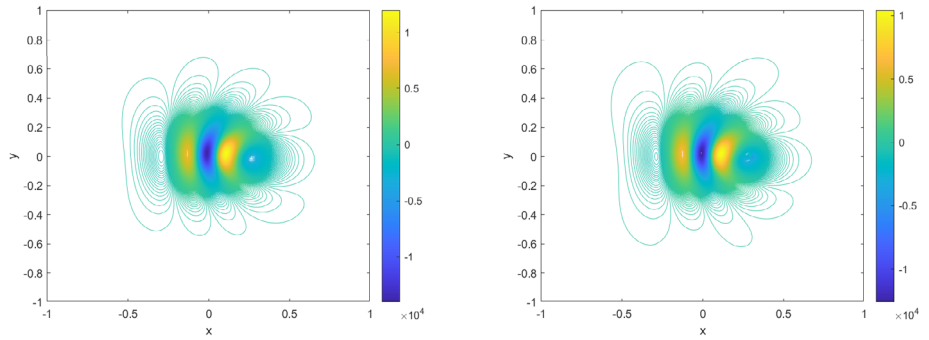
(e) $n = 0.1$

Figure 5 F, $\frac{\mathrm{d}F}{\mathrm{d}\mu}$ generated by MLP and RK methods when $n < 1$.



(a) na_1. (b) na_2

Figure 6 The magnetogram B_x that is generated by na_1 and na_2, respectively. Please note that the magnetograms are derived from the equilibria by Low and Lou and are dimensionless.



(a) na_1. (b) na_2

Figure 7 The contour of B_x that is generated by na_1 and na_2, respectively. Please note that the contours are derived from the equilibria by Low and Lou and are dimensionless.

To analyze a force-free field $\boldsymbol{B}$ with the additional condition $\frac{\partial}{\partial\phi} = 0$ imposed in the spherical coordinate system, we consider the transformation matrix P defined as $P = \begin{bmatrix} 0 & \frac{r^{n+2}\left(1-\mu^2\right)^{\frac{1}{2}}}{n} & 0 \end{bmatrix}$, see Equation 19, then one obtains that

$$P\boldsymbol{B} = \begin{bmatrix} 0 & \frac{r^{n+2}\left(1-\mu^2\right)^{\frac{1}{2}}}{n} & 0 \end{bmatrix} \begin{bmatrix} B_r & B_\theta & B_\phi \end{bmatrix}^T = F.$$

The loss function of the neural network can be designed as

$$\mathcal{L} = \lambda_1 \mathcal{L}_1 + \lambda_2 \mathcal{L}_2,$$

where

$$\mathcal{L}_1 = \frac{1}{M} \sum_{i=1}^{M} \left(\left(1-\mu_i^2\right) \frac{\mathrm{d}^2 G}{\mathrm{d}\mu^2} + \theta_2 \left(\theta_2 + 1\right) G + \theta_3 \left(1 + \frac{1}{\theta_2}\right) G \left(G^2\right)^{\frac{1}{\theta_2}} \right)^2$$

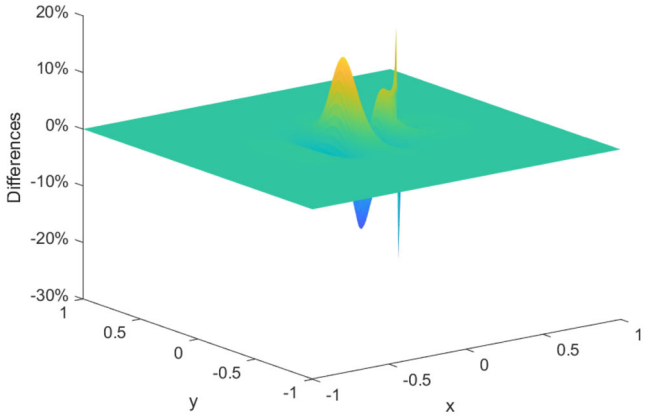


Figure 8 The normalized differences generated by na_1 and na_2.

and

$$\mathcal{L}_2 = \frac{1}{M}\sum_{i=1}^{M}\left(1-\mu_i^2\right)^{\frac{1}{2}}\frac{r^{\theta_2+2}}{\theta_2}\left((B_\theta)_i - f\left(\mu_i;\theta_1\right)\right)^2$$

and

$$G\left(\mu_i\right) = \frac{r^{n+2}}{n}\left(1-\mu_i^2\right)^{\frac{1}{2}} f\left(\mu_i;\theta_1\right).$$

In the above equations, λ_1 and λ_2 are two trade-off parameters: M represents the sample size, and $\boldsymbol{\theta}_1$ is a learnable vector for B_θ. Furthermore, θ_2 and θ_3 are two learnable parameters associated with n and a, respectively.

For a simple case, the active region is generated by Low and Lou's approach with $n = 5$ and $a = 3.9341$ ($na_1 = \{n = 5, a = 3.9341\}$). We use an MLP with five layers to identify the parameters n and a. The first layer has three inputs corresponding to the r, μ, and B_θ. There are 64 neurons for each hidden layer. The last layer has three outputs: the estimated values of the n, a, and B_θ. Set $\lambda_1 = \lambda_2 = 1$ and $M = 65{,}536$. After training the MLP, the outputs n and a are 4.9042 and 3.9830 ($na_2 = \{n = 4.9042, a = 3.9830\}$), respectively.

Figure 6 shows the magnetogram B_x that is generated by na_1 and na_2, respectively. The magnetic-field intensity in Figure 6 is visualized in the range of -5000 to 5000. Any values exceeding 5000 are limited to 5000, and any values below -5000 are set to -5000. Figure 7 shows the contours of B_x that are generated by na_1 and na_2, respectively. They are highly compatible in visual representation. In Figure 7, the colorbars are displayed in arbitrary units. It is worth noting that magnetograms can be represented as matrices in Cartesian coordinates (x, y). Figure 8 illustrates a visual representation of the quantity $\frac{(B_{n2})_x - (B_{n1})_x}{\max|(B_{n1})_x|}$. However, the largest differences occur close to the magnetic nulls in Figure 8 since the error propagation rapidly increases in Equation 3 when r is too small. The cosine similarity (Brockmeier et al. 2017) between the magnetograms generated by na_1 and na_2 is 0.9909. Cosine similarity is a measure that calculates the cosine value of the angle between two matrices. It ranges from -1 to 1, with a value closer to 1 indicating a higher degree of similarity between the two magnetograms.

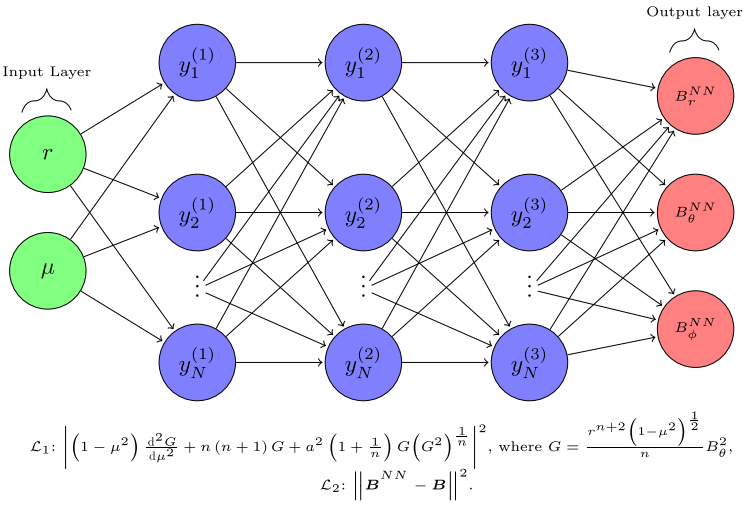


Figure 9 Neural-network architecture for the observational data.

If Low and Lou's equilibria can effectively approximate the solar photospheric observational data, we can utilize a neural network to determine the parameters and initial values of the Low and Lou's equilibria. Therefore, a neural network as shown in Figure 9 is applied to NOAA active regions (ARs) 11158 and 11302. This process can be written as:

$$\begin{bmatrix} B_r \\ B_\theta \\ B_\phi \end{bmatrix} = W^4\sigma\left(W^3\sigma\left(W^2\sigma\left(W^1\begin{bmatrix} r \\ \mu \end{bmatrix}+b^1\right)+b^2\right)+b^3\right)+b^4,$$

where W^1, W^2, W^3, and W^4 represent the weight matrices for each layer, while σ represents the activation function. The biases are represented by b^1, b^2, b^3, and b^4.

The outputs of the neural network are shown in Figures 10c and d for NOAA ARs 11158 and 11302 from the *Solar Dynamics Observatory/Helioseismic and Magnetic Imager* (SDO/HMI), respectively. It can be observed that we cannot obtain effective magnetic fields referring to Figures 10a and b. With only the data-loss term $\mathcal{L}_2$, we can obtain reasonable outputs as shown in Figures 10e and f for NOAA ARs 11158 and 11302, respectively.

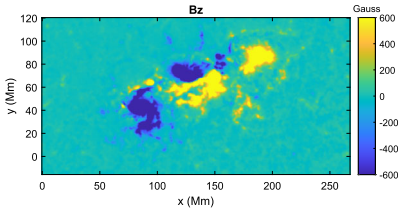
According to Wiegelmann and Sakurai (2021), the necessary conditions for determining whether the solar photospheric magnetic field is a nonlinear force-free field are:

$$\int_S B_z\left(x,y,0\right)\mathrm{d}x\,\mathrm{d}y=0,\quad \frac{|F_x|}{F_p}\ll 1,\frac{|F_y|}{F_p}\ll 1,\frac{|F_z|}{F_p}\ll 1,$$

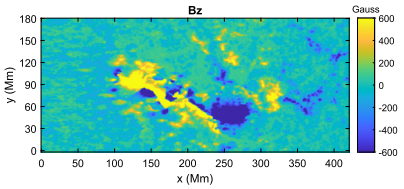
where

$$F_x=-\frac{1}{4\pi}\int_S B_xB_z\mathrm{d}x\,\mathrm{d}y,\quad F_y=-\frac{1}{4\pi}\int_S B_yB_z\mathrm{d}x\,\mathrm{d}y,$$

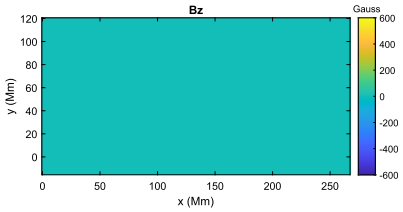
$$F_z=\frac{1}{8\pi}\int_S\left(B_x^2+B_y^2-B_z^2\right)\mathrm{d}x\,\mathrm{d}y,\quad F_p=\frac{1}{8\pi}\int_S\left(B_x^2+B_y^2+B_z^2\right)\mathrm{d}x\,\mathrm{d}y.$$



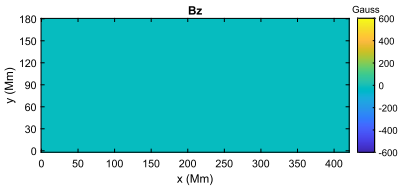
(a) Observational data for NOAA AR 11158.



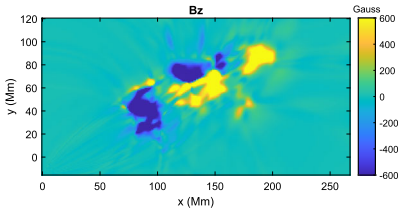
(b) Observational data for NOAA AR 11302.



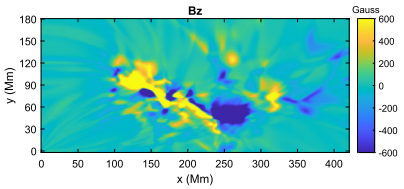
(c) Output for NOAA AR 11158 under $\lambda_1\mathcal{L}_1+\lambda_2\mathcal{L}_2$.



(d) Output for NOAA AR 11302 under $\lambda_1\mathcal{L}_1+\lambda_2\mathcal{L}_2$.



(e) Output for NOAA AR 11158 under $\mathcal{L}_2$.



(f) Output for NOAA AR 11302 under $\mathcal{L}_2$.

Figure 10 The neural-network outputs for NOAA ARs 11158 and 11302.

For practical computations, the acceptable conditions for flux imbalance, as stated by Moon et al. (2002), are defined as:

$$\frac{\left|F^{+}-F^{-}\right|}{F^{+}+F^{-}} \leqslant 0.1,$$

where F^{+} and F^{-} represent the upward and downward magnetic fluxes, respectively. Additionally, the vertical-force condition, as mentioned by Liu et al. (2013), is given by

$$\frac{|F_z|}{F_p} < 0.1,$$

where F_z denotes the vertical force and F_p represents the total magnetic pressure. If these conditions are satisfied, the magnetic field can be approximated as a force-free field $(\nabla \times \boldsymbol{B}) \times \boldsymbol{B} = 0$. The active-region data that we used in our study meets these requirements. One possible reason for the code not working is that Low and Lou's equilibria may not accurately approximate certain observational data, such as for NOAA AR 11158. To effectively utilize Low and Lou's equilibria, it is crucial to regularize the observational data to approximate the condition $\frac{\partial}{\partial\phi} = 0$. This regularization term is also essential to ensure that

the modified data closely resembles the original data. As is evident from Figures 10c and d, including this regularization term is indispensable. However, the specific form of this regularization term has not yet been determined. In our future work, we will continue to explore and optimize this regularization term.

5. Conclusion

In this article, testing a numerical algorithm for solar magnetic-field extrapolation using Low and Lou's equilibria proves to be beneficial. We have presented a machine-learning-based numerical method that effectively determines the parameter a and the function F in Low and Lou's equilibria. By employing the MLP neural network, we successfully implemented this algorithm. An area of crucial investigation lies in the adaptive selection of MLP's width and the parameters λ_1, λ_2, and λ_3 in Equation 9.

Furthermore, we have proposed a machine-learning algorithm to address the inverse problem of Low and Lou's equilibria. While it performs well on generated data, it falls short when applied to observational data. An intriguing avenue for future research involves optimizing the parameters of the generalized equilibria proposed by Low and Lou to better align with observations of active regions on the photosphere.

Moreover, a promising direction for the future is to combine solar photospheric observation data with artificial-intelligence techniques for coronal magnetic-field extrapolation. This integration holds potential for further advancements in understanding and predicting solar phenomena.

Appendix A

We focus here on mathematical concepts and derive Equations 3 and 5.

The following two theorems are commonly used in the formulation and analysis of magnetic fields.

Theorem 2 (Davis, Snider, and Davis 1979) *A vector field $\boldsymbol{B}$ is continuously differentiable in a simply connected domain D, $\nabla \cdot \boldsymbol{B} = 0$ if, and only if, there is a vector field $\boldsymbol{A}$ such that $\boldsymbol{B} = \nabla \times \boldsymbol{A}$ throughout D.*

Remark 2 $\boldsymbol{A}$ is not unique, in fact, $\nabla \cdot [\nabla \times (\beta \boldsymbol{A})] = 0$ for any $\beta \in \mathbb{R}$.

Theorem 3 (Davis, Snider, and Davis 1979) *Given any position vector $\boldsymbol{r} = x\widehat{\boldsymbol{x}} + y\widehat{\boldsymbol{y}} + z\widehat{\boldsymbol{z}}$, if $x = x(u_1, u_2, u_3)$, $y = y(u_1, u_2, u_3)$, $z = z(u_1, u_2, u_3)$, let $\boldsymbol{e_i} = \frac{1}{h_i}\frac{\partial \boldsymbol{r}}{\partial u_i}$, where $h_i = \left|\frac{\partial \boldsymbol{r}}{\partial u_i}\right|$ $(i = 1, 2, 3)$. If $\{\boldsymbol{e_1}, \boldsymbol{e_2}, \boldsymbol{e_3}\}$ are orthogonal curvilinear coordinates in $\mathbb{R}^3$, then for $\forall \boldsymbol{A} = A_x\widehat{\boldsymbol{x}} + A_y\widehat{\boldsymbol{y}} + A_z\widehat{\boldsymbol{z}} = A_1\boldsymbol{e_1} + A_2\boldsymbol{e_2} + A_3\boldsymbol{e_3}$ we have that*

$$\nabla \times \boldsymbol{A} = \frac{1}{h_1 h_2 h_3} \begin{vmatrix} h_1\boldsymbol{e_1} & h_2\boldsymbol{e_2} & h_3\boldsymbol{e_3} \\ \frac{\partial}{\partial u_1} & \frac{\partial}{\partial u_2} & \frac{\partial}{\partial u_3} \\ h_1 A_1 & h_2 A_2 & h_3 A_3 \end{vmatrix}.$$

Let $u_1 = r$, $u_2 = \theta$, $u_3 = \phi$, and $x = r\sin\theta\cos\phi$, $y = r\sin\theta\sin\phi$, $z = r\cos\phi$. By the assumption of Low and Lou

$$\frac{\partial}{\partial u_3} = \frac{\partial}{\partial \phi} = 0$$

and using Theorems 2 and 3, we obtain the axisymmetric magnetic NLFFF $\boldsymbol{B}$

$$\begin{aligned}\boldsymbol{B} &= \nabla \times \boldsymbol{A} \\ &= \frac{1}{r\sin\theta}\left(\frac{1}{r}\frac{\partial}{\partial\theta}\left(r\sin\theta A_\phi\right)\widehat{\boldsymbol{r}} - \frac{\partial}{\partial r}\left(r\sin\theta A_\phi\right)\widehat{\boldsymbol{\theta}} + \sin\theta\left(\frac{\partial}{\partial r}\left(rA_\theta\right) - \frac{\partial A_r}{\partial\theta}\right)\widehat{\boldsymbol{\phi}}\right).\end{aligned}$$

It is convenient to introduce $\widetilde{A} = r\sin\theta A_\phi$, $b_\phi = \sin\theta\left(\frac{\partial}{\partial r}\left(rA_\theta\right) - \frac{\partial A_r}{\partial\theta}\right)$, then

$$\boldsymbol{B} = \frac{1}{r\sin\theta}\left(\frac{1}{r}\frac{\partial\widetilde{A}}{\partial\theta}\widehat{\boldsymbol{r}} - \frac{\partial\widetilde{A}}{\partial r}\widehat{\boldsymbol{\theta}} + b_\phi\widehat{\boldsymbol{\phi}}\right). \tag{10}$$

Substituting Equation 10 into $\boldsymbol{B} = \alpha\boldsymbol{B}$ and simplifying, we obtain

$$\frac{1}{r}\frac{\partial b_\phi}{\partial\theta} = \alpha\frac{1}{r}\frac{\partial}{\partial\theta}\widetilde{A}, \tag{11}$$

$$\frac{\partial b_\phi}{\partial r} = \alpha\frac{\partial}{\partial r}\widetilde{A}, \tag{12}$$

$$-\left(\frac{\partial}{\partial r}\left(\frac{\partial\widetilde{A}}{\partial r}\right) + \frac{1}{r^2}\sin\theta\frac{\partial}{\partial\theta}\left(\frac{1}{\sin\theta}\frac{\partial\widetilde{A}}{\partial\theta}\right)\right) = \alpha b_\phi. \tag{13}$$

Combining Equations 11 and 12 to eliminate α,

$$\frac{\frac{\partial b_\phi}{\partial\theta}}{\frac{\partial\widetilde{A}}{\partial\theta}} = \frac{\frac{\partial b_\phi}{\partial r}}{\frac{\partial\widetilde{A}}{\partial r}} \Rightarrow 0 = \frac{\partial b_\phi}{\partial\theta}\frac{\partial\widetilde{A}}{\partial r} - \frac{\partial b_\phi}{\partial r}\frac{\partial\widetilde{A}}{\partial\theta} = \begin{vmatrix}\frac{\partial b_\phi}{\partial r} & \frac{\partial b_\phi}{\partial\theta} \\ \frac{\partial\widetilde{A}}{\partial r} & \frac{\partial\widetilde{A}}{\partial\theta}\end{vmatrix} = J\left(b_\phi, \widetilde{A}\right).$$

The following theorem provided us with the relationship between b_ϕ and $\widetilde{A}$.

Theorem 4 (Tolstykh 2020) *If f_3, f_4 are continuously differentiable functions from $(x, y) \to \mathbb{R}$ such that the determinant of the Jacobian vanishes everywhere, then $f_4(x, y) = H(f_3(x, y))$ or $f_3(x, y) = H(f_4(x, y))$, where H is a continuously differentiable function.*

Therefore, $b_\phi = b_\phi\left(\widetilde{A}\right)$, i.e. b_ϕ is an arbitrary function of $\widetilde{A}$.
Taking derivatives of b_ϕ with respect to r, we obtain

$$\frac{\partial b_\phi}{\partial r} = \frac{\partial b_\phi}{\partial\widetilde{A}}\frac{\partial\widetilde{A}}{\partial r} = \frac{\mathrm{d}b_\phi}{\mathrm{d}\widetilde{A}}\frac{\partial\widetilde{A}}{\partial r}. \tag{14}$$

Combining Equations 14 and 12 to obtain α,

$$\alpha = \frac{\mathrm{d}b_\phi}{\mathrm{d}\widetilde{A}}. \tag{15}$$

Substituting Equation 15 into 13, we obtain

$$\frac{\partial^2\widetilde{A}}{\partial r^2} + \frac{\sin\theta}{r^2}\frac{\partial}{\partial\theta}\left(\frac{1}{\sin\theta}\frac{\partial\widetilde{A}}{\partial\theta}\right) + \frac{\mathrm{d}}{\mathrm{d}\widetilde{A}}\left(\frac{1}{2}b_\phi^2\right) = 0, \tag{16}$$

known as the Grad–Shafranov equation.

Low and Lou found a set of solutions to Equation 16 if

$$\widetilde{A} = \widetilde{A}(r,\theta) = F(\cos\theta)\, r^{-n}, b_\phi = \begin{cases} a\widetilde{A}^{1+\frac{1}{n}}, n = 1 \\ a\widetilde{A}\left|\widetilde{A}\right|^{\frac{1}{n}}, n > 1 \end{cases} \tag{17}$$

for odd n, and a real constant a. Equation 16 then reduces to

$$\left(1-\mu^2\right)\frac{\mathrm{d}^2F}{\mathrm{d}\mu^2} + n(n+1)F + a^2\left(1+\frac{1}{n}\right)F^{1+\frac{2}{n}} = 0, \quad n = 1,$$
$$\left(1-\mu^2\right)\frac{\mathrm{d}^2F}{\mathrm{d}\mu^2} + n(n+1)F + a^2\left(1+\frac{1}{n}\right)F\left(F^2\right)^{\frac{1}{n}} = 0, \quad n > 1,$$

where $\mu = \cos\theta$. Thereby, a family of axisymmetric NLFFF can be generated as Equation 3.

Moreover, substituting Equation 17 into Equation 10, we obtain

$$B_r = \frac{1}{r\sin\theta}\left[\frac{1}{r}\frac{\partial}{\partial\theta}\left(\frac{F(\cos\theta)}{r^n}\right)\right] = \frac{1}{r^n\sin\theta}\frac{1}{r^2}\frac{\partial F}{\partial\cos\theta}\frac{\partial\cos\theta}{\partial\theta} = -\frac{1}{r^{n+2}}\frac{\partial F}{\partial\mu}, \tag{18}$$

$$B_\theta = \frac{1}{r\sin\theta}\left[-\frac{\partial}{\partial r}\left(\frac{F(\cos\theta)}{r^n}\right)\right] = \frac{n}{r^{n+2}\sin\theta}F = \frac{n}{r^{n+2}}\frac{1}{\left(1-\mu^2\right)^{\frac{1}{2}}}F \tag{19}$$

and

$$B_\phi = \begin{cases} a\frac{1}{r^{n+2}}\frac{1}{\left(1-\mu^2\right)^{\frac{1}{2}}}F^{1+\frac{1}{n}}, & n = 1, \\ a\frac{1}{r^{n+2}}\frac{1}{\left(1-\mu^2\right)^{\frac{1}{2}}}F|F|^{\frac{1}{n}}, & n > 1. \end{cases} \tag{20}$$

Finally, combining Equations 18, 19, and 20, we obtain

$$\boldsymbol{B} = \frac{1}{r^{n+2}}\left(-\frac{\mathrm{d}F}{\mathrm{d}\mu}\widehat{\boldsymbol{r}} + \frac{nF}{\left(1-\mu^2\right)^{\frac{1}{2}}}\widehat{\boldsymbol{\theta}} + r^{n+2}B_\phi\widehat{\boldsymbol{\phi}}\right). \tag{21}$$

Next, we multiply Equation 21 by $B_0 r_0^{n+2}$ using Remark 2, and some constant, and obtain Equation 3, where $\eta = aB_0r_0^{n+2}$.

This completes the derivation of Equations 3 and 5 in detail.

Table 5 The values of λ in Lerche and Low (2014) and our numerical algorithm for the even functions.

m	$n=0.5$		$n=1$		$n=4$	
	Lerche and Low's	ours	Lerche and Low's	ours	Lerche and Low's	ours
1	2.47	2.4702	0.0	3.7290×10^{-10}	12.24	12.2523
2	28.69	28.6935	17.9	17.8673	44.45	44.4538
3	79.7	79.6805	52.5	52.4641	−−	86.8627

Table 6 The values of λ in Lerche and Low (2014) and our numerical algorithm for the odd functions.

m	$n=0.5$		$n=1$		$n=4$	
	Lerche and Low's	ours	Lerche and Low's	ours	Lerche and Low's	ours
1	271	270.9715	34.3	34.2966	68.5	68.5917
2	15,700	15,714.5875	603.0	602.9976	186.75	186.9304

Appendix B

By conveniently adjusting the corresponding loss-function term in our model, we can tailor our approach to effectively solve various types of numerical ODE problems. With this in mind, we are actively engaged in the pursuit of solutions of the type of those of Lerche and Low (2014), i.e.

$$(1-\mu^2)\frac{\mathrm{d}^2 A_n}{\mathrm{d}\mu^2} + n\,(n+1)\,A_n + \lambda A_n\left(A_n^2\right)^{\frac{1}{n}} = 0, \tag{22}$$

where $\lambda = \frac{n+1}{n}a^2$.

The boundary condition for Equation 22 is

$$A_n\,(-1) = 0, \quad A_n\,(1) = 0 \tag{23}$$

and the initial condition for Equation 22 is

$$\begin{cases} A_n\,(0) = 1,\ A_n'\,(0) = 0; & \text{if } A_n\,(\mu) \text{ is an even function,} \\ A_n\,(0) = 0,\ A_n'\,(0) = 1; & \text{if } A_n\,(\mu) \text{ is an odd function.} \end{cases} \tag{24}$$

Regarding the aforementioned problem, the loss function in Equation 9 of the AI model that we proposed is modified as follows:

$$\mathcal{L}\,(\mu;\theta) = \lambda_1\mathcal{L}_1\,(\mu;\theta) + \lambda_2\mathcal{L}_2\,(\mu;\theta) + \lambda_3\mathcal{L}_3\,(\mu;\theta)\,, \tag{25}$$

Table 7 The MSEs of A_n and A'_n generated by the AI solver we proposed and the RK solver of Lerche and Low (2014) when A_n is an even function.

m	$n=0.5$		$n=1$		$n=4$	
	MSE($A_{0.5}$)	MSE($A'_{0.5}$)	MSE(A_1)	MSE(A'_1)	MSE(A_4)	MSE(A'_4)
1	8.2481×10^{-8}	2.1942×10^{-6}	1.0643×10^{-6}	4.3965×10^{-6}	2.8218×10^{-6}	2.0501×10^{-4}
2	2.5027×10^{-6}	7.9311×10^{-5}	1.0328×10^{-7}	1.4201×10^{-5}	5.1145×10^{-6}	1.6322×10^{-3}
3	1.1788×10^{-6}	4.7149×10^{-5}	3.1746×10^{-6}	1.7302×10^{-4}	6.5704×10^{-7}	1.9401×10^{-3}

Table 8 The MSEs of A_n and A'_n generated by the AI solver we proposed and the RK solver of Lerche and Low (2014) when A_n is an odd function.

m	$n=0.5$		$n=1$		$n=4$	
	MSE($A_{0.5}$)	MSE($A'_{0.5}$)	MSE(A_1)	MSE(A'_1)	MSE(A_4)	MSE(A'_4)
1	2.2043×10^{-7}	7.2184×10^{-6}	2.8732×10^{-7}	1.5746×10^{-5}	2.1892×10^{-7}	5.0450×10^{-5}
2	2.4288×10^{-7}	1.0826×10^{-5}	2.2260×10^{-7}	2.0346×10^{-5}	8.7093×10^{-7}	1.3188×10^{-4}

where

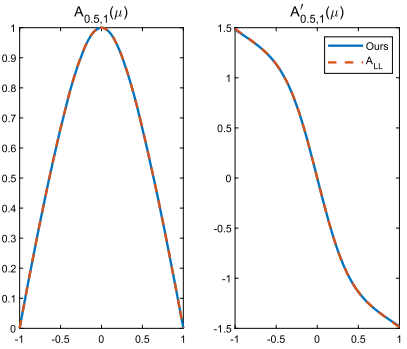
$$\mathcal{L}_1(\mu;\theta)=\frac{1}{M}\sum_{\mu=\mu_j,j=1}^{j=M}\left(\left(1-\mu^2\right)\frac{\mathrm{d}f_2}{\mathrm{d}\mu}+n\left(n+1\right)f_1+\lambda f_1\left(f_1^2\right)^{\frac{1}{n}}\right)^2,$$

$$\mathcal{L}_2(\mu;\theta)=\frac{1}{M}\sum_{\mu=\mu_j,j=1}^{M}\left(\frac{\mathrm{d}f_1}{\mathrm{d}\mu}-f_2\right)^2,$$

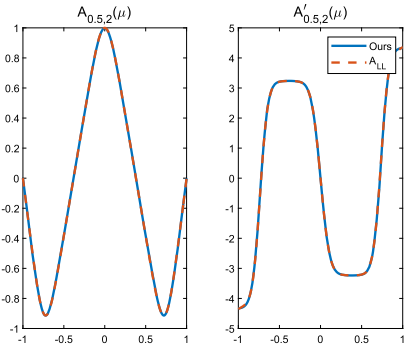
$$\mathcal{L}_3(\mu;\theta)=\begin{cases}\left(f_1(0;\theta)-1\right)^2+\left(f_2(0;\theta)\right)^2, & \text{if } f_1 \text{ is an even function,}\\ \left(f_1(0;\theta)\right)^2+\left(f_2(0;\theta)-1\right)^2, & \text{if } f_1 \text{ is an odd function.}\end{cases}$$

We have obtained the values for λ, which are listed in Tables 5 and 6. It is important to note that these corresponding values are extracted from the captions of Figures 2, 3, and 4 in Lerche and Low (2014). It can be observed that they are very close but still exhibit subtle differences, which can be attributed to factors such as grid partitioning or machine precision.

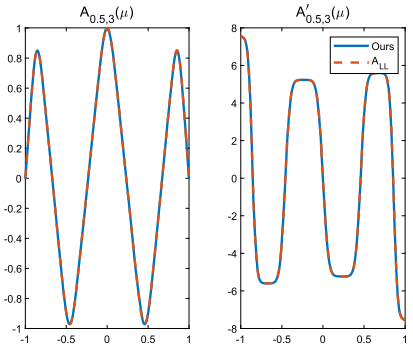
For the values of μ_i $(-1\le\mu_i\le1)$, we compare the numerical values of the AI solver we proposed with the numerical solutions (A_{LL}) to Equation 22 in Lerche and Low (2014). The MSEs are listed in Tables 7 and 8. Figures 11, 12, and 13 show how close the solutions generated by the AI solver are to the RK solver of Lerche and Low (2014). They illustrate that the AI solver works reasonably well.



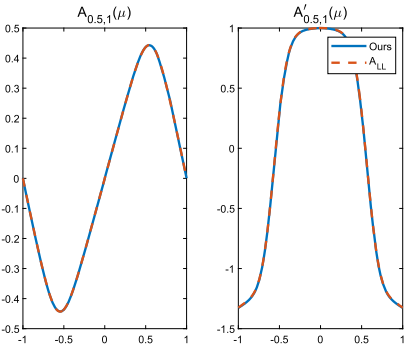
(a) Even function $A_{0.5,1}$.



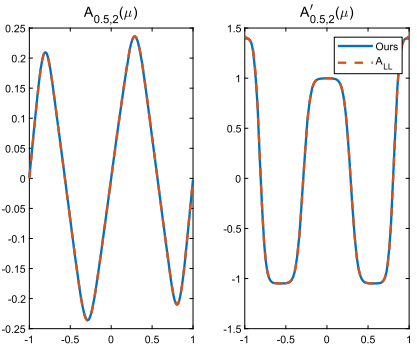
(b) Even function $A_{0.5,2}$.



(c) Even function $A_{0.5,3}$.

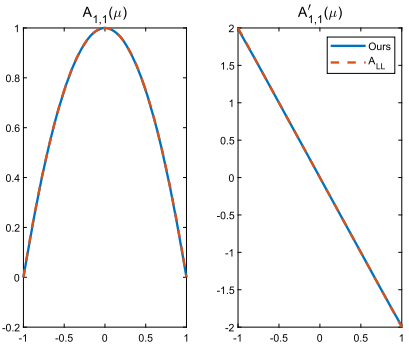


(d) Odd function $A_{0.5,1}$.

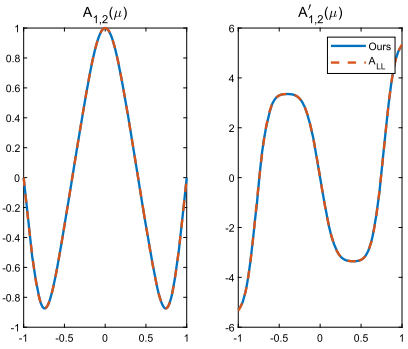


(e) Odd function $A_{0.5,2}$.

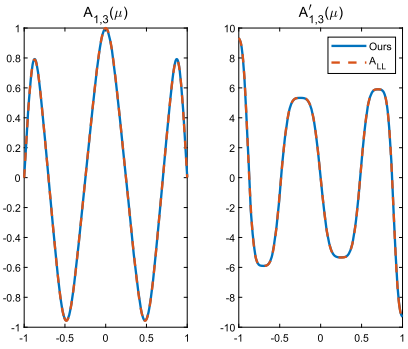
Figure 11 The values of A_n and A'_n obtained by the AI solver that we proposed and the RK solver of Lerche and Low (2014), respectively, when $n = 0.5$.



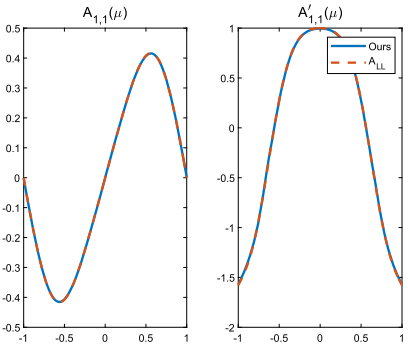
(a) Even function $A_{1,1}$.



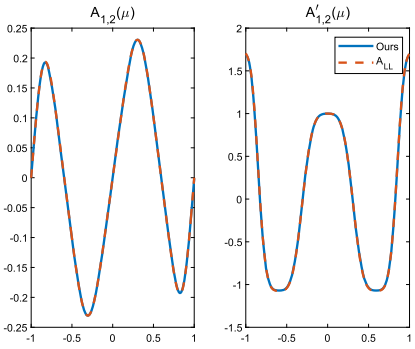
(b) Even function $A_{1,2}$.



(c) Even function $A_{1,3}$.

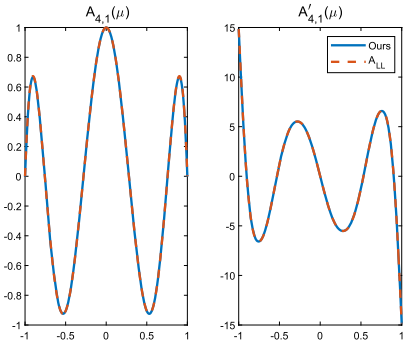


(d) Odd function $A_{1,1}$.

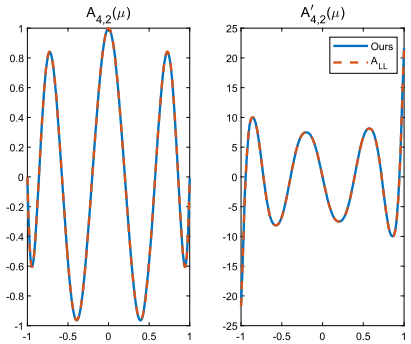


(e) Odd function $A_{1,2}$.

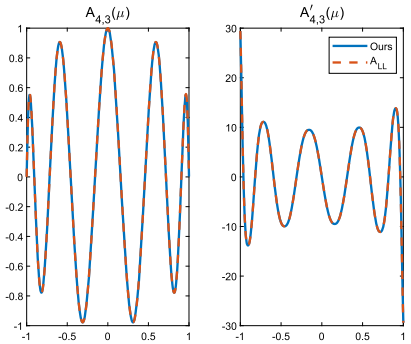
Figure 12 The values of A_n and A'_n obtained by the AI solver that we proposed and the RK solver of Lerche and Low (2014), respectively, when $n = 1$.



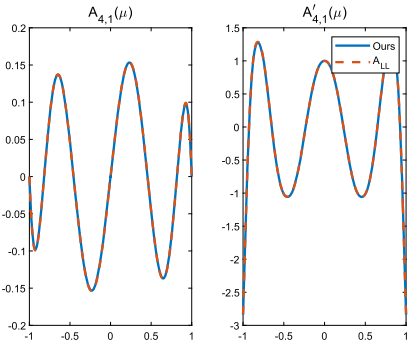
(a) Even function $A_{4,1}$.



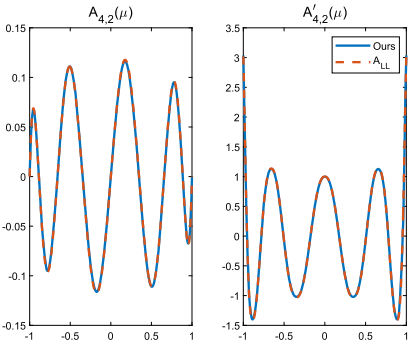
(b) Even function $A_{4,2}$.



(c) Even function $A_{4,3}$.



(d) Odd function $A_{4,1}$.



(e) Odd function $A_{4,2}$.

Figure 13 The values of A_n and A'_n obtained by the AI solver that we proposed and the RK solver of Lerche and Low (2014), respectively, when $n = 4$.

Acknowledgments We thank the anonymous reviewer for a very helpful and constructive review.

Author contributions Y. Zhang: Conceptualization, Methodology, Conduct numerical experiments, Literature survey, Review latest development, Prepare initial manuscript. L. Xu: Methodology discussion, Approach

development, Manuscript revision, Experiment verification. Y. Yan: Methodology discussion, Approach development, Manuscript revision.

Funding This work was supported by the National Key R&D Program of China (Nos. 2022YFE0133700 and 2021YFA1600504) and the National Natural Science Foundation of China (NSFC) (Nos. 11790305, 11973058, and 12103064).

Code Availability The code can be accessed at github.com/zhims/Coding/tree/main/SoPh2023b.

Declarations

Competing interests The authors declare no competing interests.

References

Altschuler, M., Newkirk, G.: 1969, Magnetic fields and the structure of the solar corona I: methods of calculating coronal fields. *Solar Phys.* **9**, 131. DOI. ADS.

Amari, T., Aly, J., Luciani, J., Boulmezaoud, T., Mikic, Z.: 1997, Reconstructing the solar coronal magnetic field as a force-free magnetic field. *Solar Phys.* **174**, 129. DOI. ADS.

Aulanier, G., Pariat, E., Démoulin, P.: 2005, Current sheet formation in quasi-separatrix layers and hyperbolic flux tubes. *Astron. Astrophys.* **444**, 961. DOI. ADS.

Baydin, A., Pearlmutter, B., Radul, A., Siskind, J.: 2018, Automatic differentiation in machine learning: a survey. *J. Mach. Learn. Res.* **18**, 1.

Brockmeier, A., Mu, T., Ananiadou, S., Goulermas, J.: 2017, Quantifying the informativeness of similarity measurements. *J. Mach. Learn. Res.* **18**, 1.

Chen, C., Shen, L., Zou, F., Liu, W.: 2022, Towards practical Adam: non-convexity, convergence theory, and mini-batch acceleration. *J. Mach. Learn. Res.* **23**, 1.

Chiu, Y., Hilton, H.: 1977, Exact Green's function method of solar force-free magnetic-field computations with constant alpha. I. theory and basic test cases. *Astrophys. J.* **212**, 873. DOI. ADS.

Cuomo, S., Cola, V.D., Giampaolo, F., Rozza, G., Raissi, M., Piccialli, F.: 2022, Scientific machine learning through physics–informed neural networks: where we are and what's next. *J. Sci. Comput.* **92**, 88. DOI.

Cybenko, G.: 1989, Approximation by superpositions of a sigmoidal function. *Math. Control Signals Syst.* **2**, 303. DOI.

Davis, H., Snider, A., Davis, C.: 1979, *Introduction to Vector Analysis*, Allyn & Bacon, London.

Démoulin, P., Mandrini, C., Driel-Gesztelyi, L., Thompson, B., Plunkett, S., Kővári, Z., Aulanier, G., Young, A.: 2002, What is the source of the magnetic helicity shed by CMEs? The long-term helicity budget of AR 7978. *Astron. Astrophys.* **382**, 650. DOI. ADS.

Dufera, T.: 2021, Deep neural network for system of ordinary differential equations: vectorized algorithm and simulation. *Mach. Learn. Appl.* **5**, 100058. DOI.

Grad, H., Rubin, H.: 1958, Hydromagnetic equilibria and force-free fields. *J. Nucl. Energy* **7**, 284. DOI.

Hornik, K.: 1991, Approximation capabilities of multilayer feedforward networks. *Neural Netw.* **4**, 251. DOI.

Kincaid, D., Cheney, E.: 2002, *Numerical Analysis: Mathematics of Scientific Computing*, Am. Math. Soc., Providence.

Lerche, I., Low, B.: 2014, A nonlinear eigenvalue problem for self-similar spherical force-free magnetic fields. *Phys. Plasmas* **21**, 81. DOI. ADS.

Liu, S., Su, J., Zhang, H., Deng, Y., Gao, Y., Yang, X., Mao, X.: 2013, A statistical study on force-freeness of solar magnetic fields in the photosphere. *Proc. Astron. Soc. Austral.* **30**, e005. DOI. ADS.

Low, B., Flyer, N.: 2007, The topological nature of boundary value problems for force-free magnetic fields. *Astrophys. J.* **668**, 557. DOI. ADS.

Low, B., Lou, Y.: 1990, Modeling solar force-free magnetic fields. *Astrophys. J.* **352**, 343. DOI. ADS.

Mikić, Z., McClymont, A.: 1994, Deducing coronal magnetic fields from vector magnetograms. In: Balasubramaniam, K., Simon, G. (eds.) *Solar Active Region Evolution: Comparing Models with Observations* **CS-68**, Astron. Soc. Pacific, San Francisco, 225. ADS.

Moon, Y., Choe, G., Yun, H., Park, Y., Mickey, D.: 2002, Force-freeness of solar magnetic fields in the photosphere. *Astrophys. J.* **568**, 422. DOI. ADS.

Nakagawa, Y.: 1974, Dynamics of the solar magnetic field. I. Method of examination of force-free magnetic fields. *Astrophys. J.* **190**, 437. DOI. ADS.

Nakagawa, Y., Raadu, M.: 1972, On the practical representation of magnetic field. *Solar Phys.* **25**, 127. DOI. ADS.
Neukirch, T.: 2005, Magnetic field extrapolation. In: Innes, D., Lagg, A., Solanki, S. (eds.) *Chromospheric and Coronal Magnetic Fields* **SP-596**, ESA, Noordwijk, 12.1. ADS.
Prasad, A., Mangalam, A., Ravindra, B.: 2014, Separable solutions of force-free spheres and applications to solar active regions. *Astrophys. J.* **786**, 102902. DOI. ADS.
Raissi, M., Perdikaris, P., Karniadakis, G.: 2019, Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **378**, 686. DOI. ADS.
Régnier, S., Amari, T., Kersalé, E.: 2002, 3D coronal magnetic field from vector magnetograms: non-constant-alpha force-free configuration of the active region NOAA 8151. *Astron. Astrophys.* **392**, 1119. DOI. ADS.
Sakurai, T.: 1989, Computational modeling of magnetic fields in solar active regions. *Space Sci. Rev.* **51**, 11. DOI. ADS.
Schmidt, H.: 1964, On the observable effects of magnetic energy storage and release connected with solar flares. In: Hess, W. (ed.) *Proc. AAS-NASA Symp.*, NASA, Washington, 107. ADS.
Tolstykh, V.A.: 2020, *Partial Differential Equations: An Unhurried Introduction*, de Gruyter, Berlin.
Valori, G., Kliem, B., Török, T., Titov, V.: 2010, Testing magnetofrictional extrapolation with the Titov-Démoulin model of solar active regions. *Astron. Astrophys.* **519**, A44. DOI. ADS.
Wheatland, M., Sturrock, P., Roumeliotis, G.: 2000, An optimization approach to reconstructing force-free fields. *Astrophys. J.* **540**, 1150. DOI. ADS.
Wiegelmann, T., Inhester, B., Sakurai, T.: 2006, Preprocessing of vector magnetograph data for a nonlinear force-free magnetic field reconstruction. *Solar Phys.* **233**, 215. DOI. ADS.
Wiegelmann, T., Neukirch, T.: 2003, Computing nonlinear force free coronal magnetic fields. *Nonlinear Process. Geophys.* **10**, 313. DOI. ADS.
Wiegelmann, T., Sakurai, T.: 2021, Solar force-free magnetic fields. *Liv. Rev. Solar Phys.* **18**, 1. DOI. ADS.
Yan, Y., Li, Z.: 2006, Direct boundary integral formulation for solar non-constant-α force-free magnetic fields. *Astrophys. J.* **638**, 1162. DOI. ADS.
Yan, Y., Sakurai, T.: 2000, New boundary integral equation representation for finite energy force-free magnetic fields in open space above the sun. *Solar Phys.* **195**, 89. DOI. ADS.