Estimation of Bicycle Dynamic Model Parameters with Physics-Informed Neural Networks for Autonomous Racing

Hassan Jardali, Andrew Meighan, Dilip Nikhil Francies
Intelligent Systems Engineering, Indiana University, Bloomington, IN, USA
{hjardali, ameighan, dfranci}@iu.edu

***Abstract*—Autonomous racing represents a rapidly emerging field, encompassing platforms from scaled-down competitions like F1Tenth to full-scale challenges such as the Indy Autonomous Challenge (IAC). Ensuring vehicle controllability at high speeds exceeding 125 mph (201 km/h) necessitates highly accurate dynamic modeling to enable robust planning and control. These dynamic models depend on parameters that are not directly available and must be inferred from observational data. In this study, we present a Physics-Informed Neural Network (PINN) framework designed to extract these parameters using data compiled from the IU-Luddy team at the IAC. The proposed approach is evaluated through a comprehensive analysis of multiple PINN variations. Our findings demonstrate the efficacy of PINNs in achieving precise dynamic parameter estimation validated on our dataset, with the aim to use these parameters with a model-based controller at CES 2025.**

***Index Terms*—Autonomous Racing, Dynamics Modeling, Physics-Informed Neural Networks (PINNs), Pacejka Model, Deep Learning.**

I. Introduction

The Indy Autonomous Challenge (IAC) has highlighted the critical importance of understanding vehicle dynamics in the context of high-speed autonomous racing. Accurate dynamic modeling is essential for achieving optimal vehicle control and performance. Traditional approaches rely on classical dynamic models, but recent advancements in machine learning, particularly Physics-Informed Neural Networks (PINNs), offer new avenues for improving prediction accuracy and adaptability.

At the high velocities achieved in autonomous racing, vehicle dynamics become highly nonlinear and sensitive to parameter variations. Accurate handling in these conditions often necessitates the use of advanced control strategies like Model Predictive Control (MPC). MPC is a finite-horizon model-based controller that predicts future poses of the robot to optimize the control sequence over a prespecifed time horizon. It has been shown in multiple literature reviews that MPC outperforms all other controllers in trajectory tracking control problems for autonomous vehicles scenarios [1].

Dynamic models used in MPC can be categorized into kinematic and dynamic models. Kinematic models are simpler and capture the geometric aspects of motion without accounting for forces and mass distribution. This makes them inadequate for high-speed applications where inertia and tire forces play significant roles. Dynamic models, on the other hand, consider the forces and moments acting on the vehicle, providing a more accurate representation of its behavior.

These dynamic models heavily rely on parameters from tire models, such as the Pacejka "Magic Formula," and drivetrain characteristics. The Pacejka model is a widely used empirical formula that describes the nonlinear behavior of tire forces as a function of slip angles, loads, and other factors. Accurately identifying these parameters typically requires extensive experimental testing, which may be impractical due to limited access to testing facilities, high costs, and time limitations [2].

In this study, we address these challenges by employing Physics-Informed Neural Networks (PINNs) to estimate the dynamic model parameters from observational data. PINNs integrate physical laws described by differential equations into the training of neural networks, enabling us to infer parameters without exhaustive testing. Specifically, we combine PINNs with recurrent neural networks (RNNs). RNNs transform the input of sequential data into sequential outputs so that the temporal component of the problem is retained. In this work, both the less computationally intensive Gated Recurrent Units (GRUs) [3] and Long Short-Term Memory (LSTM) cells [4] are used to estimate parameters and predict control outputs. Because of the high speed racing done with the vehicles generating the data, we believe the combination of long and short-term memory cells in the GRU architecture will not adversely affect performance.

In this work, we mostly rely on these two papers [5], [6]. The methods and code discussed in Chrosniak's paper [5] are open-source, thus we base our code structure around theirs while replacing their data with data generated from the IU-Luddy AV24 car at the Indianapolis Motor Speedway in September 2024.

II. Related Work

A. Tire Parameter Estimation Through Learning

Recent advancements in machine learning have facilitated the estimation of tire parameters through data-driven approaches. The Deep Pacejka model, for instance, utilizes neural networks to approximate tire dynamics, providing flexibility under diverse conditions and reducing reliance on extensive physical testing. Other studies have applied deep learning techniques to capture the complex, nonlinear behavior of vehicle dynamics [2], [7]. [6] proposes a model that integrates deep learning with a fully differentiable physics framework to incorporate prior knowledge, achieving significantly improved generalization performance and accurately representing lateral tire forces without additional training. Lastly, [5], which is more aligned with our work, utilized Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks in combination with a Physics-Informed Neural Network (PINN) to estimate the parameters of the Pacejka tire model and drivetrain models.

B. Physics-Informed Neural Networks (PINNs)

Physics-Informed Neural Networks (PINNs) have emerged as a powerful framework that integrates data-driven machine learning models with underlying physical principles, ensuring that learned relationships conform to the governing equations of the system [8]. This approach has proven effective in capturing complex dynamics, as it directly incorporates physics-based constraints into the loss function, thereby enabling the network to generalize beyond the available training data. Concurrently, advancements in Recurrent Neural Networks (RNNs) have improved the modeling of sequence data through specialized architectures such as Long Short-Term Memory (LSTM) [4] and Gated Recurrent Units (GRUs) [3]. These recurrent variants mitigate the vanishing gradient problem inherent to standard RNNs, allowing for more stable and accurate long-term temporal dependencies. By combining the physics-informed loss structure of PINNs with the robust temporal modeling capabilities of LSTMs and GRUs, one can design hybrid models that more accurately capture the evolution of dynamic systems. Such integrated approaches ensure that the learned representations remain consistent with the underlying physics while effectively handling complex temporal correlations, ultimately yielding models better suited for tasks like dynamic parameter estimation and control in high-speed autonomous racing scenarios.

III. Methodology

In this section, we describe the dynamic model used for the vehicle, the Pacejka tire model, and the drivetrain model. We also discuss the integration of Physics-Informed Neural Networks (PINNs) for parameter estimation.

A. Dynamic Model and Tire Pacejka Model

1) Model Predictive Control (MPC): Model Predictive Control (MPC) is an advanced control strategy that uses an explicit dynamic model of the system to predict future states and optimize control inputs over a finite horizon. MPC can handle multi-input, multi-output systems with constraints, making it suitable for high-speed autonomous vehicles where accurate trajectory tracking and stability are critical.

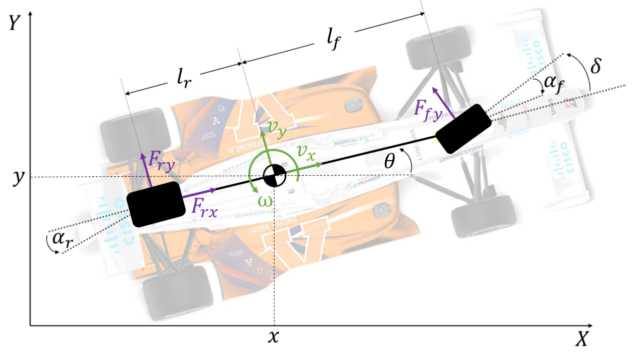


Fig. 1: The single-track dynamic model representation. Diagram from [5].

2) Vehicle Dynamics Model: We employ a bicycle dynamic model to represent the vehicle's motion. This model simplifies the vehicle by combining the left and right wheels into a single front wheel and a single rear wheel, capturing the essential dynamics while reducing complexity. For a representation, look at Fig. 1.

The equations of motion are given by:

$$m(\dot{v}_x - v_y\omega) = F_{xf}\cos\delta + F_{xr} - F_{yf}\sin\delta, \tag{1}$$

$$m(\dot{v}_y + v_x\omega) = F_{yf}\cos\delta + F_{yr} + F_{xf}\sin\delta, \tag{2}$$

$$I_z\dot{\omega} = l_f(F_{yf}\cos\delta + F_{xf}\sin\delta) - l_r F_{yr}, \tag{3}$$

where:

- m is the mass of the vehicle,
- v_x and v_y are the longitudinal and lateral velocities,
- ω is the yaw rate,
- F_{xf} and F_{xr} are the longitudinal forces at the front and rear tires,
- F_{yf} and F_{yr} are the lateral forces at the front and rear tires,
- δ is the steering angle,
- l_f and l_r are the distances from the center of gravity to the front and rear axles,
- I_z is the yaw moment of inertia.

3) Pacejka Tire Model: The Pacejka "Magic Formula" model is used to calculate tire forces based on slip angles and slip ratios. The lateral tire force F_y is given by:

$$F_y = D\sin\left(C\arctan\left(B\alpha - E\left(B\alpha - \arctan(B\alpha)\right)\right)\right) + Shy, \tag{4}$$

where:

- α is the slip angle,
- B, C, D, E, and Shy are empirical parameters defining the tire characteristics.

4) Drivetrain Model: The drivetrain model which drives the car in the longitudinal direction can be expressed as:

$$F_{rx} = (C_{m1} - C_{m2}v_x)\,T - C_{r0} - C_d v_x^2 \tag{5}$$

This equation models the longitudinal force, F_{rx}, generated by the drivetrain system of a vehicle. Here, C_{m1} and C_{m2} are coefficients representing the motor torque characteristics, with $C_{m2}v_x$ accounting for the speed-dependent torque reduction. The term T denotes the motor torque input. The rolling resistance, C_{r0}, and the aerodynamic drag, represented by $C_d v_x^2$, are subtracted to provide a complete description of the resistive forces acting against the vehicle motion. The velocity, v_x, plays a crucial role in scaling these resistive components, particularly the quadratic dependency associated with the aerodynamic drag.

5) Integration with Control: These parameters are critical for the vehicle's control system, especially in MPC, where accurate predictions of the vehicle's future states depend on precise dynamic models. The inputs to the model include steering angle and throttle/brake commands, while the outputs are the vehicle's velocities and yaw rate.

B. PINN Details

Our approach leverages PINNs to estimate the unknown parameters in the dynamic model. By incorporating the differential equations governing vehicle dynamics into the neural network's loss function, we enforce physical consistency.

The PINN architecture consists of input layers representing observable variables (e.g., velocities, yaw rate), hidden layers with activation functions, and output layers corresponding to the parameters to be estimated.

The loss function L combines data loss and physics loss:

$$L = L_{\text{data}} + \lambda L_{\text{physics}}, \tag{6}$$

where:

- L_{data} measures the discrepancy between the network's predictions and the observed data,
- L_{physics} ensures that the predicted parameters satisfy the dynamic equations,
- λ is a weighting factor balancing the two losses.

We train the PINN using mini-batch Adam optimizer, adjusting the network weights to minimize the total loss.

The PINN also includes a physics-guard layer first used in [5]. This layer ensures parameters cannot exit outside of realistic real world bounds. After the network predicts the parameters, the physics-guard puts them through a sigmoid function to get values between zero and one. This transformation can be reversed given the user-generated minimum and maximum values of the parameter. Once the transformation is reversed, the model uses the parameters in the dynamic and tire models used to calculate the vehicle's next state. All of this setup assumes that data can accurately represent real-world, high velocity scenarios. This data, along with methods used to train the data, are discussed in the following section.

IV. Data and Experiment Set Up

A. Data

The IU Luddy team is currently participating in the Indy Autonomous Challenge (IAC), with activities commencing in August 2024. Our autonomous vehicle, the AV24, is based on the design of the DALLARA IL-15 Indy Car. Notably, the AV24 has no traditional cockpit; instead, it integrates a high-performance onboard computer connected to an array of sensors, including GPS, IMU (Inertial Measurement Unit), wheel odometry sensors, LiDARs, cameras, and radars. These sensors provide critical environmental and operational data for vehicle control. The primary control inputs for the AV24 are as follows: steering for lateral control, and throttle, brake, and gear commands for longitudinal control.

To facilitate model training, we utilized data from our final performance session, which achieved the highest vehicle speed. A video of this performance can be accessed here: https://youtu.be/UitZRC3KpNE.

The training data was extracted from ROS (Robot Operating System) bag files and subsequently converted into CSV format for analysis. The headers of the resulting CSV files are as follows:

TABLE I: Data

Data Point	**Description**
time	Timestamp (s)
wheel_speed_rl	Rear-left wheel speed (m/s)
wheel_speed_fr	Front-right wheel speed (m/s)
wheel_speed_fl	Front-left wheel speed (m/s)
wheel_speed_rr	Rear-right wheel speed (m/s)
position_x	X-coordinate of position (m)
position_y	Y-coordinate of position (m)
rpy_z	Yaw angle (in rad)
velocity_x_local_filt	Long. velocity filtered (m/s))
velocity_y_local_filt	Lateral velocity filtered (m/s)
angular_velocity_z_filt	Angular velocity about the Z-axis (rad/s)
cmd_gear	Commanded gear state (1 to 6)
cmd_throttle	Commanded throttle input (in %)
cmd_brake	Commanded brake input (Kpa)
cmd_steering_degree	Commanded steering angle in degrees

It is important to note that the velocity and acceleration data contained noise due to engine-induced vibrations, especially at high RPMs. These vibrations adversely affected sensor measurements, necessitating preprocessing and filtering steps to mitigate their impact on model accuracy.

B. Training

The models are trained on locally available hardware, leveraging optimizers such as Adam and adaptive learning rate schedules. Throughout training, we closely monitor metrics such as loss curves and convergence behaviors to ensure that the model remains stable and reliable. Prior to full-scale training, key hyperparameters—including learning rate, number of layers, and latent dimensionality—are tuned using an Optuna study. This automated optimization process identifies hyperparameter configurations that minimize validation loss over a subset of the training epochs,

Fig. 2: IU-Luddy AV24 car at the Indianapolis Motor Speedway, September 2024. More info about the car can be found at indyautonomouschallenge.com/racecar

ensuring a more efficient search compared to manual trial and error.

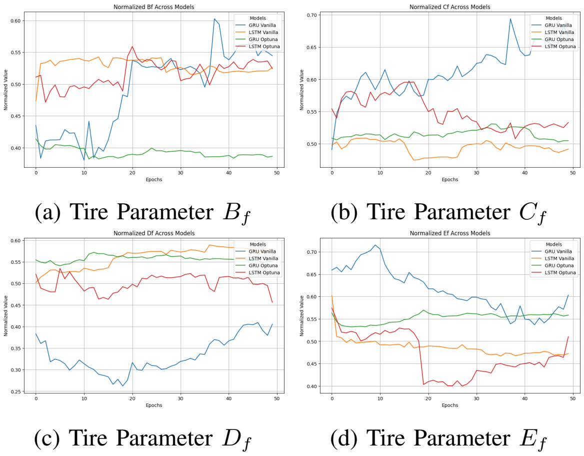
Notably, the LSTM and GRU architectures may require distinct hyperparameter choices due to their different internal gating mechanisms. For example, LSTMs often benefit from slightly larger hidden states and more layers to capture long-term dependencies, while GRUs can achieve comparable performance with fewer parameters and thus may require a smaller latent size or fewer layers. These considerations are taken into account during the hyperparameter tuning phase, where the search space for each architecture is adapted accordingly.

After arriving at the tuned hyperparameters, we train the selected models to completion, ultimately assessing their performance on our dataset. The results demonstrate that appropriate hyperparameter selection, informed by systematic optimization and architecture-specific considerations, leads to improved convergence and more accurate model predictions as shown in the results section.

V. Results

A number of parameters in the Pacejka and drivetrain models are iteratively updated during training. In Fig. 3, we can see that while certain models develop unique solutions for certain parameters, on average the models converge within a band of values between five and ten-percent apart. On top of this, the models still must update to reduce the total loss so even with unique parameters, other parameters must adjust to account for the unique solutions. This can also be seen in Fig. 4c in which the Vanilla GRU and the Optuna LSTM converge to one value of C_{m1}, while the Vanilla LSTM and the Optuna GRU converge to another.

After comparing the models, we used the data from our final run to calculate the slip angle. We then plotted the estimated lateral force, determined using Newton's second law (F = ma), against the slip angle. The data was filtered using low-pass filters to reduce noise. Finally, we plotted the results from the four models to evaluate the accuracy of the learned parameters, referencing Eq. 4 for validation.



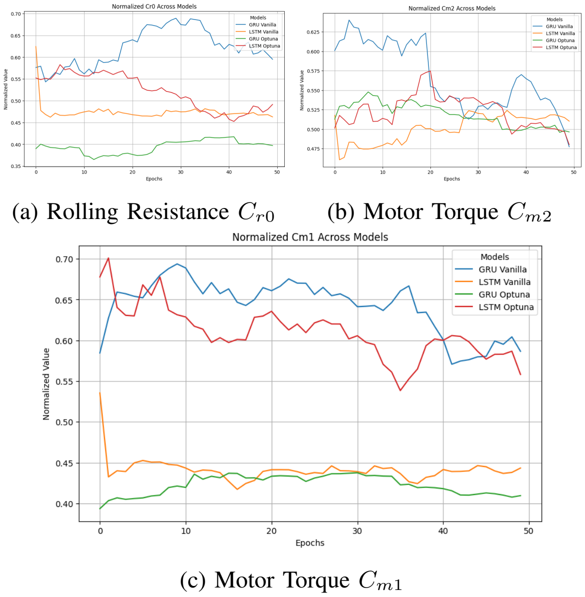
(a) Tire Parameter B_f

(b) Tire Parameter C_f

(c) Tire Parameter D_f

(d) Tire Parameter E_f

Fig. 3: Updates to normalized parameters defining the tire characteristics for each model over 50 epochs.



(a) Rolling Resistance C_{r0}

(b) Motor Torque C_{m2}

(c) Motor Torque C_{m1}

Fig. 4: Updates to normalized drivetrain parameters for each model over 50 epochs.

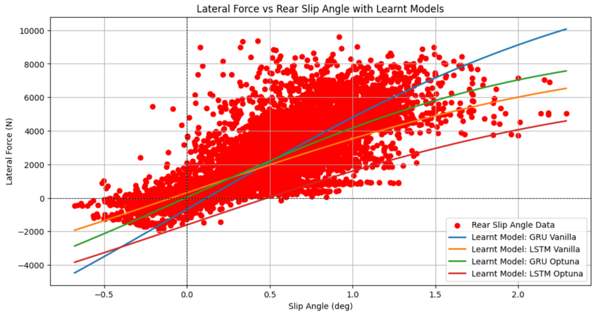


Fig. 5: Rear Wheel Lateral Force vs Slip Angle results

Parameter	GRU Vanilla	LSTM Vanilla	GRU Optuna	LSTM Optuna	Min	Max
Bf	14.6242	15.2480	13.7938	16.1558	5.0	25.0
Cf	1.7281	1.4905	1.4853	1.4856	1.0	2.0
Df	8252.9141	11663.4951	10004.1240	9073.7061	100.0	20000.0
Ef	-0.3512	-0.4920	-0.4333	-0.4800	-1.0	0.0
Br	14.9158	16.1032	19.5146	16.9719	5.0	25.0
Cr	1.5306	1.4931	1.4469	1.3589	1.0	2.0
Dr	14306.9414	8018.0210	8928.7129	8326.0938	100.0	20000.0
Er	-0.4167	-0.4704	-0.4582	-0.4126	-1.0	0.0
Cm1	5408.4966	4662.4512	3788.8503	6082.6900	100.0	10000.0
Cm2	0.4745	0.5282	0.4952	0.4491	0.0	1.0
Cr0	0.7451	0.7103	0.6127	0.6729	0.1	1.4
Cr2	0.4455	0.5387	0.6093	0.4240	0.1	1.0
Iz	1190.6558	1242.6953	1365.8519	1189.9443	500.0	2000.0
Shf	-0.0057	-0.0016	-0.0036	0.0007	-0.02	0.02
Svf	-395.6201	-418.9043	-1049.2063	238.7603	-5000.0	5000.0
Shr	-0.0041	0.0032	-0.0052	-0.0017	-0.02	0.02
Svr	-679.5713	308.5894	32.2734	-1611.9192	-5000.0	5000.0

TABLE II: Learnt parameters for GRU and LSTM models (Vanilla and Optuna). The Min and Max columns represent the constraints applied to each parameter in the Guard Layer. The Min-Max values are the taken from [5]
.

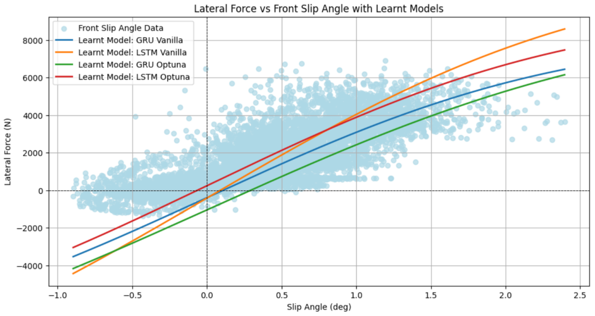


Fig. 6: Front Wheel Lateral Force vs Slip Angle results

Fig 5 illustrates the relationship between lateral force and slip angle, showcasing the performance of four different models: GRU Vanilla, LSTM Vanilla, GRU Optuna, and LSTM Optuna. The raw data, represented by red scatter points, exhibits significant variability but demonstrates a positive correlation between slip angle and lateral force. The Optuna-tuned models (GRU Optuna and LSTM Optuna) align more closely with the observed data, particularly in the mid-range slip angles, indicating superior parameter optimization and model learning. In contrast, the GRU Vanilla model overestimates the lateral force at higher slip angles, while the LSTM Vanilla model exhibits a less steep slope, diverging from the trend.

For the front wheel showed in Fig. 6, the LSTM Vanilla (orange) and LSTM Optuna (red) exhibit higher slopes, suggesting an overestimation of lateral force at larger slip angles. In contrast, the GRU Optuna (green) and GRU Vanilla (blue) display more moderate trends, with GRU Optuna closely aligning with the observed data in the mid-range slip angles.

The analysis of front and rear slip angle data shows that Optuna-tuned models (GRU Optuna and LSTM Optuna) consistently outperform vanilla models in capturing the nonlinear relationship between slip angle and lateral force. This makes sense as the parameters chosen for the Optuna models begin with hyperparameters that result in much lower initial loss than the other models. GRU Optuna provides the most balanced performance for both datasets, while LSTM Optuna tends to overestimate lateral forces at higher slip angles, especially for the front. The rear data exhibits greater variability, indicating higher complexity in its dynamics. Overall, hyperparameter optimization significantly enhances model accuracy, with GRU Optuna emerging as the most reliable across varying conditions.

VI. Conclusion

In this work, we proposed a Physics-Informed Neural Network (PINN) framework to estimate the dynamic parameters of an Indy car, enabling accurate modeling for high-speed autonomous racing scenarios. We presented a comparative analysis of four PINN variants, leveraging LSTM and GRU architectures with optimized hyperparameters tailored to each model's unique requirements. Our results demonstrate that careful tuning of hyperparameters significantly improves alignment between the learned parameters and the observed data, validating the effectiveness of the proposed approach. On top of this, the high speed racing performed by the vehicle collecting data negates the need for a long-term memory cell which is combined with the short-term cell in GRUs.

For future work, we plan to integrate these estimated parameters into a Model Predictive Controller (MPC) to enhance vehicle control and performance. Additionally, we aim to explore online parameter learning, allowing for real-time updates and corrections to the MPC during operation, further advancing the capabilities of autonomous racing systems.

REFERENCES

[1] P. Banginwar and T. Sands, "Autonomous vehicle control comparison," *Vehicles*, vol. 4, no. 4, 2022.

[2] J. L. Olazagoitia, J. A. Perez, and F. Badea, "Identification of tire model parameters with artificial neural networks," *Applied Sciences*, vol. 10, no. 24, p. 9110, 2020.

[3] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[4] S. Hochreiter, "Long short-term memory," *Neural Computation MIT-Press*, 1997.

[5] J. Chrosniak, J. Ning, and M. Behl, "Deep dynamics: Vehicle dynamics modeling with a physics-constrained neural network for autonomous racing," *IEEE Robotics and Automation Letters*, 2024.

[6] T. Kim, H. Lee, and W. Lee, "Physics embedded neural network vehicle model and applications in risk-aware autonomous driving using latent features," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 4182–4189.

[7] M. Acosta and S. Kanarachos, "Tire lateral force estimation and grip potential identification using neural networks, extended kalman filter, and recursive least squares," *Neural Computing and Applications*, vol. 30, pp. 3445–3465, 2018.

[8] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational physics*, vol. 378, pp. 686–707, 2019.