



# Optimal Parameter Estimation Techniques for Complex Nonlinear Systems

Kaushal Kumar<sup>1</sup> · Ekaterina Kostina<sup>1</sup>

Accepted: 27 February 2024  
© The Author(s) 2024

## Abstract

Accurate parameter estimation and state identification within nonlinear systems are fundamental challenges addressed by optimization techniques. This paper fills a critical gap in previous research by investigating tailored optimization methods for parameter estimation in nonlinear system modeling, with a particular emphasis on chaotic dynamical systems. We introduce and compare three optimization methods: a gradient-based iterative algorithm, the Levenberg-Marquardt algorithm, and the Nelder-Mead simplex method. These methods are strategically employed to simplify complex nonlinear optimization problems, rendering them more manageable. Through a comprehensive exploration of the performance of these methods in determining parameters across diverse systems, including the van der Pol oscillator, the Rössler system, and pharmacokinetic modeling, our study revealed that the accuracy and reliability of the Nelder-Mead simplex method were consistent. The Nelder-Mead simplex algorithm emerged as a powerful tool, that consistently outperforms alternative methods in terms of root mean squared error (RMSE) and convergence reliability. Visualizations of trajectory comparisons and parameter convergence under various noise levels further emphasize the algorithm's robustness. These studies suggest that the Nelder-Mead simplex method has potential as a valuable tool for parameter estimation in chaotic dynamical systems. Our study's implications extend beyond theoretical considerations, offering promising insights for parameter estimation techniques in diverse scientific fields reliant on nonlinear system modeling.

**Keywords** Parameter estimation · Nonlinear dynamics · Gradient-based iteration · Levenberg-Marquardt algorithm · Nelder-Mead simplex method · Pharmacokinetic modeling

---

✉ Kaushal Kumar  
kaushal.kumar@stud.uni-heidelberg.de

<sup>1</sup> Institute for Mathematics, Heidelberg University, Im Neuenheimer Feld 205, 69120 Heidelberg, Germany

## Introduction

Understanding complex systems often requires the construction of mathematical models based on empirical data—a process known as system identification [1–3]. This paper addresses the intricate challenge of accurately estimating parameters for nonlinear models that encapsulate the dynamics of these systems, such as the van der Pol oscillator, the Rössler system, and pharmacokinetic models. Parameter estimation is paramount for gaining insights into the behaviors of real-world systems and involves optimizing a cost function through various techniques such as gradient methods, Newton’s methods, and least square methods [4].

To address this challenge, we delve into optimization methods designed to simplify the intricate task of parameter estimation. Specifically, our study explores three distinct optimization strategies: a gradient-based iterative algorithm, the Levenberg-Marquardt algorithm, and the Nelder-Mead simplex method. These methods are applied to estimate the parameters of nonlinear functions, and their effectiveness is demonstrated through practical examples involving the aforementioned models.

This research contributes to the advancement of optimization techniques for parameter estimation in nonlinear system modeling, with broad applications across diverse scientific and engineering domains. The remainder of this paper is structured as follows: Sect. 3 outlines the research objectives, Sect. 4 provides a detailed description of the methods and algorithms employed for parameter identification in nonlinear dynamical systems, and Sects. 5 and 6, present the results of our experiments and conclusions, respectively.

## Related Works

Fitting model parameters to real-world data is a common yet challenging task [5], particularly when dealing with models containing numerous parameters. Algorithms often struggle in parts of the parameter space where the model does not respond to changes, requiring manual adjustments.

Optimization methods are crucial across various fields, providing solutions to complex problems. Here, we explore different optimization approaches and their relevance to our study. Specifically, we focus on three key algorithms: a gradient-based iterative method, the Levenberg-Marquardt algorithm, and the Nelder-Mead simplex method [6–9].

Gavin [10] has discussed these methods, demonstrating their application in solving curve-fitting problems using software tools. Shawash [11] demonstrated a practical implementation of the LM algorithm on hardware, specifically field programmable gate arrays (FPGAs). This successful real-time camera calibration and parameter estimation on FPGAs provides a blueprint for implementing the LM algorithm on specialized hardware for high-speed, low-power applications.

The Nelder-Mead simplex method, a derivative-free approach for function minimization, involves evaluating the objective function at various simplex vertices. Olsson [12] explained this method’s movement away from the poorest value. Wang [20] conducted a parameter sensitivity study of this method across different functions, revealing essential relationships between the parameters and the optimal solution.

Additionally, numerous methods exist for solving similar problems, including trust-region optimization [14], multiple shooting methods [15, 16], and data-driven approaches [17, 18] leveraging various machine learning algorithms [19].

## Theory: Background and Preliminaries

The challenge of parameter estimation in ordinary differential equations (ODEs) involves determining a parameter vector  $\mathbf{p} \in \mathbb{R}^k$  and a trajectory  $y : [t_a, t_b] \rightarrow \mathbb{R}^n$  such that the ODE  $\dot{y} = f(t, y, p)$  is satisfied while minimizing a least-squares functional given by

$$L(y, p) = \|r(y(t_1), \dots, y(t_\mu), p)\|_2, \quad (1)$$

where  $r \in \mathbb{R}^l$  represents least-squares conditions based on specific instants  $t_i$  and parameters.

In a common scenario, the objective function simplifies to,

$$L(y, p) = \sum_{i=1}^N \sum_{j=1}^v \frac{1}{\sigma_{ij}^2} [\eta_{ij} - g_i(t_i, y(t_i), p)]^2 \quad (2)$$

where,  $g$  relates ODE components to measured quantities,  $\eta_{ij}$  is the observed value of  $g_i$  at instant  $t_i$ , and  $\sigma_{ij}$  is the standard deviation.

Estimating parameters, as described above, may present challenges when approached directly. However, adopting an inverse perspective shows promise. We consider the initial value problems  $\dot{y}(t) = f(t, y, p)$ ,  $y(t_0) = y_0$ , and their solutions' derivatives with respect to the parameters and initial values:

$$\begin{aligned} \frac{\partial}{\partial y_0} y(t; t_0, y_0, p) &= G(t, t_0, y_0, p) \\ \frac{\partial}{\partial p} y(t; t_0, y_0, p) &= H(t, t_0, y_0, p) \end{aligned} \quad (3)$$

The following equations for the difference in variation yield insightful relationships:

$$\begin{aligned} \dot{G}(t, t_0, y_0, p) &= f_y(t, y, p) \cdot G(t, t_0, y_0, p) \\ G(t_0, t_0, y_0, p) &= 1 \\ \dot{H}(t, t_0, y_0, p) &= f_y(t, y, p) \cdot H(t, t_0, y_0, p) + f_p(t, y, p) \\ H(t_0, t_0, y_0, p) &= 0 \end{aligned} \quad (4)$$

This leads to the deduction that:

$$H(t, t_0, y_0, p) = \int_{t_0}^t G(t, s, y_0, p) \cdot \frac{\partial f}{\partial p}(s, y(s, t_0, y_0, p), p) ds \quad (5)$$

This implies that if the time scale of measurement is significantly smaller than the time scale of oscillations in  $G$  or  $f_p$ , parameter derivatives might benefit from initial value derivatives, especially in chaotic systems. Essentially, observed trajectories could contain substantial information about the parameters. Hence, parameter estimation in chaotic systems could be well-posed if error propagation is managed effectively [9].

## Methods

Experimental data arising from chaotic nonlinear dynamical systems commonly undergo evaluation using diverse time-series analysis techniques. To address the optimization challenge inherent in this scenario, a variety of optimization algorithms are employed to identify the parameter vector  $\mathbf{p}$  and the trajectory  $x$  that minimizes the objective function.

In this context, we introduce and employ three distinct optimization methods:

### The Gradient-based Iterative Algorithm

An algorithm based on gradient estimation can be used to find iterative solutions of  $\mathbf{p}$  through the gradient search principle [7, 20]. In this algorithm, let  $k$  be the iterative variable, while  $\hat{p}_k$  represents the iterative estimates of  $p$  at iteration  $k$ . The objective is to create identification techniques for estimating the parameters  $p$  by utilizing the available measurement data  $(x_i; f(x_i))$ , which is equivalent to minimizing the following cost function

$$J(p) = \sum_{ij} \frac{1}{\sigma_{ij}^2} [\eta_{ij} - g_i(x(t_j), p)]^2. \quad (6)$$

and the gradient search principle leads to the following gradient-based estimation algorithms

$$[\hat{p}_k] = [\hat{p}_{k-1}] - \mu_k \left[ \frac{\partial J(x, p)}{\partial p} \right] \quad (7)$$

where  $\mu_k > 0$  is the step-size or convergence factor and determined by

$$\mu_k = \arg \min_{\mu \geq 0} J(\hat{p}_{1,k-1} - \mu \frac{\partial J}{\partial p_1}, \hat{p}_{2,k-1} - \mu \frac{\partial J}{\partial p_2}, \dots) \quad (8)$$

The procedure for computing the gradient-based estimation algorithm for the system described in equation (1) can be summarized as follows:

1. The measured data  $(x_i, f(x_i))$ , are collected such that  $i = 1, 2, \dots, N$ .
2. To initialize, let  $k = 1$ ,  $\hat{p}_0$  be arbitrary real numbers, and the preset small  $\varepsilon$
3. Determine the step-size  $\mu_k$  by (8).
4. Compute  $\hat{p}_k$  by (7).
5. If  $\sum \|\hat{p}_k - \hat{p}_{k-1}\| > \varepsilon$ , increase  $k$  by 1 and go to step 3; otherwise, terminate the procedure and obtain the estimate  $\hat{p}_k$ .

The learning rate also needs to be chosen carefully to ensure that the algorithm converges efficiently without oscillating or overshooting the minimum.

### The Levenberg-Marquardt Algorithm

The Levenberg-Marquardt algorithm [21, 22] iteratively updates the parameter estimates using a combination of the Gauss-Newton method and the steepest descent method, with

a damping parameter that adjusts the step size based on the curvature of the cost function. This algorithm 1 is widely used for nonlinear least squares problems and can be applied to a variety of parameter estimation problems in different fields [23, 24].

**Algorithm 1** Pseudocode of the standard Levenberg-Marquardt algorithm for parameter estimation problems

---

**Require:** Initial parameter estimates  $p_0$ , damping parameter  $\lambda_0 > 0$ , and tolerance  $\epsilon > 0$ .

**Ensure:** Best estimate of the parameters  $p_0$ .

- 1: Collect the measured data  $(x_i, f(x_i))$  for  $i = 1, 2, \dots, N$ .
  - 2: Define the residual function  $e(p) = [f(x_1) - y_1, f(x_2) - y_2, \dots, f(x_N) - y_N]$ , where  $y_1, y_2, \dots, y_N$  are the measured response values.
  - 3: Calculate the Jacobian matrix  $J(p) = [\frac{\partial f}{\partial p_1}, \frac{\partial f}{\partial p_2}, \dots, \frac{\partial f}{\partial p_n}]$ , evaluated at the current parameter  $p$ .
  - 4: Initialize  $p = p_0$ ,  $\lambda = \lambda_0$ , and  $J(p)$ .
  - 5: **while**  $\Delta p > \epsilon$  **do**
  - 6:   Compute the Gauss-Newton step,  $dp_{GN}$ , by solving the linear system of equations:  $[J(p)^T J(p) + \lambda I] dp_{GN} = -J(p)^T e(p)$ , where  $I$  is the identity matrix and  $T$  denotes the matrix transpose.
  - 7:   Calculate the trial parameter estimates  $p_{trial} = p + dp_{GN}$ .
  - 8:   Compute the residual vector  $e(p_{trial})$  and evaluate the corresponding cost function  $J(p_{trial})$ .
  - 9:   **if**  $J(p_{trial}) < J(p)$  **then**
  - 10:     Update the parameter estimates  $p = p_{trial}$ , reduce the damping parameter  $\lambda$  by a factor of 10, and update  $J(p)$ .
  - 11:   **else**
  - 12:     Increase the damping parameter  $\lambda$  by a factor of 10 and update  $J(p)$ .
  - 13:   **end if**
  - 14:   Calculate  $\Delta p = \|p_{trial} - p\|$
  - 15: **end while**
  - 16: **return**  $p$
- 

## Convergence Criteria for Levenberg-Marquardt Algorithm

The convergence criteria for the Levenberg-Marquardt algorithm are typically determined based on the behavior of the cost function, which is defined as:

- Criterion 1: Residual norm convergence:  $\|e(p)\| < \epsilon_r$ ; Convergence is achieved when the norm of the residual vector  $e(p)$ , representing the difference between the model predictions and the measured responses, is smaller than a specified threshold  $\epsilon_r$ .
- Criterion 2: Parameter change convergence:  $\Delta p < \epsilon_p$ ; Convergence is achieved when the Euclidean norm of the parameter change vector  $\Delta p$  is smaller than a specified threshold ( $\epsilon_p$ ).
- Criterion 3: Cost function convergence:  $|J(p_{trial}) - J(p)| < \epsilon_J$ ; Convergence is achieved when the absolute difference between the cost functions of the trial and current parameter estimates,  $J(p_{trial})$  and  $J(p)$ , is smaller than a specified threshold  $\epsilon_J$ .

- Criterion 4: Damping parameter change convergence:  $|\lambda_{\text{trial}} - \lambda| < \epsilon_\lambda$ ; Convergence is achieved when the absolute difference between the damping parameters of the trial and current iterations,  $\lambda_{\text{trial}}$  and  $\lambda$ , is smaller than a specified threshold  $\epsilon_\lambda$ .
- Criterion 5: Maximum Iterations: Terminate the iterations if the number of iterations exceeds a pre-specified limit.

## The Nelder-Mead Simplex Method

The Nelder-Mead simplex method [25, 26] iteratively updates the simplex by reflecting, expanding, and contracting its vertices based on the cost function evaluations. This particular algorithm 2 is simple to implement and does not rely on gradient descent. This approach proves highly adept at handling nonlinear optimization problems characterized by noisy or discontinuous cost functions. However, algorithms converge slowly or become stuck in local minima in some cases, therefore, it may be necessary to use multiple starting points or combine them with other optimization techniques [20, 27, 28].

**Algorithm 2** Pseudocode of the standard Nelder-Mead simplex method for parameter estimation problems

---

**Require:** Initial parameter estimates  $p_0$  and simplex  $S_0$  with  $n + 1$  vertices, where  $n$  is the number of parameters to be estimated, and covers the parameter space of interest

**Ensure:** Best vertex  $p_0$  as the estimate

- 1: Evaluate the cost function  $J(\pi)$  for each vertex  $\pi$  in  $S_0$
  - 2: Sort the vertices in the order of increasing cost:  $J(p_0) \leq J(p_1) \leq \dots \leq J(p_n)$
  - 3: **while** stopping criterion not satisfied **do**
  - 4:   Compute the centroid of the  $n$  best vertices:  $c = \frac{1}{n}(p_0 + p_1 + \dots + p_{n-1})$
  - 5:   Reflect the worst vertex  $p_n$  about the centroid to obtain the trial point  $p_r = c + \alpha(c - p_n)$ , where  $\alpha > 0$  is a reflection coefficient
  - 6:   Evaluate the cost function  $J(p_r)$
  - 7:   **if**  $J(p_r) < J(p_0)$  **then**
  - 8:     Expand the simplex in the direction of  $p_r$ :  $p_e = c + \beta(p_r - c)$ , where  $\beta > 1$  is an expansion coefficient. Evaluate the cost function  $J(p_e)$
  - 9:   **else if**  $J(p_r) \geq J(p_1)$  **then**
  - 10:     Contract the simplex around the best vertex  $p_0$ :  $p_c = c + \gamma(p_0 - c)$ , where  $0 < \gamma < 1$  is a contraction coefficient. Evaluate the cost function  $J(p_c)$
  - 11:   **else**
  - 12:     Replace the worst vertex  $p_n$  with  $p_r$  and resort the vertices
  - 13:   **end if**
  - 14: **end while**
- 

## Convergence Criteria for the Nelder-Mead Simplex Algorithm

In the context of the Nelder-Mead simplex algorithm for parameter estimation of nonlinear systems, convergence is typically determined based on the behavior of the cost function. The stopping criterion involves assessing the size of the simplex, the function values, or

the change in parameters. The convergence criteria for the Nelder-Mead algorithm are as follows:

- Criterion 1: Simplex Size Convergence: The maximum size of the simplex is  $< \epsilon_s$ ; convergence is achieved when the maximum size of the simplex (the geometric shape formed by the current set of parameter values) is smaller than a specified threshold  $\epsilon_s$ .
- Criterion 2: Function Value Convergence:  $|J(p_{\text{worst}}) - J(p_{\text{best}})| < \epsilon_f$ ; Convergence is achieved when the absolute difference between the function values of the worst and best vertices is smaller than a specified threshold  $\epsilon_f$ .
- Criterion 3: Parameter Convergence:  $\max_i |p_{\text{worst},i} - p_{\text{best},i}| < \epsilon_x$ ; Convergence is achieved when the maximum absolute difference between the corresponding parameters of the worst and best vertices is smaller than a specified threshold  $\epsilon_x$ .
- Criterion 4: Maximum Iterations: Terminate the iterations if the number of iterations exceeds a pre-specified limit.

## Results

Our study focuses on evaluating the robustness and accuracy of the optimization methods discussed in Sect. 4. To achieve this goal, we conducted simulations involving a diverse array of systems varying in complexity. These systems were subjected to additive noise of different intensities, covering a wide spectrum of scenarios [29]. Our investigation primarily focused on understanding the impact of Gaussian noise on the performance of these optimization methods.

To assess the influence of noise, we introduce uncorrelated noise  $\eta(t)$  characterized by specific properties, as defined in Equation 9:

$$\langle \eta(t) \rangle = 0 \quad \text{and} \quad \langle \eta(t)\eta(t') \rangle = \delta(t - t') \quad (9)$$

Incorporating this noise into the equation of motion allowed us to observe its effects on the systems under scrutiny. Our analysis focused on understanding how the introduced Gaussian noise affects the robustness and accuracy of the optimization methods. We examined how varying noise intensities influenced the performance of these techniques, providing insights into their adaptability to different noise levels and their ability to accurately estimate parameters under such conditions.

The accuracy of the parameter estimation is judged by the root mean squared error (RMSE). The root mean square error is commonly used to evaluate the accuracy of a model's predictions. It assesses the variance or residuals between estimated and true values and is utilized to compare the estimated errors of various models for a given model. The formula for calculating the RMSE is as follows:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y^{Tr} - y^{Es})^2} \quad (10)$$

where  $N$  represents the total number of observations,  $y^{Tr}$  represents the actual value, and  $y^{Es}$  represents the estimated value.

The error-propagation problem can be adequately addressed by the boundary-value-problem methods for parameter estimation in ODE. In this section, we conduct numerical tests of the methods discussed in Sect. 4. Although the examples we use are not

representative of the full range of problems that these methods can address, they do allow us to evaluate the numerical properties of the algorithms. Specifically, we can assess the stability, reliability, efficiency, and accuracy of the methods, and examine their broader applicability.

### Van der Pol Oscillator

The van der Pol oscillator, a nonlinear second-order differential equation demonstrating limit cycle behavior [30], is defined as follows:

$$x'' - \mu(1 - x^2)x' + x = 0 \quad (11)$$

where  $x$  represents the displacement of the oscillator from its equilibrium, and  $\mu$  is a parameter governing nonlinear damping. This equation can be expressed as a system of first-order equations:

$$\begin{aligned} \frac{dx_1}{dt} &= x_2 \\ \frac{dx_2}{dt} &= \mu(1 - x_1^2)x_2 - x_1 \end{aligned} \quad (12)$$

The van der Pol oscillator serves as a canonical example of self-sustained oscillations.

To identify the system parameters, we conducted simulations using the model with the true parameter  $\mu = 1.5$  and initial condition  $[x_{1_0}, x_{2_0}]^T = [2.0, 0.0]^T$  from time  $t = 0$  to  $t = 20$  with a time-step size of  $\delta t = 0.01$ . Gaussian noise with a level of 0.1 was added to the simulated data for data collection. Subsequently, we applied the discussed optimization methods to estimate the system parameters. The results for all methods are presented in Table 1. The estimated parameters closely align with the true parameters, especially with the Nelder-Mead simplex algorithm.

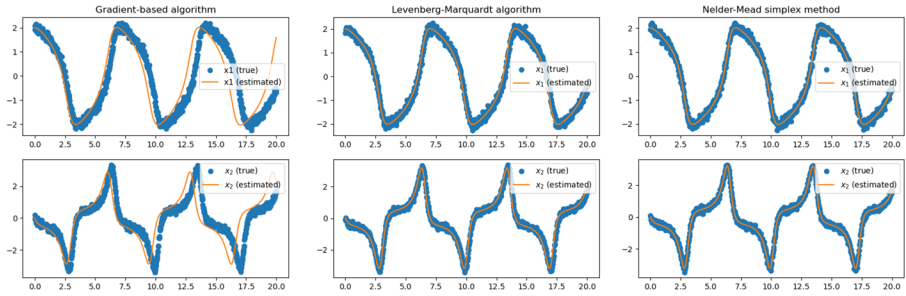
In Table 2, we computed the root mean squared error (RMSE) values and the computational cost for the discussed methods. The Nelder-Mead simplex algorithm outperformed the gradient-based and Levenberg-Marquardt methods, showing the lowest RMSE value of 0.1023. However, this approach incurs a slightly greater computational cost than does the Levenberg-Marquardt method.

We further compared the true and estimated trajectories in Fig. 1 and the phase space in Fig. 2. Remarkably, the Nelder-Mead simplex algorithm accurately captured the system's dynamic evolution with an estimated parameter very close to the true value.

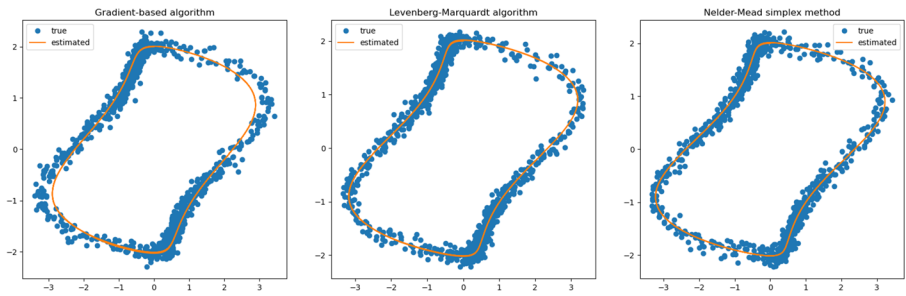
In this section, which focuses on the Nelder-Mead method, we explored the evolution of parameters from an initial estimate,  $p_{\text{initial}}$ , to values closer to the true parameter,  $p_{\text{true}}$ . Our analysis involved visualizations and metrics to evaluate convergence under different levels of noise  $\eta = [0.0001, 0.001, 0.01, 0.1]$ , as shown in Fig. 3. After a few iterations, the estimated parameters consistently converged to the true parameters, demonstrating a reliable approach.

A crucial aspect of our analysis involved evaluating the disparity between observed data values and the corresponding curve-fit estimates, as illustrated in Fig. 4. The histogram depicts the distribution of these fit errors. An ideal scenario involves these errors adhering to a normal distribution, indicating a robust fit.

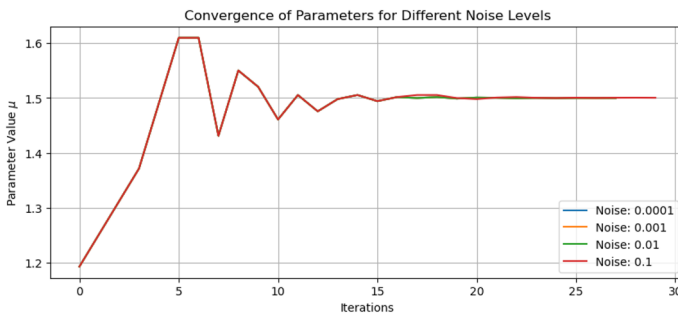




**Fig. 1** Exact trajectories of the van der Pol oscillator compared to the learned dynamics. The Blue lines represent the exact dynamics, while the red lines demonstrate the learned dynamics



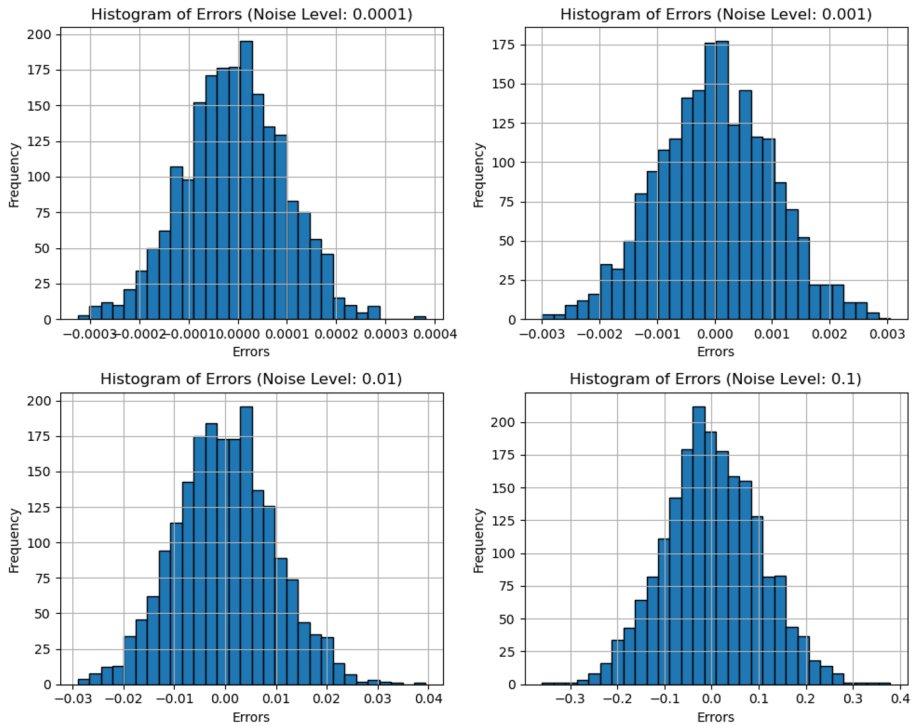
**Fig. 2** Exact phase portrait of the van der Pol Oscillator compared to the learned dynamics using various methods



**Fig. 3** Convergence of parameters for the van der Pol oscillator at different noise levels

## Rössler Systems

The Rössler system, a set of three coupled nonlinear ordinary differential equations known for exhibiting chaotic behavior, was introduced by Otto Rössler in 1976 [31–33]. These equations are given by:



**Fig. 4** Histogram of the errors between the data and the fit at different noise levels for van der Pol oscillator

**Table 1** Parameter Identification for van der Pol oscillators

Parameter	True value	Gradient-based	Levenberg-Marquardt	Nelder-Mead
$\mu$	1.5	1.2018	1.4611	1.5007

**Table 2** Root mean square error (RMSE) and computational cost for van der Pol Oscillator with different algorithms

Methods	RMSE	Computational time (seconds)
Gradient-based	0.8799	0.1505
Levenberg-Marquardt	0.1409	0.4356
Nelder-Mead	0.1023	0.4045

$$\begin{aligned}\frac{dx_1}{dt} &= -x_2 - x_3 \\ \frac{dx_2}{dt} &= x_1 + p_1 x_2 \\ \frac{dx_3}{dt} &= p_2 + x_3(x_1 - p_3)\end{aligned}\quad (13)$$

Here,  $x_1$ ,  $x_2$ , and  $x_3$  represent state variables, while  $p_1$ ,  $p_2$ , and  $p_3$  denote system parameters. The Rössler system is a classic chaotic system, that has been extensively studied across diverse fields due to its nonlinear and feedback-driven nature, leading to the emergence of a strange attractor.

To identify the parameters of the Rössler system, we conducted simulation using the model with true parameters  $p_1 = 0.2$ ,  $p_2 = 0.2$ , and  $p_3 = 5.7$  and initial conditions of  $[x_1, x_2, x_3]^T = [0.1, 0.1, 0.1]^T$  over  $t = 0$  to  $t = 120$  with a time-step size of  $\Delta t = 0.01$ . Gaussian noise with a level of 0.1 was added to the simulated data for data collection.

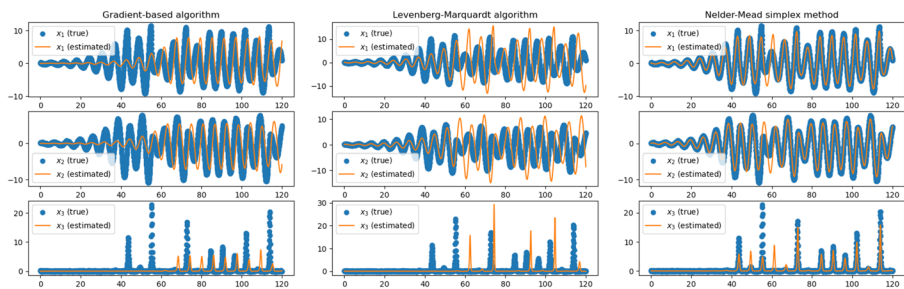
We employed optimization methods, including iterative gradient-based, Levenberg-Marquardt, and Nelder-Mead simplex methods, to estimate the parameters of the Rössler system. The results for all the methods discussed are summarized in Table 3. The estimated parameters closely align with the true values, particularly with the Nelder-Mead simplex algorithm.

Table 4 presents the root mean squared error (RMSE) values and computational costs for the discussed methods. The Nelder-Mead simplex algorithm outperformed the gradient-based and Levenberg-Marquardt methods, exhibiting the lowest RMSE value of 1.3199. However, this approach incurs a slightly greater computational cost than does the Levenberg-Marquardt method.

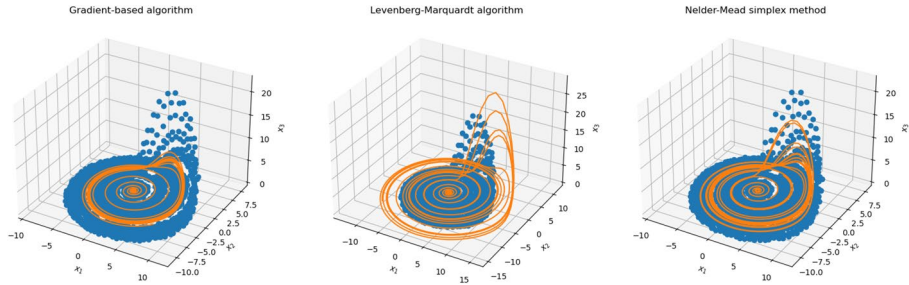
We further compared the true and estimated trajectories in Fig. 5 and the phase space in Fig. 6. Remarkably, the Nelder-Mead simplex algorithm accurately captured the system's dynamic evolution with an estimated parameter very close to the true value.

To assess the impact of noisy derivatives in a controlled environment, we introduced zero-mean Gaussian measurement noise with varying levels of  $\eta = [0.0001, 0.001, 0.01, 0.1]$ . Figure 7 shows the trajectories of the Rössler system over  $t = 0$  to  $t = 120$  under different levels of additive noise, highlighting the system's dynamics captured with estimated parameters even in the presence of noise.

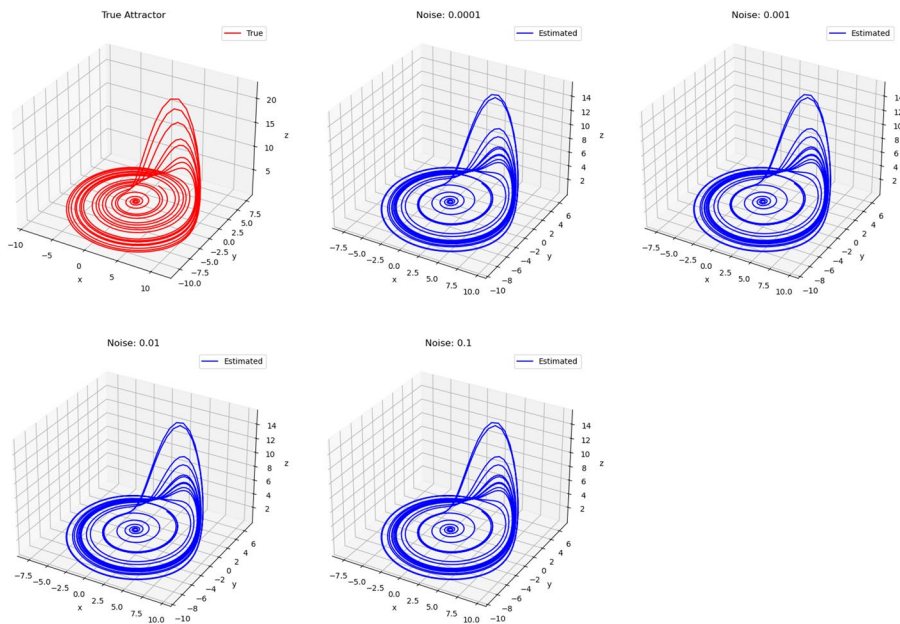
Focusing specifically on the Nelder-Mead simplex method, we explored the evolution of parameters from an initial estimate,  $p_{\text{initial}}$ , toward values closer to the true



**Fig. 5** Trajectories of the Rössler systems' exact dynamics (blue solid lines) compared to learned dynamics (red solid lines)



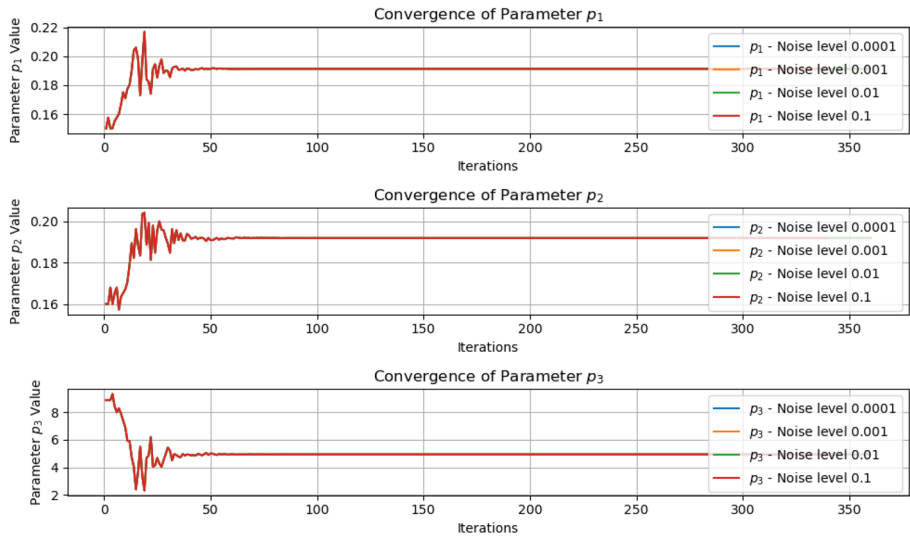
**Fig. 6** Phase portrait of the Rössler system's exact dynamics compared to learned dynamics using various methods



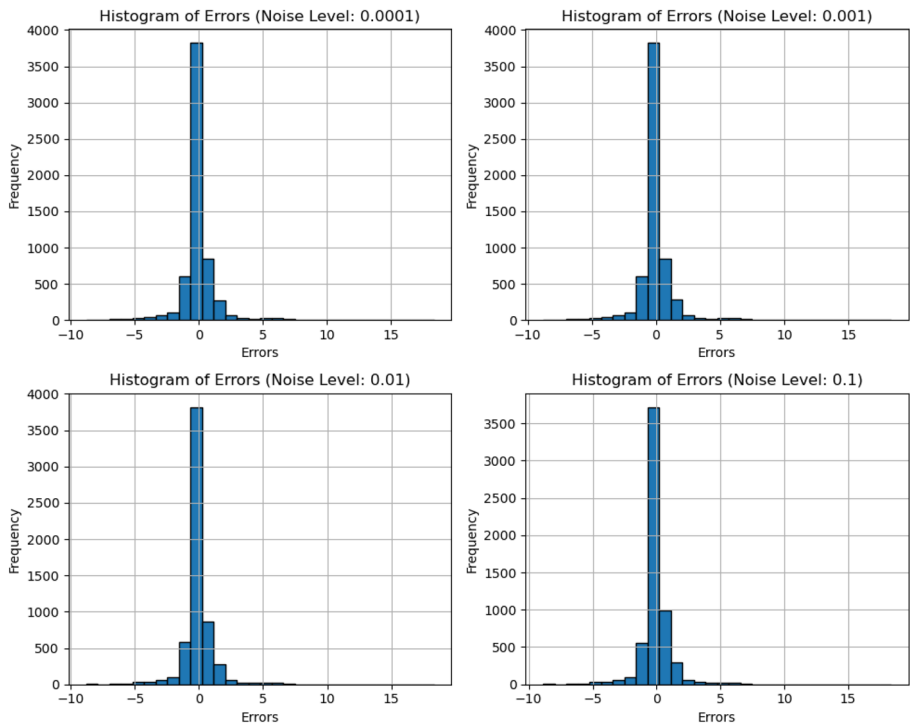
**Fig. 7** In this figure, we present the trajectories of the Rössler system for  $t = 0$  to  $t = 120$ . The true dynamics are depicted in red, while the identified systems obtained from the estimated parameters are displayed in blue. The performance of the identified systems is evaluated under different levels of additive noise

parameters,  $p_{\text{true}}$ . Our analysis involved visualizing and evaluating several metrics for convergence under different levels of noise  $\eta = [0.0001, 0.001, 0.01, 0.1]$ , as shown in Fig. 8. After a few iterations, the estimated parameters consistently converged to the true parameters. The depicted monotonic convergence emphasizes the reliability and consistency of our parameter estimation methodology.

Central to our assessment is the scrutiny of the disparities between the observed data and the curve-fit predictions. In Fig. 9, we present a histogram detailing the distribution of these fit errors. Ideally conforming to a normal distribution, these errors are indicative of the robustness of our curve fitting. Encouragingly, our analysis demonstrated



**Fig. 8** Convergence of parameters for the Rössler system at different noise levels



**Fig. 9** Histogram of the errors between the data and the fit of the Rössler system at different noise levels

**Table 3** Parameter Identification for the Rössler system

Parameter	True Value	Gradient-based	Levenberg-Marquardt	Nelder-Mead
$p_1$	0.2	0.1711	0.1499	0.1913
$p_2$	0.2	0.1501	0.1601	0.1918
$p_3$	5.7	4.418	8.8767	4.9344

**Table 4** Root mean square error (RMSE) and computational cost (seconds) for Rössler systems using different algorithms

Methods	RMSE	Computational cost (seconds)
Gradient-based	4.4776	0.2246
Levenberg-Marquardt	6.9895	1.8856
Nelder-Mead	1.3199	9.7509

close alignment with the anticipated normal distribution, underscoring the reliability of our model's predictive capabilities within the Rössler system.

## Pharmacokinetic Modeling

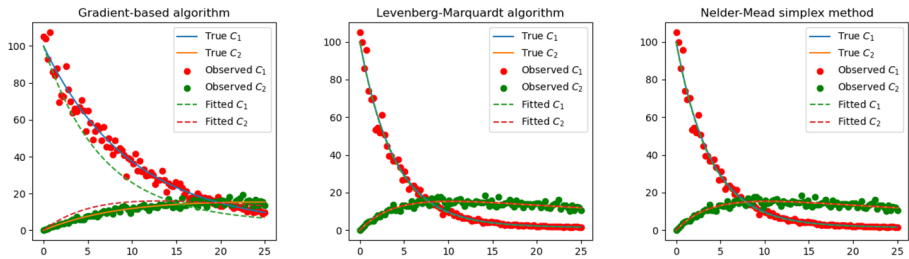
In drug development, scientists model how a new drug moves through the body-how it is absorbed, distributed, metabolized, and eliminated. The processes involved are complex and often nonlinear [34, 35]. One commonly used model is the two-compartment model called pharmacokinetic modeling, represented by the following equations:

$$\begin{aligned}\frac{dC_1}{dt} &= -k_{10}C_1 - k_{12}C_1 + k_{21}C_2 \\ \frac{dC_2}{dt} &= k_{12}C_1 - k_{21}C_2\end{aligned}\quad (14)$$

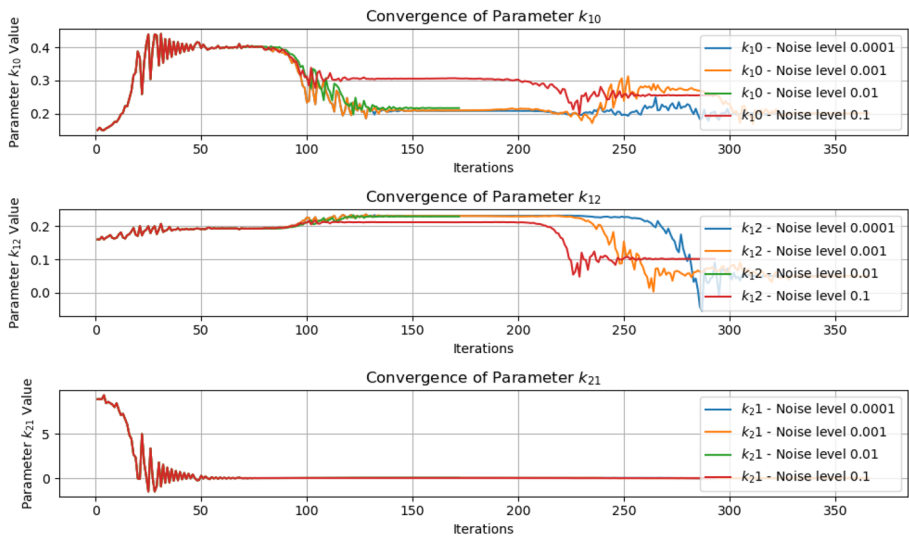
Here,  $C_1$  and  $C_2$  are the drug concentrations in different compartments, and  $k_{10}$ ,  $k_{12}$ , and  $k_{21}$  are related to the drug's behavior. The aim is to estimate these parameters from experimental data.

The two-compartment model becomes nonlinear due to exponential terms, leading to difficulty in finding an analytical solution. Nonlinear optimization methods such as the Levenberg-Marquardt or Nelder-Mead are commonly used to find the best-fit parameters for experimental data.

To assess the model's performance, we conducted a simulation with true parameters  $k_{10} = 0.2$ ,  $k_{12} = 0.05$ , and  $k_{21} = 0.03$ , using an initial condition of  $C_1(0) = 100$ , and  $C_2(0) = 0$  over the time interval  $t = 0$  to  $t = 25$  with 100 evenly spaced points. Multiplicative Gaussian noise with a level of 0.1 was added to the simulated data for data collection. We employed optimization methods, including iterative gradient-based, Levenberg-Marquardt, and Nelder-Mead simplex methods, to estimate the parameters of the system. During the optimization process to minimize the difference between the observed data and model predictions, the initial guesses for the parameters were set to  $k_{10}^{(0)} = 0.1$ ,  $k_{12}^{(0)} = 0.1$ ,  $k_{21}^{(0)} = 0.1$ . The results for all discussed methods are summarized in



**Fig. 10** Exact trajectories of pharmacokinetic modeling compared to the learned dynamics. The Blue and the red lines represent the exact dynamics, while the dash lines demonstrate the learned dynamics



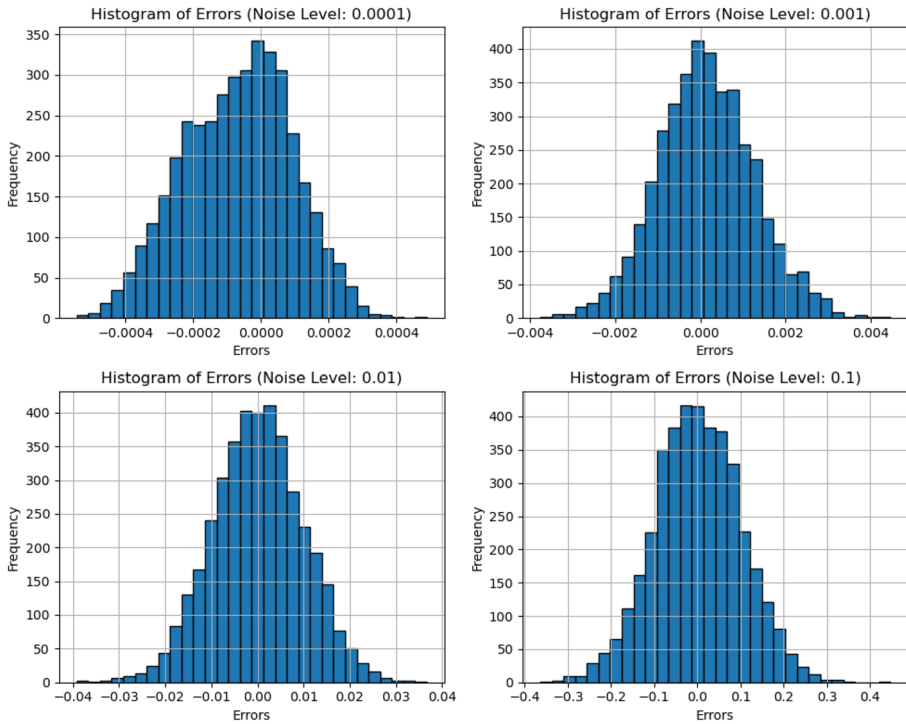
**Fig. 11** Convergence of parameters of the pharmacokinetic model at different noise levels

Table 5. The estimated parameters closely align with the true values, particularly with the Nelder-Mead simplex algorithm.

Table 6 presents the root mean squared error (RMSE) values and computational costs for the discussed methods. The Nelder-Mead simplex algorithm outperformed the gradient-based and Levenberg-Marquardt methods, exhibiting the lowest RMSE value of 1.8796. However, this approach incurs a slightly greater computational cost than does the Levenberg-Marquardt method. We further compared the true and estimated trajectories in Fig. 10. Remarkably, the Nelder-Mead simplex algorithm accurately captured the system's dynamic evolution with an estimated parameter very close to the true value.

Our focus on the Nelder-Mead simplex method involved exploring the evolution of parameters from an initial estimate,  $p_{\text{initial}}$ , toward values closer to the true parameter,  $p_{\text{true}}$ . Our analysis included visualizations and metrics to evaluate convergence under different levels of noise  $\eta = [0.0001, 0.001, 0.01, 0.1]$ .

We conducted a comprehensive analysis of parameter convergence and the accuracy of curve fitting derived from the initial guesses toward their true values, as shown in Fig. 11. The depicted monotonic convergence emphasizes the reliability and consistency of our



**Fig. 12** Histogram of the errors between the data and the fit of the pharmacokinetic model at different noise levels

**Table 5** Estimated parameters for pharmacokinetic modeling

Parameter	True Value	Gradient-based	Levenberg-Marquardt	Nelder-Mead
$k_{10}$	0.2	0.2727	0.2042	0.2041
$k_{12}$	0.05	0.0923	0.0509	0.0508
$k_{21}$	0.03	0.1044	0.0292	0.0291

**Table 6** Root mean square error (RMSE) and computational cost (seconds) for the pharmacokinetic model using different algorithms

Methods	RMSE	Computational cost (seconds)
Gradient-based	6.6439	0.1629
Levenberg-Marquardt	1.9870	0.1379
Nelder-Mead	1.8796	0.1609

parameter estimation methodology. A histogram of the errors between the data and the fit of the pharmacokinetic model at different noise levels is shown in Fig. 12. Ideally, the errors from curve fitting should follow a normal distribution. In this example, at various noise levels, it seems that these errors exhibit a normal distribution.



These simulation and parameter estimation results underscore the significance of nonlinear optimization techniques in the field of pharmacokinetics, where understanding drug behavior in the body is crucial for drug development and dosage optimization.

## Conclusions

In conclusion, our comprehensive exploration of three distinct nonlinear systems-namely, the van der Pol oscillator, the Rössler system, and pharmacokinetic modeling-provides valuable insights into the effectiveness of optimization methods for parameter estimation. The Nelder-Mead simplex algorithm consistently demonstrated superior performance, showcasing its robustness and accuracy across diverse and complex dynamic systems.

For the van der Pol oscillator, our simulations revealed that the Nelder-Mead simplex algorithm outperformed alternative optimization methods, achieving the lowest root Mean Squared Error of 0.1023. The ability of the algorithm to accurately capture system dynamics, even under the influence of noise, highlights its reliability. Moreover, the visualizations of parameter convergence illustrated consistent and monotonic convergence toward true values.

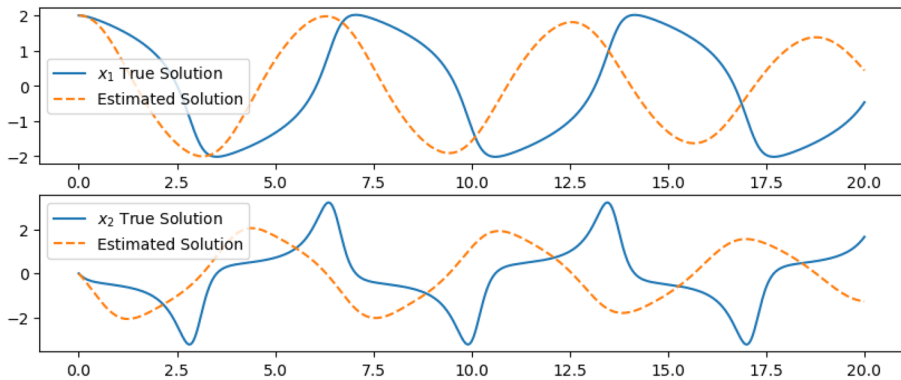
Similar commendable results were observed in the Rössler system, where the Nelder-Mead simplex algorithm exhibited the lowest RMSE value of 1.3199. The algorithm's accuracy in capturing the system's dynamics, coupled with the monotonic convergence of parameters under varying noise levels, emphasized its reliability and consistency in parameter estimation.

In the realm of pharmacokinetic modeling, the Nelder-Mead simplex algorithm continued to perform well, achieving the lowest RMSE value of 1.8796. The algorithm's accuracy in capturing the dynamic evolution of the system, coupled with its reliable parameter convergence, underscores its efficacy in the intricate context of drug behavior within the body.

Overall, our study reinforces the critical importance of thoughtful optimization method selection in nonlinear system parameter identification. The Nelder-Mead simplex algorithm has emerged as a powerful tool, consistently outperforming alternative methods and enhancing model accuracy across diverse systems. This approach not only refines parameter estimation methodologies but also deepens our understanding of complex dynamics, improving predictions and paving the way for advancements in vital research fields.

## Appendix A Comparison with Gauss-Newton Methods

In this section, we evaluate our methods by comparing them to other gradient-based approaches, such as Gauss-Newton methods, which are specifically applied to the van der Pol oscillator. 13 shows the disparities between the estimated trajectories and the true trajectories. Gauss-Newton methods, which are optimal for linear systems, are employed here. However, when dealing with nonlinear dynamics such as the van der Pol oscillator, the method is suboptimal, leading to significant errors. The root Mean Square Error (RMSE) for Gauss-Newton is 1.9965, while gradient-descent methods exhibit a lower RMSE of 0.8799.



**Fig. 13** Comparison of the exact trajectories of the Van der Pol oscillator with the learned dynamics. The Blue lines represent the exact dynamics, and the red lines depict the learned dynamics

**Author Contributions** KK was responsible for the conception and design of the study, as well as overseeing the data collection, development of the models, analysis, and drafting of the article. EK assisted in writing the manuscript and contributed to the development of models. All authors (KK, EK) reviewed and edited the manuscript.

**Funding** Open Access funding enabled and organized by Projekt DEAL. No funding was received for conducting this study.

**Data availability** The codes that produced the results in this paper are accessible to the public on GitHub after the manuscript has been accepted. If you require more information or clarification regarding the results, feel free to reach out to the corresponding author via email.

## Declarations

**Conflict of interest** The authors declare no Conflict of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Strogatz, S.: Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering. CRC Press, Chapman & Hall book (2019). <https://doi.org/10.1201/9780429492563>
2. Hirsch, M.W., Smale, S., Devaney, R.L.: 7 - nonlinear systems. In: Hirsch, M.W., Smale, S., Devaney, R.L. (eds.) Differential Equations, Dynamical Systems, and an Introduction to Chaos, 3rd edn., pp. 139–157. Academic Press, Boston (2013). <https://doi.org/10.1016/B978-0-12-382010-5.00007-5>
3. Ott, E.: Chaos in Dynamical Systems, 2nd edn. Cambridge University Press, Cambridge (2002). <https://doi.org/10.1017/CBO9780511803260>
4. Transtrum, M.K., Machta, B.B., Sethna, J.P.: Geometry of nonlinear least squares with applications to sloppy models and optimization. Phys. Rev. E **83**, 036701 (2011). <https://doi.org/10.1103/PhysRevE.83.036701>
5. Transtrum, M.K., Machta, B.B., Sethna, J.P.: Why are nonlinear fits to data so challenging? Phys. Rev. Lett. **104**, 060201 (2010). <https://doi.org/10.1103/PhysRevLett.104.060201>

6. Nocedal, J., Wright, S.: Numerical optimization. Springer Ser. Op. Res. Financial Eng. (2006). <https://doi.org/10.1007/978-0-387-40065-5>
7. Li, J., Ding, R.: Parameter estimation methods for nonlinear systems. Appl. Math. Comput. **219**(9), 4278–4287 (2013). <https://doi.org/10.1016/j.amc.2012.09.045>
8. Johnson, M.L., Faunt, L.M.: Methods in Enzymology. In: Parameter estimation by least-squares methods, vol. 210, pp. 1–37. Academic Press, Cambridge (1992). [https://doi.org/10.1016/0076-6879\(92\)10003-V](https://doi.org/10.1016/0076-6879(92)10003-V)
9. Baake, E., Baake, M., Bock, H.G., Briggs, K.M.: Fitting ordinary differential equations to chaotic data. Phys. Rev. A **45**, 5524–5529 (1992). <https://doi.org/10.1103/PhysRevA.45.5524>
10. Gavin, H.P.: The levenberg-marquardt method for nonlinear least squares curve-fitting problems c ©. (2013). <https://api.semanticscholar.org/CorpusID:5708656>
11. Shawash, J., Selviah, D.R.: Real-time nonlinear parameter estimation using the levenberg-marquardt algorithm on field programmable gate arrays. IEEE Trans. Industr. Electron. **60**(1), 170–176 (2013). <https://doi.org/10.1109/TIE.2012.2183833>
12. Olsson, D.M., Nelson, L.S.: The nelder-mead simplex procedure for function minimization. Technometrics **17**(1), 45–51 (1975). <https://doi.org/10.1080/00401706.1975.10489269>
13. Wang, P.C., Shoup, T.E.: Parameter sensitivity study of the nelder-mead simplex method. Adv. Eng. Softw. **42**(7), 529–533 (2011). <https://doi.org/10.1016/j.advengsoft.2011.04.004>
14. Kumar, Kaushal: Data-driven modeling and parameter estimation of nonlinear systems. Eur. Phys. J. B **96**(7), 107 (2023). <https://doi.org/10.1140/epjb/s10051-023-00574-3>
15. Bock, H.G., Kostina, E., Schlöder, J.P.: Numerical methods for parameter estimation in nonlinear differential algebraic equations. GAMM Mitteilungen **30**(2), 376–408 (2007)
16. Bock, H.G., Kostina, E., Schlöder, J.P.: Direct multiple shooting and generalized gauss-newton method for parameter estimation problems in ode models. In: Carraro, T., Geiger, M., Körkel, S., Rannacher, R. (eds.) Multiple Shooting and Time Domain Decomposition Methods, pp. 1–34. Springer, Cham (2015)
17. Brunton, S.L., Proctor, J.L., Kutz, J.N.: Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proc. Natl. Acad. Sci. **113**(15), 3932–3937 (2016). <https://doi.org/10.1073/pnas.1517384113>
18. Raissi, M., Perdikaris, P., Karniadakis, G.E.: Multistep Neural Networks for Data-driven Discovery of Nonlinear Dynamical Systems (2018). [arXiv:1801.01236v1](https://arxiv.org/abs/1801.01236v1)
19. Kumar, K.: Machine learning in parameter estimation of nonlinear systems (2023). [arXiv:2308.12393v1](https://arxiv.org/abs/2308.12393v1)
20. Wang, D., Yang, G., Ding, R.: Gradient-based iterative parameter estimation for box-jenkins systems. Comput. Math. Appl. **60**(5), 1200–1208 (2010). <https://doi.org/10.1016/j.camwa.2010.06.001>
21. Least-Squares Problems, pp. 245–269. Springer, New York, NY (2006). [https://doi.org/10.1007/978-0-387-40065-5\\_10](https://doi.org/10.1007/978-0-387-40065-5_10)
22. Fan, J.-Y.: A modified levenberg-marquardt algorithm for singular system of nonlinear equations. J. Comput. Math. **21**(5), 625–636 (2003)
23. Amini, K., Rostami, F.: A modified two steps levenberg-marquardt method for nonlinear equations. J. Comput. Appl. Math. **288**, 341–350 (2015). <https://doi.org/10.1016/j.cam.2015.04.040>
24. Marquardt, D.W.: An algorithm for least-squares estimation of nonlinear parameters. J. Soc. Ind. Appl. Math. **11**(2), 431–441 (1963). <https://doi.org/10.1137/0111030>
25. Derivative-Free Optimization, pp. 220–244. Springer, New York, NY (2006). [https://doi.org/10.1007/978-0-387-40065-5\\_9](https://doi.org/10.1007/978-0-387-40065-5_9)
26. Nelder, J.A., Mead, R.: A simplex method for function minimization. Comput. J. **7**, 308–313 (1965)
27. Barton, R.R., Ivey, J.S.: Nelder-mead simplex modifications for simulation optimization. Manage. Sci. **42**(7), 954–973 (1996)
28. Xu, S., Wang, Y., Wang, Z.: Parameter estimation of proton exchange membrane fuel cells using eagle strategy based on jaya algorithm and nelder-mead simplex method. Energy **173**, 457–467 (2019). <https://doi.org/10.1016/j.energy.2019.02.106>
29. Gottwald, G.A., Harlim, J.: The role of additive and multiplicative noise in filtering complex dynamical systems. Proc. R. Soc. A Math. Phys. Eng. Sci. **469**(2155), 20130096 (2013). <https://doi.org/10.1098/rspa.2013.0096>
30. Kanamaru, T.: Van der Pol oscillator. Scholarpedia **2**(1), 2202 (2007). <https://doi.org/10.4249/scholarpedia.2202>
31. Letellier, C., Rossler, O.E.: Rossler attractor. Scholarpedia **1**(10), 1721 (2006). <https://doi.org/10.4249/scholarpedia.1721.revision>
32. Rössler, O.E.: Different types of chaos in two simple differential equations. Zeitschrift Naturforschung Teil A **31**(12), 1664–1670 (1976). <https://doi.org/10.1515/zna-1976-1231>
33. Rössler, O.E.: Chaotic behavior in simple reaction systems. Zeitschrift für Naturforschung A **31**(3–4), 259–264 (1976). <https://doi.org/10.1515/zna-1976-3-408>

34. Liu, Z., Yang, Y.: Pharmacokinetic model based on multifactor uncertain differential equation. *Appl. Math. Comput.* **392**, 125722 (2021). <https://doi.org/10.1016/j.amc.2020.125722>
35. Lon, H.-K., Liu, D., Jusko, W.J.: Pharmacokinetic/pharmacodynamic modeling in inflammation. *Crit Rev Trade Biomed Eng* **40**(4), 295–312 (2012)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.