

On classical and modern approaches to the inverse problem of celestial mechanics

Sergey Shorokhov

Abstract. The inverse problem of celestial mechanics seeks to determine the potential force field that produces a given family of orbits. We address both classical and modern approaches to this problem. We present a classical analytical approach based on a system of partial differential equations (PDEs) for the force and energy functions. We also explore a neural network approach using neural ordinary differential equations (neural ODEs), outlining the forward- and reverse-time ODE systems required to train the neural network to learn the force function. This synthesis of approaches highlights the complementary nature of classical analytical methods and modern computational techniques.

Introduction

The inverse problem of celestial mechanics is a foundational challenge in dynamical astronomy and theoretical mechanics: given a prescribed family of orbits or a dataset of trajectory observations, determine the underlying potential force field that governs a dynamical system's motion [1].

The systematic study of the inverse problem gained significant momentum in the mid-20th century, driven by observations of satellites and other space objects. V. Szebehely [2] formalized the condition (first-order PDE) under which a given family of curves can be realized as orbits of a dynamical system, particularly in the context of satellite observations and Earth's rotation. G. Bozis [3] investigated the inverse problem for families of orbits in two and three dimensions, highlighting the role of energy function and integrability conditions. The exact solvability of partial differential equation (PDE) systems for force and energy functions was established in [4], where specific classes of potentials admitting prescribed orbital families were derived. Recent papers by T. Kotoulas [5] have explored the algebraic and geometric structures underlying these PDE systems, offering new approaches to their resolution.

Recently, the inverse problem has evolved beyond purely analytical methods. Massive astrometric datasets from missions such as Gaia, combined with advanced computational methods, demand new approaches. Meanwhile, machine learning techniques have proven effective at inferring functional dependencies directly from noisy, discrete observations. Our goal is to bridge these approaches by presenting both traditional PDE formulations and their contemporary neural network counterparts.

1. Partial differential equations approach

We study the inverse problem of constructing a force function $U(\mathbf{x})$, $\mathbf{x} = (x_1, x_2, x_3)$ for a material point in three-dimensional space in a Cartesian coordinate system, with equations of motion

$$\ddot{\mathbf{x}} = \nabla_{\mathbf{x}} U(\mathbf{x}), \quad (1)$$

where $\nabla_{\mathbf{x}}$ denotes the operation of computing the gradient with respect to $\mathbf{x}$, admitting a family of trajectories defined in domain $\Omega \subset \mathbb{R}^3$ by relations of the form

$$\omega_1(\mathbf{x}) = c_1, \omega_2(\mathbf{x}) = c_2, c_1, c_2 \in \mathbb{R} \quad (2)$$

The Jacobian matrix $\frac{\partial(\omega_1, \omega_2)}{\partial(x_1, x_2, x_3)}$ is assumed to have maximum rank.

When a material point moves in a potential force field, the energy integral is conserved:

$$\frac{1}{2} \|\dot{\mathbf{x}}\|^2 - U(\mathbf{x}) = h, \|\dot{\mathbf{x}}\|^2 = \dot{x}_1^2 + \dot{x}_2^2 + \dot{x}_3^2.$$

Each curve within the family (2) corresponds to a specific energy level. Thus, the energy function $h(\mathbf{x})$ remains constant along any given trajectory:

$$h(\mathbf{x}) = h(\omega_1(\mathbf{x}), \omega_2(\mathbf{x})).$$

As was demonstrated in [4], solving the inverse problem of celestial mechanics requires investigating the compatibility and finding a solution to the following system of first-order partial differential equations for the unknown functions $U(\mathbf{x})$ and $h(\mathbf{x})$:

$$\begin{cases} \langle \nabla_{\mathbf{x}} \omega_1, \nabla_{\mathbf{x}} U - (U + h) \mathbf{p} \rangle = 0, \\ \langle \nabla_{\mathbf{x}} \omega_2, \nabla_{\mathbf{x}} U - (U + h) \mathbf{p} \rangle = 0, \\ \langle E \mathbf{s}, \nabla_{\mathbf{x}} U - (U + h) \mathbf{p} \rangle = -2(U + h) \operatorname{div}(E \mathbf{s}) - \langle \mathbf{q}, \nabla_{\mathbf{x}} h \rangle, \\ \langle \mathbf{s}, \nabla_{\mathbf{x}} h \rangle = 0, \end{cases} \quad (3)$$

where $\mathbf{s} = \nabla_{\mathbf{x}} \omega_1 \times \nabla_{\mathbf{x}} \omega_2$, $E = \frac{1}{\|\mathbf{s}\|^2} \langle \mathbf{s}, \operatorname{rots} \rangle$, $\mathbf{p} = \frac{1}{\|\mathbf{s}\|^2} (\nabla_{\mathbf{x}}^T \mathbf{s}) \mathbf{s}$, $\mathbf{q} = \mathbf{p} \times \mathbf{s}$. Here, the angle brackets denote the scalar product in $\mathbb{R}^3$, and the operation $\times$ denotes the vector product in $\mathbb{R}^3$.

If the functions $\omega_1(\mathbf{x})$ and $\omega_2(\mathbf{x})$ in (2) are known, the PDE system (3) is a linear first-order system of four PDEs in the two unknowns $U(\mathbf{x})$ and $h(\mathbf{x})$. But often the functions $\omega_1(\mathbf{x})$ and $\omega_2(\mathbf{x})$ in (2) are not given in advance, and PDE

system (3) is a nonlinear system of four PDEs in the four unknowns $\omega_1(\mathbf{x})$, $\omega_2(\mathbf{x})$, $U(\mathbf{x})$, and $h(\mathbf{x})$, where three PDEs in (3) are first-order and one is third-order.

Because solving the inverse problem of celestial mechanics involves a system of PDEs for several unknown functions, this suggests the possible application of either Riquier theory [6] or the Gröbner basis method [7]. In practice, finding exact solutions to PDE system (3) is highly challenging. To address this complexity, various simplifying assumptions are introduced regarding the form of the unknown functions [5]. This yields simpler differential equations for which exact analytical solutions can sometimes be obtained.

2. Neural network approach

In the neural network approach, the unknown force function U is assumed to be the output of a neural network $\mathcal{F}$ with input $\mathbf{x} \in \mathbb{R}^3$ and parameters (weights) $\boldsymbol{\theta} \in \mathbb{R}^p$.

So we analyze a second-order ODE system

$$\ddot{\mathbf{x}} = \nabla_{\mathbf{x}} U(\mathbf{x}, \boldsymbol{\theta}) \quad (4)$$

where the gradient of the force function $U(\mathbf{x}, \boldsymbol{\theta})$ with respect to the variables $\mathbf{x}$ in (4) is computed using the exact automatic differentiation implemented in modern deep learning frameworks.

We train a neural network $\mathcal{F}$ on the existing set of data points

$$\mathbf{D} = \left\{ \left(x_1^{(k)}, x_2^{(k)}, x_3^{(k)}, x_1'^{(k)}, x_2'^{(k)}, x_3'^{(k)}, t_k \right), k = \overline{1, N} \right\},$$

where the vector $\mathbf{x}_k = \left(x_1^{(k)}, x_2^{(k)}, x_3^{(k)} \right)$ represents the observed position and the vector $\mathbf{x}'_k = \left(x_1'^{(k)}, x_2'^{(k)}, x_3'^{(k)} \right)$ represents the observed velocity of the material point at time t_k , so that the trajectories $\mathbf{x}(t; \boldsymbol{\theta})$ of system (4) match the observations in $\mathbf{D}$ as closely as possible (i.e., the trajectories pass through the points in $\mathbf{D}$ or are in their vicinity).

We replace the system (4) of three second-order equations with the following system of six first-order equations in the variables $(\mathbf{x}, \mathbf{x}') = (x_1, x_2, x_3, x'_1, x'_2, x'_3)$:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{x}', \\ \dot{\mathbf{x}}' = \nabla_{\mathbf{x}} U(\mathbf{x}, \boldsymbol{\theta}), \end{cases} \quad (5)$$

By integrating the system (5) using a numerical algorithm, we can obtain the set of trajectory points $(\mathbf{x}(t; \boldsymbol{\theta}), \mathbf{x}'(t; \boldsymbol{\theta}))$ at times $t_k, k = \overline{1, N}$: $\mathbf{x}_k = \mathbf{x}(t_k; \boldsymbol{\theta})$, $\mathbf{x}'_k = \mathbf{x}'(t_k; \boldsymbol{\theta})$.

We assume that the loss functional $\mathcal{L}$ for training the neural network $\mathcal{F}$ takes the form of the sum of the estimates of the deviations of the trajectory $\mathbf{x}(t; \boldsymbol{\theta})$

from the points in the observation set $\mathbf{D}$, measured in both position and velocity:

$$\mathcal{L}[\mathbf{x}(t; \boldsymbol{\theta}), \mathbf{x}'(t; \boldsymbol{\theta})] = \sum_{k=1}^N \rho(\mathbf{x}(t_k; \boldsymbol{\theta}), \mathbf{x}'(t_k; \boldsymbol{\theta}), \mathbf{x}^{(k)}, \mathbf{x}'^{(k)}), \quad (6)$$

where $\rho(\mathbf{x}(t_k; \boldsymbol{\theta}), \mathbf{x}'(t_k; \boldsymbol{\theta}), \mathbf{x}^{(k)}, \mathbf{x}'^{(k)}) = \|\mathbf{x}(t_k; \boldsymbol{\theta}) - \mathbf{x}^{(k)}\|^2 + \|\mathbf{x}'(t_k; \boldsymbol{\theta}) - \mathbf{x}'^{(k)}\|^2$. To minimize the loss $\mathcal{L}$ using gradient descent in the parameter space $\boldsymbol{\theta}$ of the neural network $\mathcal{F}$, we must compute the gradient of $\mathcal{L}$ with respect to $\boldsymbol{\theta}$.

Following [8], we introduce the adjoint variables $\boldsymbol{\chi}(t) = \nabla_{\mathbf{x}} \mathcal{L}$, $\boldsymbol{\chi}'(t) = \nabla_{\mathbf{x}'} \mathcal{L}$, and $\boldsymbol{\vartheta}(t) = \nabla_{\boldsymbol{\theta}} \mathcal{L}$, which represent the gradients of the loss functional $\mathcal{L}$ with respect to the states $\mathbf{x}$, the velocities $\mathbf{x}'$, and the parameters $\boldsymbol{\theta}$ at time $t \in [t_0, T_N]$.

The dynamics of the variables (gradients) $\boldsymbol{\chi}(t)$, $\boldsymbol{\chi}'(t)$, and $\boldsymbol{\vartheta}(t)$ are governed by systems of first- and second-order ODEs in reverse time, defined on the intervals $[t_k, t_{k+1}]$ for $k = N-1, N-2, \dots, 0$:

$$\begin{cases} \ddot{\mathbf{x}} = \nabla_{\mathbf{x}} U, & \mathbf{x}(t_{k+1}) = \mathbf{x}_{k+1}, \\ & \dot{\mathbf{x}}(t_{k+1}) = \mathbf{x}'_{k+1}, \\ \ddot{\boldsymbol{\chi}}' = \nabla_{\mathbf{x}'}^2 U \boldsymbol{\chi}', & \boldsymbol{\chi}'(t_{k+1}) = \nabla_{\mathbf{x}'} \rho(\mathbf{x}_{k+1}, \mathbf{x}'_{k+1}, \mathbf{x}^{(k+1)}, \mathbf{x}'^{(k+1)}), \\ & \dot{\boldsymbol{\chi}}'(t_{k+1}) = -\nabla_{\mathbf{x}} \rho(\mathbf{x}_{k+1}, \mathbf{x}'_{k+1}, \mathbf{x}^{(k+1)}, \mathbf{x}'^{(k+1)}), \\ \dot{\boldsymbol{\vartheta}} = -\nabla_{\boldsymbol{\theta}} (\nabla_{\mathbf{x}}^T U \boldsymbol{\chi}'), & \boldsymbol{\vartheta}(t_{k+1}) = \mathbf{0}. \end{cases} \quad (7)$$

The system of first- and second-order ODEs (7) in the variables $\mathbf{x}(t)$, $\boldsymbol{\chi}'(t)$, and $\boldsymbol{\vartheta}(t)$ must be solved backward in time from t_N to t_0 .

Since the functional $\mathcal{L}$ depends on the position and velocity at the final time t_N and on the positions and velocities at intermediate times $t_1, t_2, \dots, t_{N-1}$, we can solve the system of equations (7) in reverse time on each interval $[t_{N-1}, t_N]$, $[t_{N-2}, t_{N-1}]$, etc. (in reverse order), and obtain the resulting gradients $\boldsymbol{\vartheta}(t_k)$ of the loss functional $\mathcal{L}$ at times $t_0, t_1, \dots, t_{N-1}$.

Thus, the gradient of the loss functional $\mathcal{L}[\mathbf{x}(t; \boldsymbol{\theta})]$ with respect to $\boldsymbol{\theta}$ is obtained as the sum of the gradients $\boldsymbol{\vartheta}(t_k)$ computed by solving the systems (7) on the time intervals $[t_{N-1}, t_N]$, $[t_{N-2}, t_{N-1}]$, ..., $[t_0, t_1]$:

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}[\mathbf{x}(t; \boldsymbol{\theta})] = \sum_{k=0}^{N-1} \boldsymbol{\vartheta}(t_k; \boldsymbol{\theta}). \quad (8)$$

Training the neural network for the force function U on families of curves obtained in [4] demonstrates the convergence of the learning process and the closeness of the trajectories of the ODE system (4) to the curves of the target family of orbits.

Conclusion

The inverse problem of celestial mechanics has evolved from a purely theoretical mathematical challenge into a dynamic, data-driven field of research. The classical approach, based on partial differential equations, provides rigorous necessary and sufficient conditions for the existence of potentials that produce prescribed orbital families.

However, the limitations of classical PDE methods in handling noisy, high-dimensional, or incomplete observational data have paved the way for modern machine learning techniques. Among these, the neural ODE framework offers a powerful alternative to traditional PDE-based formulations. By casting the learning process as a coupled system of forward- and reverse-time ODEs, this method efficiently computes gradients and learns complex potentials directly from data while preserving the continuous-time dynamical structure of the underlying physics.

Ultimately, the future of this field lies in the synergy between both paradigms. Classical PDE theory can contribute physical constraints, symmetries, and regularization terms to guide neural network training, ensuring that the learned potentials are both data-consistent and analytically sound. As astrometric datasets grow in volume and precision, this hybrid framework will become indispensable for uncovering the hidden gravitational landscapes of the universe.

References

- [1] A. S. Galiullin, *Inverse Problems of Dynamics*. Mir Publishers, Moscow, 1984.
- [2] V. Szebehely, On the determination of the potential by satellite observations. In *Proceedings of the International Meeting on Earth's Rotation by Satellite Observation*, Ed. by G. Proverbio (The University of Cagliari Bologna, Bologna, 1974), p. 31–35.
- [3] G. Bozis, The inverse problem of dynamics: basic facts, *Inverse Probl.* **11**, 687–708 (1995). <https://doi.org/10.1088/0266-5611/11/4/006>
- [4] S. G. Shorokhov, Solution of an inverse problem of the dynamics of a particle. *Celest. Mech.* **44**, 193–206 (1988). <https://doi.org/10.1007/BF01230715>
- [5] T. Kotoulas, New Solutions of the 3D Inverse Problem of Newtonian Dynamics, *Lobachevskii J Math* **46**, 4046–4059 (2025). <https://doi.org/10.1134/s1995080225609270>
- [6] C. Riquier, *Les Systèmes d'Équations aux Dérivées Partielles*. Gauthier-Villars, Paris, 1910.
- [7] B. Buchberger, Grobner Bases: an Algorithmic Method in Polynomial Ideal Theory. In: *Recent Trends in Multidimensional System Theory*, Bose, N.K. (ed.), Reidel, Dordrecht, 1985, pp. 184–232.
- [8] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, Neural Ordinary Differential Equations. In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS'18)* (Curran Associates Inc., 2018), p. 6572.

- [9] S. G. Shorokhov, Building a Neural Ordinary Differential Equation Using Methods for Solving Inverse Problems of Dynamics, *Moscow Univ. Phys.* **80**, S996–S1003 (2025).
<https://doi.org/10.3103/S0027134925702856>

Sergey Shorokhov

Dept. of Mathematical Modeling and Artificial Intelligence

Peoples' Friendship University of Russia named after Patrice Lumumba

Moscow, Russia

e-mail: shorokhov-sg@rudn.ru