

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

Shorokhov S.G.

Analytical Methods of Celestial Mechanics 2026

Euler International Mathematical Institute
St. Petersburg, Russia
August 17-21, 2026





Problem Statement:

Determine the force function $U(\mathbf{x})$ of a potential force field that admits a given family of curves as particle trajectories.

Key milestones:

Period	Authors	Results
1687	I. Newton	The first inverse problem: the law of gravitation from Kepler's laws
1877–1890	J. Bertrand, U. Dainelli, Н.Е. Жуковский, Г.К. Суслов	Force functions for conic sections; force functions for planar (2D) trajectories; force functions for 3D trajectories; PDEs for n -dimensional trajectories
1960– present	V. Szebehely, F. Varadi, B. Erdi, G. Bozis, T. Kotoulas et al.	First-order PDE for the force function; geometric approaches with reduction to algebraic equations, etc.



SOLUTION OF AN INVERSE PROBLEM OF THE DYNAMICS OF A PARTICLE

S. G. SHOROKHOV

Department of Theoretical Mechanics, Moscow Friendship University, U.S.S.R.

(Received: 2 May 1988; accepted: 2 May 1988)

Abstract. The three-dimensional inverse problem of particle dynamics is studied here. The potential U and the corresponding energy h are determined by the given family of possible trajectories. The classification of the solutions due to the geometry of the given family is obtained.

Celestial Mechanics **44** (1988) 193–206.

© 1988 by Kluwer Academic Publishers.



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

$$\ddot{\mathbf{x}} = \nabla U(\mathbf{x}), \quad \ddot{\mathbf{x}} = (\ddot{x}_1, \ddot{x}_2, \ddot{x}_3), \quad (1)$$

where $\nabla = \left(\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \frac{\partial}{\partial x_3} \right)$ denotes the operation of computing the gradient with respect to $\mathbf{x}$, admitting a family of trajectories defined in a domain $\Omega \subset \mathbb{R}^3$ by equalities of the form

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

The Jacobian matrix $\frac{\partial(\omega_1, \omega_2)}{\partial(x_1, x_2, x_3)}$ is assumed to have maximum rank 2.



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

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

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 of (2):

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

This condition can be written using a first-order linear PDE of the form

$$\langle \nabla \omega_1 \times \nabla \omega_2, \nabla h \rangle = 0,$$

where the angle brackets denote the scalar product in $\mathbb{R}^3$, the operation $\times$ denotes the vector product in $\mathbb{R}^3$.



Solving the inverse problem of celestial mechanics requires investigating the compatibility and finding a solution to the following PDE system for the unknown functions $U(\mathbf{x})$ and $h(\mathbf{x})$:

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

where

$$\begin{aligned} \mathbf{s} &= \nabla \omega_1 \times \nabla \omega_2, \\ E &= \frac{1}{\|\mathbf{s}\|^2} \langle \mathbf{s}, \operatorname{rots} \rangle, \\ \mathbf{p} &= \frac{1}{\|\mathbf{s}\|^2} (\nabla^T \mathbf{s}) \mathbf{s}, \\ \mathbf{q} &= \mathbf{p} \times \mathbf{s}. \end{aligned}$$



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 the 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.

In practice, finding exact solutions to the PDE system (3) is highly challenging. To address this complexity, various simplifying assumptions were introduced regarding the form of the unknown functions. These assumptions yield simpler differential equations for which exact analytical solutions can sometimes be obtained.

When analyzing the compatibility of a system of PDEs, we say that the generality of the solution is $\infty^{k_0+k_1 f(1)+k_2 f(2)+\dots}$ if the general solution contains k_0 arbitrary constants, k_1 arbitrary functions of one argument, k_2 arbitrary functions of two arguments, and so on.



Along each curve of the family (2), we consider the Frenet trihedron $\{\boldsymbol{\tau}, \boldsymbol{\nu}, \boldsymbol{\beta}\}$, where $\boldsymbol{\tau}$ is the tangent vector, $\boldsymbol{\nu}$ is the principal normal, and $\boldsymbol{\beta}$ is the binormal. Consequently, a moving reference frame $\{\boldsymbol{\tau}(\mathbf{x}), \boldsymbol{\nu}(\mathbf{x}), \boldsymbol{\beta}(\mathbf{x})\}$ is defined at each point $\mathbf{x}$ within the domain Ω .

Theorem 0. The force function $u(\mathbf{x}) \neq \text{const}$ is a nontrivial solution to the inverse problem in Ω if and only if $u(\mathbf{x})$ and the corresponding total energy distribution $h(\mathbf{x})$ satisfy the following system of first-order PDEs:

$$\begin{cases} \langle \boldsymbol{\tau}, \nabla h \rangle = 0, & (E_1) \\ \langle \boldsymbol{\nu}, \nabla u \rangle - 2k_1 (u + h) = 0, & (E_2) \\ \langle \boldsymbol{\beta}, \nabla u \rangle = 0, & (E_3) \\ \langle \boldsymbol{\tau}, \text{rot} \boldsymbol{\tau} \rangle \langle \boldsymbol{\tau}, \nabla u \rangle + 2\sigma (u + h) - 2k_1 \langle \boldsymbol{\beta}, \nabla h \rangle = 0 & (E_4) \end{cases}$$

and the inequality $u(\mathbf{x}) + h(\mathbf{x}) > 0$.

Here, k_1 is the curvature, $\sigma = -\text{div}(k_1 \boldsymbol{\beta}) = \text{div}(\langle \boldsymbol{\tau}, \text{rot} \boldsymbol{\tau} \rangle \boldsymbol{\tau})$. When considering specific problems, it is preferable not to predefine the domain Ω . Instead, one should seek global solutions to the system (E_1) – (E_4) and subsequently determine Ω using the inequality $u(\mathbf{x}) + h(\mathbf{x}) > 0$.



Theorem 1. If the curvature k_1 of each curve in the family (2) vanishes identically ($k_1 \equiv 0$), i.e. the equations (2) define a congruence of straight lines, then:

- If $\langle \boldsymbol{\tau}, \text{rot} \boldsymbol{\tau} \rangle \neq 0$, the problem does not admit any nontrivial solutions.
- If $\langle \boldsymbol{\tau}, \text{rot} \boldsymbol{\tau} \rangle \equiv 0$, the general solution of the problem is given by

$$u(\mathbf{x}) = F(V(\mathbf{x})), h(\mathbf{x}) = H(\omega_1(\mathbf{x}), \omega_2(\mathbf{x})),$$

where F and H are arbitrary functions of one and two arguments, respectively; that is, the generality of the solution is $\infty^{f(2)+f(1)}$.

Geometrically, the condition $\langle \boldsymbol{\tau}, \text{rot} \boldsymbol{\tau} \rangle \equiv 0$ implies that the curves of the family (2) are orthogonal to a certain family of surfaces, namely, $V(\mathbf{x}) = \text{const.}$



Theorem 2. For $k_1 \neq 0$ and $\langle \boldsymbol{\tau}, \text{rot} \boldsymbol{\tau} \rangle \equiv 0$, the inverse problem admits nontrivial solutions, and:

- if $\langle \boldsymbol{\nu}, \text{rot} \boldsymbol{\nu} \rangle \neq 0$, then $u(\mathbf{x}) = W(\mathbf{x}) F(V(\mathbf{x})) - h$ and $h = \text{const}$, where F is an arbitrary function and h is an arbitrary constant; that is, the generality of the solution is $\infty^{f(1)+1}$;
- if $\langle \boldsymbol{\nu}, \text{rot} \boldsymbol{\nu} \rangle \equiv 0$, then

$$u(\mathbf{x}) = W(\mathbf{x}) \left(F(V(\mathbf{x})) + \int_{\delta}^{W(\mathbf{x})} \frac{H(h^*(V(\mathbf{x}), w))}{w^2} dw \right),$$

$$h(\mathbf{x}) = H(h^*(V(\mathbf{x}), W(\mathbf{x}))),$$

where F and H are arbitrary functions; that is, the generality of the solution is $\infty^{2f(1)}$.

The condition $\langle \boldsymbol{\nu}, \text{rot} \boldsymbol{\nu} \rangle \equiv 0$ admits the following geometric interpretation. This condition is satisfied if and only if there exists a family of surfaces for which the curves of the family (2) are geodesics.



In cases where either the curvature k_1 or the quantity $\langle \boldsymbol{\tau}, \text{rot} \boldsymbol{\tau} \rangle$ vanishes identically, the force function can be determined by restricting the analysis solely to first-order PDEs. It can be shown that in the general case, where $k_1 \neq 0$ and $\langle \boldsymbol{\tau}, \text{rot} \boldsymbol{\tau} \rangle \neq 0$, it is impossible to complete the system $(E_1)-(E_4)$ with new first-order PDEs. However, because this system is not involutive, one must consider second-order PDEs.

Basically, if $k_1 \neq 0$ and $\langle \boldsymbol{\tau}, \text{rot} \boldsymbol{\tau} \rangle \neq 0$, then the problem either does not admit any nontrivial solutions or admits a unique solution up to two arbitrary constants (i.e., the generality of the solution is ∞^2).



Bozis and Nakhla (1986) noted the absence of an example of a family of curves that is not a family of trajectories of a material point in a potential force field with a nontrivial force function.

Let the family of trajectories be given by the equations

$$\omega_1(\mathbf{x}) \equiv \frac{x_1}{x_2}, \omega_2(\mathbf{x}) \equiv x_2 + x_3.$$

Obviously, we have a congruence of lines and curvature $k_1 \equiv 0$. In the case under consideration,

$$\boldsymbol{\tau} = \frac{1}{\sqrt{x_1^2 + 2x_2^2}} \begin{pmatrix} -x_1 \\ -x_2 \\ x_2 \end{pmatrix}, \langle \boldsymbol{\tau}, \text{rot} \boldsymbol{\tau} \rangle = \frac{-x_1}{x_1^2 + 2x_2^2} \neq 0.$$

Thus, we have the case $k_1 \equiv 0$, $\langle \boldsymbol{\tau}, \text{rot} \boldsymbol{\tau} \rangle \neq 0$ and the problem allows only a trivial solution.



Let

$$\omega_1(\mathbf{x}) \equiv \frac{x_1}{x_2}, \omega_2(\mathbf{x}) \equiv x_3.$$

This choice corresponds to the case where $k_1 \equiv 0$ and $\langle \boldsymbol{\tau}, \text{rot} \boldsymbol{\tau} \rangle \equiv 0$.

The general solution can then be written as

$$u(\mathbf{x}) = F(x_1^2 + x_2^2), h(\mathbf{x}) = H\left(\frac{x_1}{x_2}, x_3\right),$$

where F and H are arbitrary functions of one and two arguments, respectively.

Thus, the generality of the solution is $\infty^{f(2)+f(1)}$.



For the family of curves defined by

$$\omega_1(\mathbf{x}) \equiv x_1 x_2, \omega_2(\mathbf{x}) \equiv x_1^2 + x_2^2 + x_3^2,$$

the conditions

$$k_1 \neq 0, \langle \boldsymbol{\tau}, \text{rot} \boldsymbol{\tau} \rangle \equiv 0, \langle \boldsymbol{\nu}, \text{rot} \boldsymbol{\nu} \rangle \neq 0$$

hold. The corresponding general solution is given by:

$$u(\mathbf{x}) = \frac{(x_1^2 + x_2^2) x_3^2 + (x_1^2 - x_2^2)^2}{x_3^6} F\left(\frac{x_1^2 - x_2^2}{x_3^2}\right) - h, h(\mathbf{x}) = h \in \mathbb{R},$$

where F is an arbitrary function of one argument and h is an arbitrary constant.

The generality of the solution is $\infty^{f(1)+1}$.



If

$$\omega_1(\mathbf{x}) \equiv \frac{x_1}{x_2}, \omega_2(\mathbf{x}) \equiv x_1^2 + x_2^2 + x_3^2,$$

we obtain the case

$$k_1 \neq 0, \langle \boldsymbol{\tau}, \text{rot} \boldsymbol{\tau} \rangle \equiv 0, \langle \boldsymbol{\nu}, \text{rot} \boldsymbol{\nu} \rangle \equiv 0.$$

The general solution is given by:

$$u(\mathbf{x}) = \frac{x_1^2 + x_2^2 + x_3^2}{x_3^4} \left(F\left(\frac{x_1^2 + x_2^2}{x_3^2}\right) + \int_{\delta}^{\frac{x_1^2 + x_2^2 + x_3^2}{x_3^4}} H\left(\frac{x_1^2 + x_2^2 + x_3^2}{x_3^2 w}\right) \frac{dw}{w^2} \right),$$

$$h(\mathbf{x}) = H(x_1^2 + x_2^2 + x_3^2),$$

where F and H are arbitrary functions of one argument.

The generality of the solution is $\infty^{2f(1)}$.



The family of curves defined by

$$\omega_1(\mathbf{x}) \equiv x_2^2 + x_3^2, \omega_2(\mathbf{x}) \equiv 2x_1 - \arcsin \frac{2x_2x_3}{x_2^2 + x_3^2},$$

corresponds to the case

$$k_1 \neq 0, \langle \boldsymbol{\tau}, \text{rot} \boldsymbol{\tau} \rangle \neq 0, \langle \boldsymbol{\nu}, \text{rot} \boldsymbol{\nu} \rangle \equiv 0.$$

The general solution depends on two arbitrary constants, c_1 and c_2 :

$$u(\mathbf{x}) = \frac{c_1}{1 + x_2^2 + x_3^2} - c_2, h(\mathbf{x}) = c_2.$$

If $c_1 < 0$, the motion along this family of curves is impossible; if $c_1 > 0$, the domain Ω coincides with the entire configuration space.

This example refutes G. K. Suslov's assertion regarding the necessary existence of a family of surfaces orthogonal to an isoenergetic family of trajectories.

The generality of the solution is ∞^2 .



In the neural ODE approach, the unknown force function U is assumed to be the output of a neural network $\mathcal{F}$ with inputs $\mathbf{x} \in \mathbb{R}^3$ 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 a dataset $\mathbf{D}$ of position and velocity measurements taken at specific instants as a material point moves along distinct trajectories.



We assume that observations are made over M trajectories, $\Omega^{(i)}$, $i = 1, \dots, M$, and that each trajectory $\Omega^{(i)}$ contains an identical number of observations N .

Each observation has the form $(\tilde{\mathbf{x}}_j^{(i)}, \tilde{\mathbf{x}}_j^{\prime(i)}, t_j^{(i)})$, $j = 1, \dots, N$, with $t_1^{(i)} < \dots < t_j^{(i)} < \dots < t_N^{(i)}$. Here, $\tilde{\mathbf{x}}_j^{(i)}$ is the observed position, and $\tilde{\mathbf{x}}_j^{\prime(i)}$ is the observed velocity at time $t_j^{(i)}$. We aim to train a neural network $\mathcal{F}$ on the dataset

$$\mathbf{D} = \left\{ \left(\tilde{\mathbf{x}}_j^{(i)}, \tilde{\mathbf{x}}_j^{\prime(i)}, t_j^{(i)} \right), i = 1, \dots, M, j = 1, \dots, N \right\}$$

so that the trajectories $\mathbf{x}(t; \boldsymbol{\theta})$ generated by system (4) closely match the observations.

Specifically, given the initial conditions $\mathbf{x}^{(i)}(t_1^{(i)}) = \tilde{\mathbf{x}}_1^{(i)}$ and $\mathbf{x}'^{(i)}(t_1^{(i)}) = \tilde{\mathbf{x}}_1^{\prime(i)}$, the predicted trajectories $\mathbf{x}^{(i)}(t; \boldsymbol{\theta})$ should pass through or near the remaining data points $\mathbf{D}^{(i)} = \left\{ \left(\tilde{\mathbf{x}}_j^{(i)}, \tilde{\mathbf{x}}_j^{\prime(i)}, t_j^{(i)} \right), j = 1, \dots, N \right\}$.



We can replace the system (4) of three second-order ODEs with the following system of six first-order ODEs 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 with initial conditions

$$\mathbf{x}(t_1^{(i)}) = \tilde{\mathbf{x}}_1^{(i)}, \mathbf{x}'(t_1^{(i)}) = \tilde{\mathbf{x}}_1'^{(i)}, i = 1, \dots, M,$$

we can obtain the set of trajectory points $(\mathbf{x}(t; \boldsymbol{\theta}), \mathbf{x}'(t; \boldsymbol{\theta}))$ at times $t_j^{(i)}$, $j = 2, \dots, N$.



We assume that the loss function $\mathcal{L}$ 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 observations in the dataset $\mathbf{D}$, measured in both position and velocity:

$$\mathcal{L}[\mathbf{x}(t; \boldsymbol{\theta}), \mathbf{x}'(t; \boldsymbol{\theta})] = \sum_{i=1}^M \sum_{j=1}^N \rho\left(\mathbf{x}(t_j^{(i)}; \boldsymbol{\theta}), \mathbf{x}'(t_j^{(i)}; \boldsymbol{\theta}), \tilde{\mathbf{x}}_j^{(i)}, \tilde{\mathbf{x}}_j'^{(i)}\right), \quad (6)$$

where

$$\rho\left(\mathbf{x}(t_j^{(i)}; \boldsymbol{\theta}), \mathbf{x}'(t_j^{(i)}; \boldsymbol{\theta}), \tilde{\mathbf{x}}_j^{(i)}, \tilde{\mathbf{x}}_j'^{(i)}\right) = \left\| \mathbf{x}(t_j^{(i)}; \boldsymbol{\theta}) - \tilde{\mathbf{x}}_j^{(i)} \right\|^2 + \left\| \mathbf{x}'(t_j^{(i)}; \boldsymbol{\theta}) - \tilde{\mathbf{x}}_j'^{(i)} \right\|^2.$$

To minimize the loss function $\mathcal{L}$ 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 neural ODE framework, we introduce the adjoint variables $\chi = \nabla_{\mathbf{x}} \mathcal{L}$, $\chi' = \nabla_{\mathbf{x}'} \mathcal{L}$, $\vartheta = \nabla_{\boldsymbol{\theta}} \mathcal{L}$, which represent the gradients of the loss function $\mathcal{L}$ with respect to the positions $\mathbf{x}$, the velocities $\mathbf{x}'$, and the parameters $\boldsymbol{\theta}$.

To simplify the notation, we focus on a single orbit $\Omega^{(i)}$ and drop the index i hereafter. The dynamics of the variables (gradients) $\chi(t)$, $\chi'(t)$ and $\vartheta(t)$ are governed by ODE systems in reverse time, defined on the intervals $[t_j, t_{j+1}]$, $j = N - 1, N - 2, \dots, 1$:

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

Here $\nabla_{\mathbf{x}}^2 U$ is the matrix of second derivatives (the Hess matrix) of the force function U , the vectors $\mathbf{x}_{k+1}$ and $\mathbf{x}'_{k+1}$ in the terminal conditions are obtained by integrating the system (5) in direct time from t_1 to t_N .



The extended system of first-order ODEs can be replaced by the extended system of first- and second-order ODEs

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

Here $\boldsymbol{\chi}'$ and $\boldsymbol{\vartheta}$ are the adjoint variables (gradients of the loss function) with respect to $\dot{\mathbf{x}}$ and $\boldsymbol{\theta}$.



The system of first and second order ODEs (7) in variables $\mathbf{x}(t)$, $\boldsymbol{\chi}'(t)$ and $\boldsymbol{\vartheta}(t)$ must be solved backward in time from t_N to t_1 .

Since the loss function $\mathcal{L}$ 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 $\boldsymbol{\vartheta}(t_j)$ of the loss function $\mathcal{L}$ at times $t_1, t_2, \dots, t_{N-1}$.

The gradient of the loss function $\mathcal{L}[\mathbf{x}(t; \boldsymbol{\theta})]$ with respect to $\boldsymbol{\theta}$ is obtained as the sum of the gradients $\boldsymbol{\vartheta}(t_j)$ computed by solving the systems (7) on the time intervals $[t_{N-1}, t_N]$, $[t_{N-2}, t_{N-1}]$, ..., $[t_1, t_2]$:

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

Thus, we obtain the gradient $\nabla_{\boldsymbol{\theta}} \mathcal{L}$ for a dataset of points on a single orbit. For multiple orbits, the total gradient is computed by summing (8) over all orbits.



We define the family of trajectories by equalities

$$\omega_1(\mathbf{x}) \equiv x_1 x_2 x_3 + x_1 + x_2 = c_1, \quad \omega_2(\mathbf{x}) \equiv x_3 = c_2. \quad (9)$$

Motion along curves of the family (9) is possible for the force function

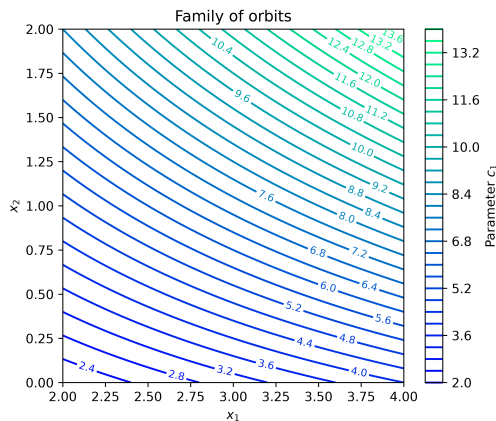
$$U(\mathbf{x}) = \frac{C_1}{(x_1 - x_2)^2} - C_2 \quad (10)$$

and the energy function

$$h(\mathbf{x}) = \frac{C_1}{2} \frac{\omega_2^2(\mathbf{x})}{1 + \omega_1(\mathbf{x}) \omega_2(\mathbf{x})} + C_2. \quad (11)$$

For $C_1 > 0$, motion is possible in the domain $1 + (x_1 x_2 x_3 + x_1 + x_2) x_3 \geq 0$, $x_1 \neq x_2$,
for $C_1 < 0$, motion is possible in the domain $1 + (x_1 x_2 x_3 + x_1 + x_2) x_3 \leq 0$, $x_1 \neq x_2$.

Orbits of the family in axes x_1 and x_2 in the plane x_3



The family of space curves (9) in $\mathbb{R}^3$ is formed by the intersection of algebraic surfaces

$$x_1 x_2 x_3 + x_1 + x_2 = c_1$$

with horizontal planes $x_3 = c_2$. Within any plane $x_3 = c_2$ where $c_2 \neq 0$, the curves in (9) are rectangular hyperbolas oriented at a 45° angle to the coordinate axes x_1 and x_2 .



We construct a synthetic dataset $\mathbf{D}$ of positions and velocities sampled at various times for a material point moving along distinct trajectories of the family (9), driven by the ODE system (1) with the force function $U(\mathbf{x})$, determined by (10), and the energy function $h(\mathbf{x})$, determined by (11).

When training a neural network $\mathcal{F}$ using stochastic gradient descent, a mini-batch is formed as follows.

We chose hyperparameters $b_S < M$ for the mini-batch size and $b_T < N$ for the number of consecutive time instants $t_j^{(i)}$ of data points from $\mathbf{D}^{(i)}$ included in the mini-batch.

Next, the trajectory indices $i_k \in \{1, \dots, M\}$, $k = 1, \dots, b_S$, were randomly selected. For each index i_k , a time moment $t_{j_k}^{(i_k)}$, $j_k \in \{1, \dots, N - b_T\}$, was randomly selected, and the points

$$\left(\mathbf{x}_{j_k}^{(i_k)}, \mathbf{x}_{j_k}'^{(i_k)}, t_{j_k}^{(i_k)} \right), \left(\mathbf{x}_{j_k+1}^{(i_k)}, \mathbf{x}_{j_k+1}'^{(i_k)}, t_{j_k+1}^{(i_k)} \right), \dots, \left(\mathbf{x}_{j_k+b_T}^{(i_k)}, \mathbf{x}_{j_k+b_T}'^{(i_k)}, t_{j_k+b_T}^{(i_k)} \right)$$

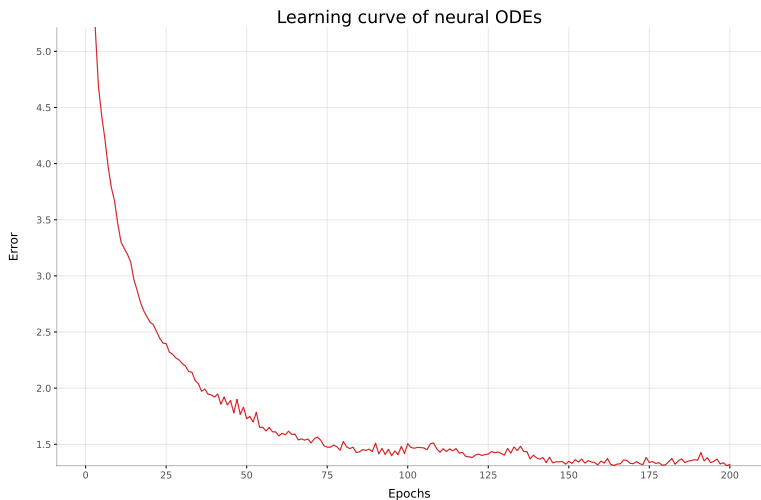
were included in the mini-batch $\mathbf{B}$.

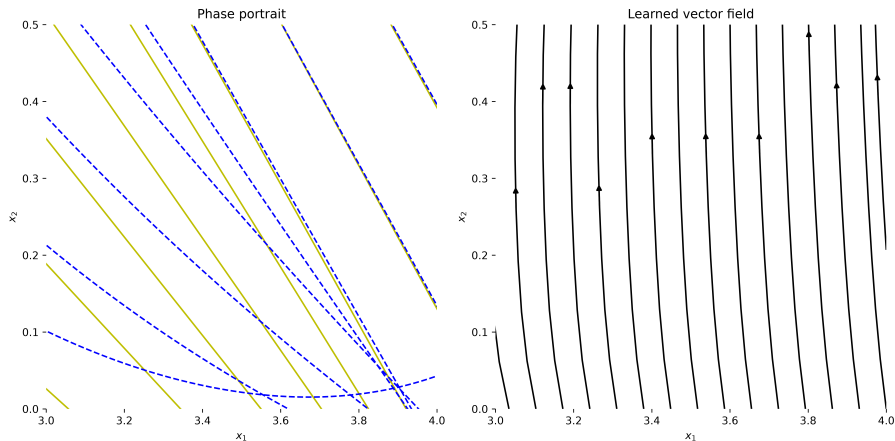


The computational experiment was conducted using a feedforward neural network with dense layers and the following architecture and hyperparameters:

- input layer contains 3 neurons representing the material point position (x_1, x_2, x_3) ,
- hidden layer contains 50 neurons with a hyperbolic tangent (tanh) activation function,
- output layer contains 1 neuron without an activation function to predict the force function $U(x_1, x_2, x_3)$,
- optimization was performed using the Adam optimizer with an initial learning rate of 10^{-3} ,
- training was performed for 200 epochs with a batch size of 100 points, randomly sampled from datasets of 500 points for each of 15 curves.

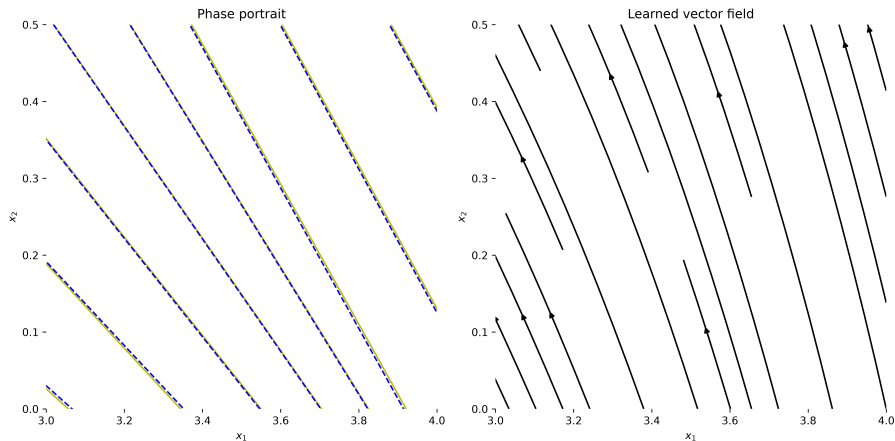
The code was implemented using the PyTorch framework and the torchdiffeq library.





Yellow solid lines are true trajectories, blue dashed lines are the trajectories of neural ODEs.

Phase portrait and learned vector field at epoch 199



Yellow solid lines are true trajectories, blue dashed lines are the trajectories of neural ODEs.



The inverse problem of celestial mechanics is evolving from a purely theoretical mathematical challenge into a dynamic, data-driven field of research.

The classical approach, based on PDEs, 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 approach 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 of a loss function 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.

Any questions?