

On neural inverse problem of celestial mechanics

Shorokhov S.G.

The 10th International Conference on Deep Learning
in Computational Physics

SINP MSU, Moscow, Russia
July 6-8, 2026





Постановка задачи:

Определить силовую функцию $U(\mathbf{x})$ потенциального силового поля, при котором материальная точка движется по траекториям из заданного семейства кривых

Ключевые этапы:

Период	Авторы	Результаты
1687	И. Ньютон	Первая обратная задача: закон тяготения по законам Кеплера
1877–1890	J. Bertrand, U. Dainelli, Н.Е. Жуковский, Г.К. Суллов	Силовая функций для конических сечений; общий вид силовой функции для плоских траекторий; явные формулы силовой функции для 2D/3D
1960–наст. время	V. Szebehely, F. Varadi, B. Erdi, G. Bozis, T. Kotoulas	ДУЧП 1-го порядка для силовой функции; геометрические подходы с редукцией к алгебраическим уравнениям и т.д.



Рассматривается обратная задача построения уравнений движения материальной точки в трехмерном пространстве в декартовой системе координат

$$\ddot{x}_i = \frac{\partial U}{\partial x_i}, \quad i = 1, 2, 3 \quad (1)$$

в потенциальном силовом поле с силовой функцией $U(x_1, x_2, x_3)$ по семейству траекторий движения, заданному соотношениями вида

$$\omega_1(x_1, x_2, x_3) = c_1, \quad \omega_2(x_1, x_2, x_3) = c_2, \quad c_1, c_2 \in \mathbb{R}. \quad (2)$$

Предполагается, что выполняется условие максимальности ранга матрицы $\frac{\partial(\omega_1, \omega_2)}{\partial(x_1, x_2, x_3)}$ (условие регулярности). Требуется найти силовую функцию $U(x_1, x_2, x_3)$, при которой возможно движение по кривым семейства (2).



При движении материальной точки в потенциальном силовом поле будет соблюдаться интеграл энергии

$$\frac{1}{2} (\dot{x}_1^2 + \dot{x}_2^2 + \dot{x}_3^2) - U(x_1, x_2, x_3) = h.$$

Движение по кривым семейства (2) может осуществляться при различном уровне энергии h , поэтому функция энергии $h(x_1, x_2, x_3)$ будет постоянной на траекториях семейства (2):

$$h(x_1, x_2, x_3) = h(\omega_1(x_1, x_2, x_3), \omega_2(x_1, x_2, x_3)).$$

Это условие можно записать при помощи линейного ДУЧП первого порядка вида

$$(\nabla\omega_1, \nabla\omega_2, \nabla h) = 0,$$

где круглые скобки обозначают смешанное произведение векторов, ∇ – операция вычисления градиента по $\mathbf{x}$.



Для решения обратной задачи небесной механики необходимо исследовать совместность и найти решение следующей системы уравнений с частными производными первого порядка относительно неизвестных функций $U(\mathbf{x})$ и $h(\mathbf{x})$:

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

где $\mathbf{s} = \nabla \omega_1 \times \nabla \omega_2$, $E = \frac{1}{|\mathbf{s}|^2} \langle \mathbf{s}, \operatorname{rots} \rangle$, $\mathbf{p} = \frac{1}{|\mathbf{s}|^2} (\nabla^T \mathbf{s}) \mathbf{s}$, $\mathbf{q} = \mathbf{p} \times \mathbf{s}$. Здесь угловыми скобками обозначено скалярное произведение в $\mathbb{R}^3$, операция $\times$ обозначает векторное произведение в $\mathbb{R}^3$.



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.



Рассмотрим систему ОДУ второго порядка

$$\ddot{\mathbf{x}} = \nabla_{\mathbf{x}} U(\mathbf{x}, \boldsymbol{\theta}), \mathbf{x} = (x_1, x_2, x_3), \dot{\mathbf{x}} = (\dot{x}_1, \dot{x}_2, \dot{x}_3), \ddot{\mathbf{x}} = (\ddot{x}_1, \ddot{x}_2, \ddot{x}_3) \quad (3)$$

с начальными условиями:

$$\begin{cases} \mathbf{x}(t_0) = \mathbf{x}_0 = (x_1^{(0)}, x_2^{(0)}, x_3^{(0)}), \\ \dot{\mathbf{x}}(t_0) = \mathbf{x}'_0 = (x_1'^{(0)}, x_2'^{(0)}, x_3'^{(0)}), \end{cases} \quad (4)$$

где силовая функция $U(\mathbf{x}, \boldsymbol{\theta})$ в правой части системы ОДУ (3) представляет собой выходное значение нейронной сети $\mathcal{F}$ с параметрами (весами) $\boldsymbol{\theta} \in \mathbb{R}^p$ и входными значениями $\mathbf{x} = (x_1, x_2, x_3)$.

Градиент силовой функции $U(\mathbf{x}, \boldsymbol{\theta})$ по переменным $\mathbf{x}$ в уравнениях (3) вычисляется при помощи механизма точного автоматического дифференцирования, реализованного в современных фреймворках глубокого обучения, таких, как TensorFlow, PyTorch, JAX или MLX.



Требуется обучить нейронную сеть $\mathcal{F}$ на имеющемся наборе точек данных $\mathbf{D} = \left\{ \left(x_1^{(i)}, x_2^{(i)}, x_3^{(i)}, x_1'^{(i)}, x_2'^{(i)}, x_3'^{(i)}, t_i \right), i = \overline{1, N} \right\}$ так, чтобы траектории $\mathbf{x}(t; \boldsymbol{\theta})$ системы нейродифференциальных уравнений (3) максимально соответствовали имеющемуся набору наблюдений $\mathbf{D}$ (траектории $\mathbf{x}(t; \boldsymbol{\theta})$ проходили через точки набора $\mathbf{D}$ или были в непосредственной близости от этих точек).

Заменим систему ОДУ (3) из трех уравнений второго порядка следующей системой из шести уравнений первого порядка относительно переменных $(\mathbf{x}, \mathbf{x}') = (x_1, x_2, x_3, x_1', x_2', x_3')$:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{x}', & \mathbf{x}(t_0) = \mathbf{x}_0 = \left(x_1^{(0)}, x_2^{(0)}, x_3^{(0)} \right), \\ \dot{\mathbf{x}}' = \nabla_{\mathbf{x}} U(\mathbf{x}, \boldsymbol{\theta}), & \mathbf{x}'(t_0) = \mathbf{x}'_0 = \left(x_1'^{(0)}, x_2'^{(0)}, x_3'^{(0)} \right). \end{cases} \quad (5)$$

Интегрируя систему (5) при помощи какого-либо численного алгоритма, можем получить фактический набор точек траектории $(\mathbf{x}(t; \boldsymbol{\theta}), \mathbf{x}'(t; \boldsymbol{\theta}))$ для моментов времени $t_k, k = \overline{1, N}$: $\mathbf{x}_k = \mathbf{x}(t_k; \boldsymbol{\theta}), \mathbf{x}'_k = \mathbf{x}'(t_k; \boldsymbol{\theta})$.



Допустим, что функционал потерь для обучения нейронной сети $\mathcal{F}$ имеет вид суммы оценок отклонений траектории $\mathbf{x}(t; \boldsymbol{\theta})$ от точек набора наблюдений $\mathbf{D}$ по положению и по скорости:

$$\begin{aligned} \mathcal{L}[\mathbf{x}(t; \boldsymbol{\theta}), \mathbf{x}'(t; \boldsymbol{\theta})] &= \sum_{k=1}^N \rho(\mathbf{x}(t_k; \boldsymbol{\theta}), \mathbf{x}'(t_k; \boldsymbol{\theta}), \mathbf{x}^{(k)}, \mathbf{x}'^{(k)}) = \\ &= \sum_{k=1}^N \left(\|\mathbf{x}(t_k; \boldsymbol{\theta}) - \mathbf{x}^{(k)}\|^2 + \|\mathbf{x}'(t_k; \boldsymbol{\theta}) - \mathbf{x}'^{(k)}\|^2 \right). \end{aligned} \quad (6)$$

Чтобы минимизировать потери $\mathcal{L}$ при помощи алгоритма градиентного спуска в пространстве параметров $\boldsymbol{\theta}$ нейронной сети $\mathcal{F}$, нужно знать градиент $\mathcal{L}$ по $\boldsymbol{\theta}$. Введем сопряженные (adjoint) переменные $\boldsymbol{\chi}(t)$, $\boldsymbol{\chi}'(t)$ и $\boldsymbol{\vartheta}(t)$:

$$\boldsymbol{\chi}(t) = \frac{\partial \mathcal{L}}{\partial \mathbf{x}(t)}, \quad \boldsymbol{\chi}'(t) = \frac{\partial \mathcal{L}}{\partial \mathbf{x}'(t)}, \quad \boldsymbol{\vartheta}(t) = \frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}}, \quad (7)$$

которые представляют собой градиенты функционала потерь $\mathcal{L}$ относительно состояний $\mathbf{x}$, скоростей $\mathbf{x}'$ и параметров $\boldsymbol{\theta}$ в момент времени $t \in [t_0, T_N]$.



Согласно работе R. T. Q. Chen et al. “Neural Ordinary Differential Equations”, 2018, динамика переменных (градиентов) $\mathbf{x}(t)$, $\mathbf{x}'(t)$ и $\boldsymbol{\vartheta}(t)$ задается системами ОДУ в обратном времени на отрезках $[t_k, t_{k+1}]$, $k = N - 1, N - 2, \dots, 0$:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{x}', & \mathbf{x}(t_{k+1}) = \mathbf{x}_{k+1}, \\ \dot{\mathbf{x}}' = \nabla_{\mathbf{x}} U(\mathbf{x}, \boldsymbol{\theta}), & \mathbf{x}'(t_{k+1}) = \mathbf{x}'_{k+1}, \\ \dot{\boldsymbol{\chi}} = -\nabla_{\mathbf{x}}^2 U(\mathbf{x}, \boldsymbol{\theta}) \boldsymbol{\chi}', & \boldsymbol{\chi}(t_{k+1}) = \nabla_{\mathbf{x}} \rho(\mathbf{x}_{k+1}, \mathbf{x}'_{k+1}, \mathbf{x}^{(k+1)}, \mathbf{x}'^{(k+1)}), \\ \dot{\boldsymbol{\chi}}' = -\boldsymbol{\chi}, & \boldsymbol{\chi}'(t_{k+1}) = \nabla_{\mathbf{x}'} \rho(\mathbf{x}_{k+1}, \mathbf{x}'_{k+1}, \mathbf{x}^{(k+1)}, \mathbf{x}'^{(k+1)}), \\ \dot{\boldsymbol{\vartheta}} = -\nabla_{\boldsymbol{\theta}} (\nabla_{\mathbf{x}}^T U(\mathbf{x}, \boldsymbol{\theta}) \boldsymbol{\chi}'), & \boldsymbol{\vartheta}(t_{k+1}) = \mathbf{0}. \end{cases} \quad (8)$$

Здесь $\nabla_{\mathbf{x}}^2 U$ – матрица вторых производных (матрица Гесса) силовой функции U , векторы $\mathbf{x}_{k+1}$ и $\mathbf{x}'_{k+1}$ в терминальных условиях получаются при интегрировании системы (5) в прямом времени от t_0 до t_N .



Из системы (8) вытекает, что $\chi = -\dot{\chi}'$, поэтому $\dot{\chi} = -\ddot{\chi}'$ и подсистема ОДУ первого порядка

$$\begin{cases} \dot{\chi} = -\nabla_{\mathbf{x}}^2 U(\mathbf{x}, \boldsymbol{\theta}) \chi', & \chi(t_{k+1}) = \nabla_{\mathbf{x}} \rho(\mathbf{x}_{k+1}, \mathbf{x}'_{k+1}, \mathbf{x}^{(k+1)}, \mathbf{x}'^{(k+1)}), \\ \dot{\chi}' = -\chi, & \chi'(t_{k+1}) = \nabla_{\mathbf{x}'} \rho(\mathbf{x}_{k+1}, \mathbf{x}'_{k+1}, \mathbf{x}^{(k+1)}, \mathbf{x}'^{(k+1)}), \end{cases}$$

может быть заменена на подсистему ОДУ второго порядка

$$\begin{cases} \ddot{\chi}' = \nabla_{\mathbf{x}}^2 U(\mathbf{x}, \boldsymbol{\theta}) \chi', & \chi'(t_{k+1}) = \nabla_{\mathbf{x}'} \rho(\mathbf{x}_{k+1}, \mathbf{x}'_{k+1}, \mathbf{x}^{(k+1)}, \mathbf{x}'^{(k+1)}), \\ \dot{\chi}'(t_{k+1}) = -\nabla_{\mathbf{x}} \rho(\mathbf{x}_{k+1}, \mathbf{x}'_{k+1}, \mathbf{x}^{(k+1)}, \mathbf{x}'^{(k+1)}). \end{cases}$$



Итак, расширенная система ОДУ (8) первого порядка может быть заменена на расширенную систему ОДУ первого и второго порядков

$$\left\{ \begin{array}{ll} \ddot{\mathbf{x}} = \nabla_{\mathbf{x}} U(\mathbf{x}, \boldsymbol{\theta}), & \mathbf{x}(t_{k+1}) = \mathbf{x}_{k+1}, \\ & \dot{\mathbf{x}}(t_{k+1}) = \mathbf{x}'_{k+1}, \\ \ddot{\boldsymbol{\chi}}' = \nabla_{\mathbf{x}}^2 U(\mathbf{x}, \boldsymbol{\theta}) \boldsymbol{\chi}', & \boldsymbol{\chi}'(t_{k+1}) = \nabla_{\mathbf{x}'} \rho(\mathbf{x}_{k+1}, \mathbf{x}'_{k+1}, \mathbf{x}^{(k+1)}, \mathbf{x}'^{(k+1)}), \\ & \dot{\boldsymbol{\chi}}'(t_{k+1}) = -\nabla_{\mathbf{x}} \rho(\mathbf{x}_{k+1}, \mathbf{x}'_{k+1}, \mathbf{x}^{(k+1)}, \mathbf{x}'^{(k+1)}), \\ \dot{\boldsymbol{\vartheta}} = -\nabla_{\boldsymbol{\theta}} (\nabla_{\mathbf{x}}^T U(\mathbf{x}, \boldsymbol{\theta}) \boldsymbol{\chi}'), & \boldsymbol{\vartheta}(t_{k+1}) = \mathbf{0}. \end{array} \right. \quad (9)$$

Здесь $\boldsymbol{\chi}'$ и $\boldsymbol{\vartheta}$ – сопряженные переменные (градиенты функционала потерь) по $\dot{\mathbf{x}}$ и $\boldsymbol{\theta}$.

Подобно обратному распространению ошибки систему ОДУ первого порядка (8) относительно переменных $\mathbf{x}(t)$, $\mathbf{x}'(t)$, $\boldsymbol{\chi}(t)$, $\boldsymbol{\chi}'(t)$ и $\boldsymbol{\vartheta}(t)$ или систему ОДУ (9) первого и второго порядков относительно переменных $\mathbf{x}(t)$, $\boldsymbol{\chi}'(t)$ и $\boldsymbol{\vartheta}(t)$ необходимо решать в обратном направлении во времени от t_N до t_0 .



Так как функционал $\mathcal{L}$ зависит от положения и скорости в последний момент времени t_N и зависит также от положений и скоростей в промежуточные моменты времени $t_1, t_2, \dots, t_{N-1}$, то можно решать систему уравнений (9) в обратном времени на каждом из интервалов $[t_{N-1}, t_N]$, $[t_{N-2}, t_{N-1}]$ и т.д. в обратном порядке и в конце просуммировать полученные градиенты функционала потерь $\vartheta(t_k)$ для моментов времени $t_0, t_1, \dots, t_{N-1}$.

Таким образом, градиент функционала потерь $\mathcal{L}[\mathbf{x}(t; \boldsymbol{\theta})]$ по $\boldsymbol{\theta}$ получается как сумма значений градиентов, полученных решением систем (9) на интервалах времени $[t_{N-1}, t_N]$, $[t_{N-2}, t_{N-1}]$, ..., $[t_0, t_1]$:

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}[\mathbf{x}(t; \boldsymbol{\theta})] = \sum_{k=0}^{N-1} \vartheta(t_k; \boldsymbol{\theta}) \quad (10)$$



Пусть семейство траекторий задается соотношениями

$$\omega_1(x_1, x_2, x_3) \equiv x_1 x_2 x_3 + x_1 + x_2 = c_1, \quad \omega_2(x_1, x_2, x_3) \equiv x_3 = c_2 \quad (11)$$

Тогда можно показать, что для

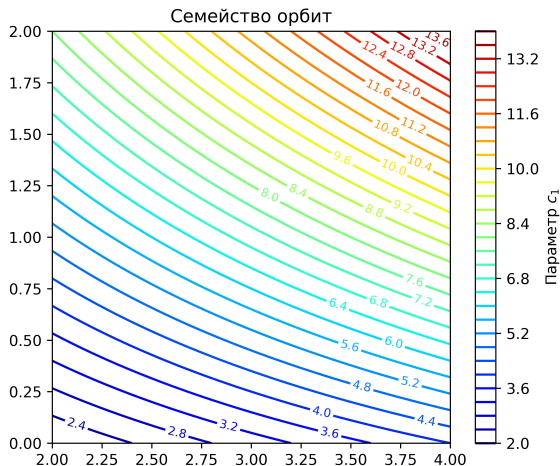
$$U(\mathbf{x}) = \frac{C_1}{(x_1 - x_2)^2} - C_2,$$

$$h(\mathbf{x}) = \frac{C_1}{2} \frac{x_3^2}{1 + (x_1 x_2 x_3 + x_1 + x_2) x_3} + C_2 = \frac{C_1}{2} \frac{\omega_2^2}{1 + \omega_1 \omega_2} + C_2$$

возможно движение по кривым семейства (11).

При $C_1 > 0$ движение возможно в области $1 + (x_1 x_2 x_3 + x_1 + x_2) x_3 \geq 0$, а при $C_1 < 0$ движение возможно в области $1 + (x_1 x_2 x_3 + x_1 + x_2) x_3 \leq 0$.

Используя функции $U(\mathbf{x})$ и $h(\mathbf{x})$, построим набор точек данных для семейства траекторий (11) в квадрате $[2, 4] \times [0, 2]$.



- Все кривые семейства — это гиперболы с общим центром в точке $(-1, -1)$.
- Асимптоты у всех гипербол одни и те же: прямые $x_1 = -1$ и $x_2 = -1$
- Семейство кривых симметрично относительно прямой $x_2 = x_1$



Пусть имеется набор точек данных $\mathbf{D}$, представляющий собой измерения положений и скоростей материальной точки в некоторые моменты времени при прохождении материальной точки по отдельным траекториям. Будем предполагать, что производились наблюдения за движением материальной точки по M траекториям $\Omega^{(i)}, i = \overline{1, M}$, причем для простоты будем считать, что для каждой из траекторий движения $\Omega^{(i)}$ сделано одинаковое количество N наблюдений вида $(\mathbf{x}_j^{(i)}, \mathbf{x}_j^{\prime(i)}, t_j^{(i)}), j = \overline{1, N}$, где $t_1^{(i)} < \dots < t_j^{(i)} < \dots < t_N^{(i)}$. Требуется обучить нейронную сеть $\mathcal{F}$ на имеющемся наборе точек данных $\mathbf{D} = \left\{ \left(\mathbf{x}_j^{(i)}, \mathbf{x}_j^{\prime(i)}, t_j^{(i)} \right), i = \overline{1, M}, j = \overline{1, N} \right\}$ так, чтобы траектории $\mathbf{x}(t; \boldsymbol{\theta})$ системы нейродифференциальных уравнений (5) максимально соответствовали имеющемуся набору наблюдений $\mathbf{D}$ в следующем смысле: траектории $\mathbf{x}^{(i)}(t; \boldsymbol{\theta})$ системы ОДУ (3) с начальными условиями $\mathbf{x}^{(i)}(t_1^{(i)}) = \mathbf{x}_1^{(i)}, \mathbf{x}^{\prime(i)}(t_1^{(i)}) = \mathbf{x}_1^{\prime(i)}$, проходили через точки данных $\mathbf{D}^{(i)} = \left\{ \left(\mathbf{x}_j^{(i)}, \mathbf{x}_j^{\prime(i)}, t_j^{(i)} \right), j = \overline{1, N} \right\}$ или были в непосредственной близости от этих точек.

Поэтому при обучении нейронной сети $\mathcal{F}$ можно использовать функционал потерь

$$\mathcal{L}[\mathbf{x}(t; \boldsymbol{\theta}), \mathbf{x}'(t; \boldsymbol{\theta})] = \sum_{i=1}^M \sum_{j=1}^N \left(\left\| \mathbf{x}^{(i)}(t_j^{(i)}; \boldsymbol{\theta}) - \mathbf{x}_j^{(i)} \right\|^2 + \left\| \mathbf{x}'^{(i)}(t_j^{(i)}; \boldsymbol{\theta}) - \mathbf{x}_j'^{(i)} \right\|^2 \right)$$

При обучении нейронной сети $\mathcal{F}$ при помощи стохастического градиентного спуска мини-пакет $\mathbf{B}$ формируется следующим образом. Выбираются гиперпараметры $b_S < M$ для размера мини-пакета и $b_T < N$ для количества последовательных моментов времени $t_j^{(i)}$ для точек данных из $\mathbf{D}^{(i)}$, входящих в мини-пакет. Далее случайным образом выбираются индексы траекторий $i_k \in \{1, \dots, M\}$, $k = \overline{1, b_S}$ и для каждого индекса i_k случайным образом выбирается момент времени $t_{j_k}^{(i_k)}$, $j_k \in \{1, \dots, N - b_T\}$ и в мини-пакет $\mathbf{B}$ включаются точки

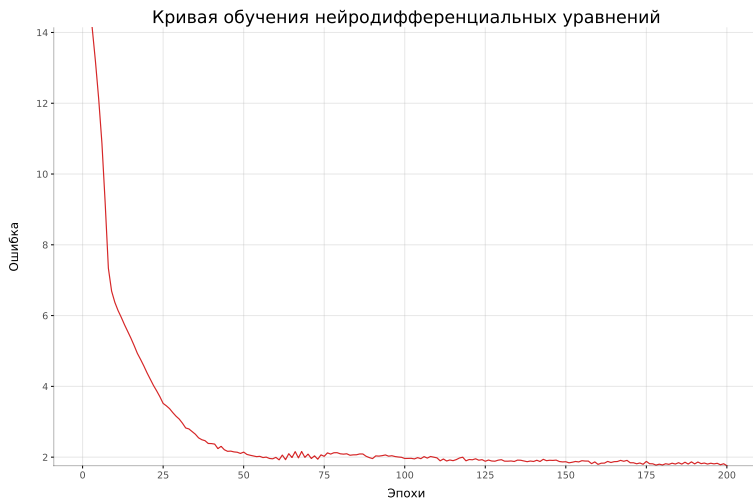
$$\left(\mathbf{x}_{j_k}^{(i_k)}, \mathbf{x}_{j_k}'^{(i_k)}, t_{j_k}^{(i_k)} \right), \left(\mathbf{x}_{j_k+1}^{(i_k)}, \mathbf{x}_{j_k+1}'^{(i_k)}, t_{j_k+1}^{(i_k)} \right), \dots, \left(\mathbf{x}_{j_k+b_T}^{(i_k)}, \mathbf{x}_{j_k+b_T}'^{(i_k)}, t_{j_k+b_T}^{(i_k)} \right).$$

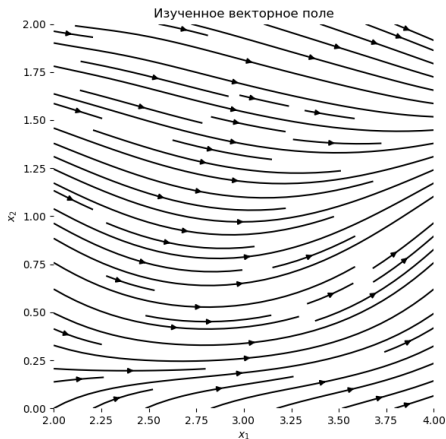
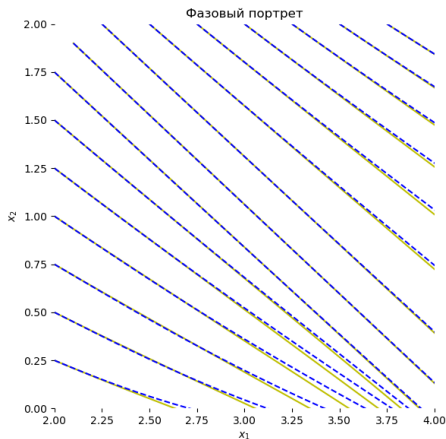


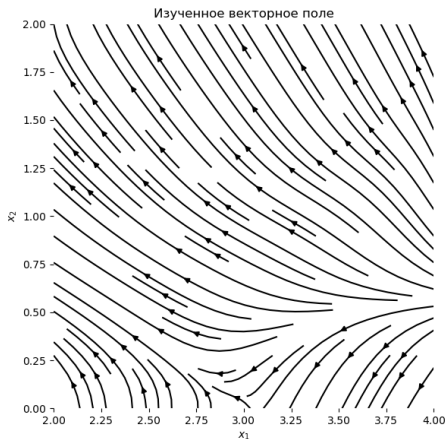
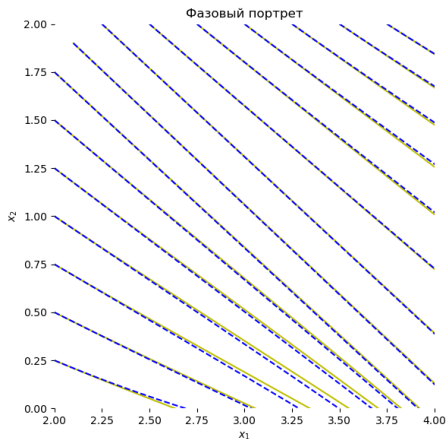
Вычислительный эксперимент выполняется для архитектуры, указанной ниже, со следующими гиперпараметрами:

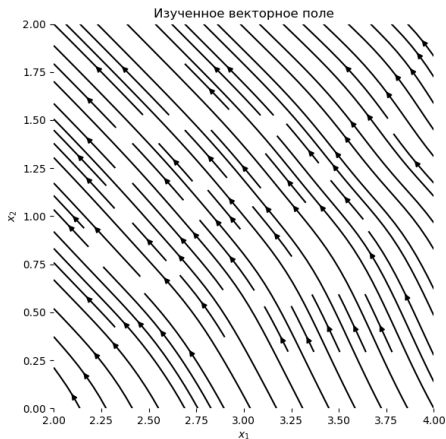
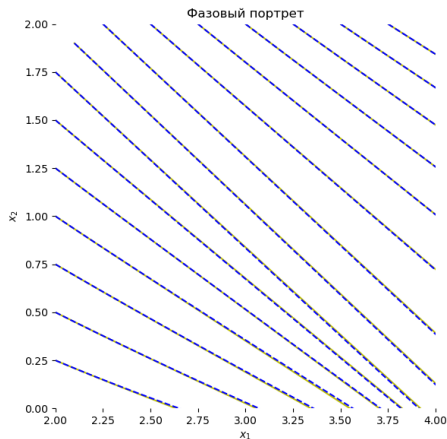
- нейронная сеть прямого распространения с плотными слоями
- входной слой с тремя нейронами (для x_1, x_2, x_3)
- один скрытый слой с функцией активации $\tanh$
- число нейронов в скрытом слое равно 50
- один нейрон без активации в выходном слое (для $U(x_1, x_2, x_3)$)
- выполняется 200 эпох обучения
- используется оптимизатор Adam с начальной скоростью обучения 10^{-3}
- пакет состоит из 100 точек, которые выбираются случайным образом из 500 точек на 15 кривых

Программный код реализован при помощи фреймворка PyTorch и библиотеки torchdiffeq.











- Сформулирована нейросетевая версия обратной задачи небесной механики: силовая функция $U(\mathbf{x})$ параметризуется нейронной сетью с обучением по дискретным наблюдениям.
- Предложена эффективная схема вычисления градиентов функционала потерь через сопряжённую систему ОДУ: интегрирование в обратном времени с использованием к уравнений 2-го порядка.
- Численный эксперимент подтвердил сходимость обучения: нейронная сеть восстанавливает структуру потенциального поля по выборкам положений и скоростей без априорного задания энергетической функции $h(\mathbf{x})$.

Вопросы?