Inverse Eigenvalue Problem on Quantum Mechanics

Fabián Trigo[1,*]

[1] *Instituto de Física y Astronomía, Universidad de Valparaíso, Avenida Gran Bretaña 1111, Valparaíso, Chile*

In this work, we delve on the inverse problem of predicting a force behind a physics phenomena, for that a trainable activation function is included and the weights build a discrete numerical differential equation integrator

keywords: numerical, inverse physics, astrophysics, quantum mechanics

PACS numbers:

Contents

I. INTRODUCTION AND MOTIVATION

Being able to understand the response of a quantum system provides insight into its structure; at an undergraduate level we begin quantum mechanics by calculating eigenfunctions whichs correspond to basic oscilation of the system that can be summed up to produce all posible combinations of states; and to calculate eigenvalues which correspond to a measurable quantity such as energy or angular momentum. There exist numerical approaches to calculate the eigenfunctions and eigenvalues (which we will denote by ψ, λ respectibly) but recent articles make use of neural networks that evolve as to provide continuos and differentiable solutions. Training this machine learning models can be expensive so its required the use of parallel computer and use of GPU

Based on the article on the simplectic integrator concept and with the help of [1], we are interested in modeling $z \in \mathbb{R}^D$ having ($D = 2d$) and its evolution on time

$$\dot{\vec{z}} = \begin{pmatrix} \dot{q_1} \\ \cdots \\ \dot{q_d} \\ \dot{p_1} \\ \cdots \\ \dot{p_d} \end{pmatrix} = \begin{pmatrix} \partial_{p_1}\mathcal{H} \\ \cdots \\ \partial_{p_d}\mathcal{H} \\ -\partial_{q_1}\mathcal{H} \\ \cdots \\ -\partial_{q_d}\mathcal{H} \end{pmatrix} \tag{1}$$

commonly this expresion with the help of another matrix, where $\hat{0}$ and $\hat{1}$ are 0 matrix and unitary matrix of size $d \times d$ each one

$$J = \begin{pmatrix} \hat{0} & \hat{1} \\ \hat{-1} & \hat{0} \end{pmatrix} \tag{2}$$

*Electronic address: fabian.trigo@alumnos.uv.cl

This makes use of the change of signs in the above expression, and making use of

$$\nabla_z \mathcal{H}(z) = \frac{\partial \mathcal{H}}{\partial z} = \begin{pmatrix} \partial_{p_1}\mathcal{H} \\ \cdots \\ \partial_{p_d}\mathcal{H} \\ -\partial_{q_1}\mathcal{H} \\ \cdots \\ -\partial_{q_d}\mathcal{H} \end{pmatrix} \tag{3}$$

so we can re construct (1) as a gradient times a matrix:

$$\dot{\vec{z}} = J \cdot \nabla_z \mathcal{H}(z) \tag{4}$$

Things can be simplified considering most common systems $\mathcal{H} = T(p_i) + V(q_i)$, such that Euler simplectic integrator is specified

$$\begin{aligned} q_i^{n+1} &= q_i^n + \Delta t \frac{\partial T(p_i^n)}{\partial p_i^n} \\ p_i^{n+1} &= p_i^n - \Delta t \frac{\partial V(q_i^n)}{\partial q_i^n} \end{aligned} \tag{5}$$

Then solution of z can be written parametrically

$$\vec{z}(t) = \vec{z}(0) + g(t)\,\vec{N}(t) \tag{6}$$

such as $g(t=0) = 0$, this parametric function $g(t)$ allows for fixing up the restriction for the initial value problem

II. REPRESENTING UNKNOWNS

The main objective of the system is to provide with a good approximation on the shapes of T, V, they are $d-$dimensional functions, for basic examples this model will be build on 2 dimensions; and applying the kinetic energy as a funcion of $T(p_x, p_y) = \frac{1}{2}(p_x^2 + p_y^2)$ the potential will be left as a function $V(x, y)$. The objective is to obtain a function for $V(x, y)$.

A. Discrete Grid

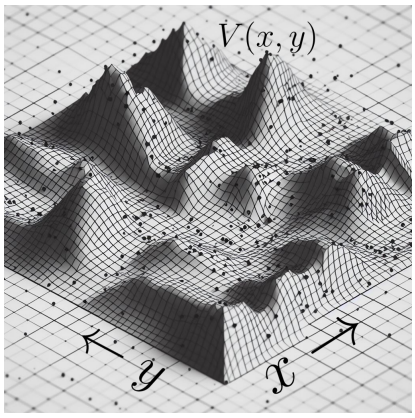

FIG. 1: Discrete grid structure generated by Dalle-3, LaTex code was edited in upper layers

Previous attemps were made with grid like data structure, storing a height map for every (x, y) point, interpolating this points for in between data and therefore extracting values for V given the point; having a grid like also allows for trivial derivatives.

B. Orthogonal functions

Another idea used was to train the system with the aid of orthogonality, such as Legendre Polynomials, Sines and Cosines, etc...

III. THE FORWARD PROBLEM

Lets have a problem with m eigenvalues known and listed, different potential produce a different set of eigenvalues; we will be listing different methods and they will be benchmark againts each other

For a quantum mechanics problem we have the wave equation with ψ_n

$$\mathcal{H}\psi_n = \lambda_n \psi_n \tag{7}$$

since $\mathcal{H}$ includes differential operations, they can be approximated by discrete derivative formulas

A. Discrete Eigenvalues

where we can quickly generate the first eigenvalues by calculating the discrete eigenvalues and eigenfunctions of the Hamiltonian $\mathcal{H}$. As it can be shown this process introduces numerical errors, and the more eigenvalues are calculated it deviates more and more from true eigenvalues; so only the few first can be accounted correclty for.

A known problem is the existence of divergence points such as a potential thats inversely proportional to the radius $V \propto \frac{1}{r}$, this proves that singular point need special treatment

The objective is to explore literature and generate efficiently and with less possible numerical error a set of eigenvalues using linear algebra and gpu parallel computing

B. Convolutional NN

A good a idea is to teach the system to produce a list of eigenvalues witha given potential,

$$V_{i,j} \rightarrow \text{CNN} \rightarrow \{\lambda_n\} \tag{8}$$

then invert the network to do the opposite, have a list of eigenvalues and generate the potential. This ideas are highly inspired in how computers dream

The objective is to use the recent advancements of convolutional neural networks and generative techniques to beat the discrete calculation algorithm and be able to generate a set of eigenvalues with lower error

---

[1] M. Mattheakis, D. Sondak, A. S. Dogra, and P. Protopapas, "Hamiltonian neural networks for solving equations of motion," *Phys. Rev. E*, vol. 105, p. 065305, Jun 2022.