# Preparatory Materials for Symplectic Neural Networks

Josh Burby, Nathan Garland, Amy Lovell, Qi Tang and Mark Zammit

August 9, 2021

# Contents

# Chapter 1

# Motivation and goals

## 1.1 Motivation

A structure-preserving time integration is very challenging. There are many recent works on very specific approaches such as maximum-principle-preserving [1], positivity-preserving, energy-preserving [2], area-preserving, flux-preserving, etc. Some discussions on this topic in the plasma physics can be found in the review article [3] and the book [4]. Most of them however are designed for very specific problems with some tweaking of existing numerical schemes. It is hard to propose a general framework to preserve structure in the conventional approaches. One common issue in the conventional numerical approaches is it is not very flexible after a spatial representation basis is chosen (finite element, finite difference, discontinuous Galerkin etc). For instance, for an ideal MHD equation, it is almost impossible to be divergence-free, positivity-preserving, and conservative at the same time in a numerical solution. It is common to sacrify some of the properties for others, depending on the applications.

In many systems, there is a stronger property, symplecticity, that implies other properties such as energy conservation. It is however even more challenging to embed symplecticity in conventional approaches since it is a stronger property and it is closely related to both space and time discretization. A reasonable approach might be a space-time discretization where the symplecticity is built into the approximation basis. This is still very challenging if not impossible in the conventional Galerkin-based approaches. The common approach along this line is through symplectic integrators [4] but there are only limited applications of symplectic integrators beyond some ODE dynamics. On the other hand, the neural networks provide nonlinear approximations to arbitrary continuous functions with a lot of flexibility thanks to its design, and we found it is easy to build symplecticity directly into the network architecture. We thus plan to study a general structure-preserving framework based on symplectic neural networks.

## 1.2 Scientific goals

The goal of the current project is to propose a general framework to study Hamiltonian systems through symplectic neural networks (HenonNet). We plan to propose the framework in the following aspects:

- approximate flow maps in parameterized high-dimensional Hamiltonian systems
- approximate parameterized B fields in fusion devices
- learning dynamics in PDEs with underlying Hamiltonian structure
- symplectic reduced-order modeling (ROM)
- bifurcation analysis
- inverse scattering problems
- ...

## 1.3 Code development

The current project requires some developments of ML-based codes as well as some communication tools with existing codes to achieve data for training. There are several network structures we would like to build.

We also plan to explore model parallelism through parallel-in-layer. Since such a framework is currently missing in the ML packages, we may need to implement such a parallelism through conventional packages such as PETSc or hypre.

## 1.4 Limitation and risks

There are some known limitations of the current network framework.

- It is only applicable to non-dissipative Hamiltonian systems. It is not clear how to handle a general dissipation or how to be structure-preserving for general dynamics if there is no Hamiltonian structure.
- The training can be extremely expensive or impossible for high-dimensional problems (PDEs, high dimensional parameter space, etc). We have observed this when we use other similar network architecture to train the Pioncare maps, and the loss for those networks is not decaying to the desired accuracy. HenonNet did a lot better thanks to the design, but we expect same thing can happen with HenonNet when we increase problem dimensions.

# Chapter 2

# Hamiltonian system

## 2.1 Symplectic vector spaces

Physically, quantum mechanics generalizes the classical mechanics of Newton. However, mathematically, classical mechanics may be seen as a generalization of quantum mechanics. To understand why, we will introduce the notion of a symplectic vector space.

To begin, consider the dynamical arena for a quantum system, i.e. a complex Hilbert space $\mathcal{H}$. Denote the inner product of vectors $\psi_1, \psi_2 \in \mathcal{H}$ using the abbreviated Dirac notation $\langle \psi_1, \psi_2 \rangle$. For concreteness, we require linearity in $\psi_2$ and antilinearity in $\psi_1$. The inner product assigns a complex number to each pair of vectors in $\mathcal{H}$. We may therefore assign *real* numbers to pairs of vectors by extracting the real and imaginary parts of the inner product:

$$\omega(\psi_1, \psi_2) = \operatorname{Im} \langle \psi_1, \psi_2 \rangle \tag{2.1}$$

$$\alpha(\psi_1, \psi_2) = \operatorname{Re} \langle \psi_1, \psi_2 \rangle. \tag{2.2}$$

The imaginary part in particular satisfies the properties

$$\omega(\psi_1, \psi_2) = -\omega(\psi_2, \psi_1) \tag{2.3}$$

$$\omega(\psi, \psi_1 + c\psi_2) = \omega(\psi, \psi_1) + c\,\omega(\psi, \psi_2) \tag{2.4}$$

where $c \in \mathbb{R}$ is an arbitrary real (not complex!) constant. Therefore the space $\mathcal{H}$ regarded as a real vector space naturally comes equipped with skew-symmetric bilinear form $\omega$.

This skew form $\omega$ gives an alternative way to study the geometry of $\mathcal{H}$ that complements the picture offered by the (Hermetian) symmetric form $\langle \cdot, \cdot \rangle$. For example, consider the characterization of the zero vector given by the inner product. A vector $\psi$ is zero if and only if $\langle \psi, \varphi \rangle = 0$ for all $\varphi \in \mathcal{H}$. The zero vector may also be characterized in terms of the skew form $\omega$ in a similar manner.

**Lemma 2.1.1.** A vector $\psi \in \mathcal{H}$ is equal to the zero vector if and only if $\omega(\psi, \varphi) = 0$ for all $\varphi \in \mathcal{H}$.

*Proof.* It is obvious that if $\psi = 0$ then $\omega(\psi, \varphi) = 0$ for each $\varphi \in \mathcal{H}$.

So suppose that $\psi$ is any vector in $\mathcal{H}$ that satisfies $\omega(\psi, \varphi) = 0$ for each $\varphi \in \mathcal{H}$. This means that for each $\varphi$ the imaginary part of $\langle \psi, \varphi \rangle$ vanishes. In other words, $\langle \psi, \varphi \rangle$ is real for each $\varphi$. In particular, if $i = \sqrt{-1}$ and $\varphi \in \mathcal{H}$ then $0 = \operatorname{Im} \langle \psi, i\varphi \rangle = \operatorname{Im}(i \langle \psi, \varphi \rangle) = \langle \psi, \varphi \rangle$. We conclude that $\langle \psi, \varphi \rangle = 0$ for all $\varphi$, which implies $\psi = 0$ by the properties of the inner product. □

As a second example, consider the usual characterization of unitary maps in terms of the inner product. A linear map $U : \mathcal{H} \to \mathcal{H}$ is unitary if and only if $\langle U\psi, U\varphi \rangle = \langle \psi, \varphi \rangle$ for each pair of vectors $\psi, \varphi \in \mathcal{H}$. Unitary maps obey a similar property if the inner product is replaced with the skew form.

**Lemma 2.1.2.** If $U : \mathcal{H} \to \mathcal{H}$ is unitary then $\omega(U\psi, U\varphi) = \omega(\psi, \varphi)$ for each pair of vectors $\psi, \varphi \in \mathcal{H}$.

*Proof.* The proof is a direct calculation. We have

$$\omega(U\psi, U\varphi) = \operatorname{Im}\langle U\psi, U\varphi\rangle = \operatorname{Im}\langle \psi, \varphi\rangle = \omega(\psi, \varphi).$$

□

It is crucial to note however that a linear map $A : \mathcal{H} \to \mathcal{H}$ that satisfies $\omega(A\psi, A\varphi) = \omega(\psi, \varphi)$ for each $\psi, \varphi \in \mathcal{H}$ is *not* necessarily unitary. For example, let $\mathcal{H} = \mathbb{C}$ be the complex plane. For a vector $\psi \in \mathbb{C}$ write $\psi = x + i\,y$ where $x, y \in \mathbb{R}$ are the real and imaginary components of $\psi$. Since the Hermetian inner product is $\langle \psi_1, \psi_2\rangle = \psi_1^* \psi_2 = x_1\,x_2 + y_1\,y_2 + i\,(x_1 y_2 - x_2 y_1)$ the skew form $\omega$ is defined by $\omega(\psi_1, \psi_2) = x_1\,y_2 - x_2\,y_1$. For $s$ a positive real constant, the linear map $A\psi = (x/s) + i(sy)$ satisfies

$$\omega(A\psi_1, A\psi_2) = \omega(x_1/s + isy_1, x_2/s + isy_2) = (x_1/s)(s\,y_2) - (x_2/s)(sy_1) = x_1\,y_2 - x_2\,y_1 = \omega(\psi_1, \psi_2).$$

But since $||A\psi||^2 = (x/s)^2 + (sy)^2 \neq x^2 + y^2$, $A$ is not unitary.

Now we will forget about the inner product structure altogether, and consider spaces equipped with a skew-form $\omega$ that mimics the Hilbert space construction axiomatically. Given a real vector space $V$, a skew form $\omega$ on $V$ is **non-degenerate** if it satisfies Lemma 2.1.1. A real vector space $V$ equipped with a non-degenerate skew form is called a **symplectic vector space**. A linear map $A : V \to V$ is **symplectic** if $\omega(Av, Aw) = \omega(v, w)$ for each pair of vectors $v, w \in V$.

Linear classical mechanics takes place in symplectic vector spaces. For example, consider a 1-degree-of-freedom Hamiltonian system on $\mathbb{R}^2 \ni (q, p) = z$ with the quadratic Hamiltonian

$$H(z) = \frac{1}{2} z^T B z, \tag{2.5}$$

where $B$ is any $2 \times 2$ symmetric matrix with real entries. If we introduce the non-degenerate skew form on $\mathbb{R}^2$

$$\omega\,(z_1, z_2) = z_1^T \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} z_2 \equiv z_1^T \mathbb{J} z_2, \tag{2.6}$$

then Hamilton's equations $\dot{z} = (\dot{q}, \dot{p}) = (\partial_p H, -\partial_q H)$ may be written

$$\dot{z} = \mathbb{J}\nabla H = \mathbb{J} B z. \tag{2.7}$$

The deep significance of these purely formal manipulations may be understood by studying the time advance map $T_{\Delta t} = \exp(\Delta t\, \mathbb{J}B)$ for this linear system. Since $\partial_{\Delta t} T_{\Delta t} = T_{\Delta t}\mathbb{J}B = \mathbb{J}B T_{\Delta t}$ and $\omega(\mathbb{J}a, b) = a^T b$, we have

$$\begin{aligned} \frac{d}{d\Delta t}\omega(T_{\Delta t} z_1, T_{\Delta t} z_2) &= \omega(\mathbb{J}B T_{\Delta t} z_1, T_{\Delta t} z_2) + \omega(T_{\Delta t} z_1, \mathbb{J}B T_{\Delta t} z_2) \\ &= (B T_{\Delta t} z_1)^T T_{\Delta t} z_2 - (T_{\Delta t} z_1)^T B T_{\Delta t} z_2 \\ &= (T_{\Delta t} z_1)^T (B T_{\Delta t} z_2) - (T_{\Delta t} z_1)^T (B T_{\Delta t} z_2) \\ &= 0. \end{aligned} \tag{2.8}$$

Therefore $\omega(T_{\Delta t} z_1, T_{\Delta t} z_2) = \omega(T_0 z_1, T_0 z_2) = \omega(z_1, z_2)$. Thus, the time advance map for any linear classical mechanical system is a linear symplectic map! In fact, this can be taken as the definition of a linear Hamiltonian system: a linear dynamical system on a symplectic vector space is Hamiltonian precisely when the time advance map is symplectic.

**Lemma 2.1.3.** Let $\dot{z} = Az$ be a linear dynamical system on $\mathbb{R}^2$ equipped with the non-degenerate skew form $\omega(z_1, z_2) = z_1^T \mathbb{J} z_2$. If $T_{\Delta t} = \exp(\Delta t A)$ is a symplectic map for each $\Delta t$ then there is a symmetric matrix $B$ such that $A = \mathbb{J}B$. In particular, the system $\dot{z} = Az$ is Hamiltonian with Hamiltonian function $H(z) = \frac{1}{2} z^T B z$.

*Proof.* Differentiating $\omega(T_{\Delta t} z_1, T_{\Delta t} z_2) = \omega(z_1, z_2)$ in $\Delta t$ at $\Delta t = 0$ implies $z_1^T A^T \mathbb{J} z_1 + z_1^T \mathbb{J} A z_2$. Since $z_1, z_2$ are arbitrary, we must have

$$A^T \mathbb{J} + \mathbb{J} A = 0. \tag{2.9}$$

Now, since $\mathbb{J}$ is invertible, there must be a matrix $B$ such that $A = \mathbb{J}B$. Using $A^T = B^T\mathbb{J}^T = -B^T\mathbb{J}$, $\mathbb{J}^2 = -1$, and substituting into the above formula then gives

$$\begin{aligned} 0 &= -B^T\mathbb{J}\mathbb{J} + \mathbb{J}\mathbb{J}B \\ &= B^T - B, \end{aligned} \tag{2.10}$$

which is the desired result. □

It is now simple to understand why quantum mechanics is a special case of classical mechanics. According to Lemma 2.1.2, every unitary operator on a Hilbert space preserves the underlying non-degenerate skew-form. Since the time advance map for a quantum system is unitary, quantum time evolution is therefore symplectic. But Lemma 2.1.3 says that linear systems with symplectic time advance maps must be Hamiltonian. Therefore every quantum system is also a Hamiltonian system!

## 2.2 Hamiltonian system

A Hamiltonian dynamical system is

$$\begin{aligned} \dot{\mathbf{p}} &= -\nabla_{\mathbf{q}} H(\mathbf{r}) \\ \dot{\mathbf{q}} &= \nabla_{\mathbf{p}} H(\mathbf{r}) \end{aligned}$$

or equivalently

$$\dot{\mathbf{r}} = S_N \nabla H(\mathbf{r})$$

where $\mathbf{r} = (\mathbf{q}, \mathbf{p})$ is a 2N-dimensional vector of the canonical coordinate, $H$ is the *Hamiltonian*, and

$$S_N = \begin{bmatrix} 0 & I_N \\ -I_N & 0 \end{bmatrix}$$

For more details on Hamiltonian systems, see the introduction in [5]. Among all the properties of Hamiltonian systems, the simplest is

**Proposition 2.2.1.** (Conservation of the total energy) For Hamiltonian systems, the Hamiltonian function $H(p, q)$ is a first integral, i.e., $H(p, q) = const.$

The above property is closely related to many numerical schemes being long-time stable or structure-preserving.

Before we discuss its numerical integrations, we first introduce an important concept of symplectic maps (symplecticity),

**Definition 2.2.1.** A differentiable map from $g : U \in \mathbb{R}^{2N} \to \mathbb{R}^{2N}$ is called **symplectic** if its Jacoiban matrix $g'(\mathbf{p}, \mathbf{q})$ satisfies

$$g'(\mathbf{p}, \mathbf{q})^T \, S_N \, g'(\mathbf{p}, \mathbf{q}) = S_N$$

Now consider the flow map $\psi_\tau$ for the Hamiltonian system of $H(\mathbf{p}, \mathbf{q})$ for given time $\tau$, which is defined by

$$\psi_t(\mathbf{p}_0, \mathbf{q}_0) := (\mathbf{p}(t, \mathbf{p}_0, \mathbf{q}_0), \mathbf{q}(t, \mathbf{p}_0, \mathbf{q}_0))$$

There exists a very famous theorem regarding the flow maps in Hamiltonian systems

**Theorem 2.2.1.** Let $H(\mathbf{p}, \mathbf{q})$ be a twice continuously differentiable function on $U \in \mathbb{R}^{2N}$. Then, for each fixed $\tau$, the flow map $\psi_\tau$ is a symplectic transformation wherever it is defined.

In fact, symplecticity of the flow is a characteristic property for Hamiltonian systems. We call a differential equation $\dot{y} = f(y)$ *locally Hamiltonian*, if for any point, there exists a neighborhood where $f(y) = S_N \nabla H$ for some function $H$. There exists a theorem that relates symplecticity with Hamiltonian systems,

**Theorem 2.2.2.** Let $f : U \in \mathbb{R}^{2N} \to \mathbb{R}^{2N}$ be continuously differentiable. Then, $\dot{y} = f(y)$ is locally Hamiltonian if and only if its flow $\psi_\tau$ is symplectic for all $y \in U$ and for all sufficiently small $\tau$.

## 2.3 Symplectic integrators

In general, a numerical integrator can be viewed as a discrete realization of flow maps with a fixed time step $\Delta t$. Therefore, for Hamiltonian systems, it is advantageous if a time integrator can mimic the symplectic property of Hamiltonian systems. This class of time integrators is referred to as *symplectic integrators*. The formal definition is as follows

**Definition 2.3.1.** A numerical integrator $(\mathbf{p}^{n+1}, \mathbf{q}^{n+1}) = g_h(\mathbf{p}^n, \mathbf{q}^n)$ is called symplectic if it satisfies symplecticity **exactly**, i.e.,

$$g_h'(\mathbf{p}^n, \mathbf{q}^n)^T \, S_N \, g_h'(\mathbf{p}^n, \mathbf{q}^n) = S_N$$

Although such an integrator is generally preferred, there are a few disadvantages which prevent its broad usage in large-scale simulations.

- the integrator is typically semi-implicit or fully implicit
- it is hard to accelerate through a parallel-in-time integrator
- it is generally challenging to apply to continuous PDEs directly (it may be applicable in PIC simulations)

Thus, a ML-based approach may dramatically overcome some of those difficulties.

## 2.4 Unitary matrix

As a parallel interest, we consider the unitary matrix $U$ satisfying $UU^* = I$ and consider possible approaches to directly learn those matrices using ML. The properties of unitary matrix include

$$|\det U| = 1 \tag{2.11}$$

and its eigenspaces are orthogonal. The application of a unitary matrix include the inverse scattering problem and quantum computing.

Consider some alternative representations of the unitary matrix for the purpose of learning unitary matrices. First, the matrix is unitary if and only if there exists a Hermitian matrix $H$ such that

$$U = \exp(iH). \tag{2.12}$$

Similarly, the matrix is unitary if and only if

$$U = (I + S)(I - S)^{-1} \tag{2.13}$$

where $S$ is a skew-Hermitian matrix. Another representation is from numerical linear algebra, in which the unitary matrix is commonly expressed as a product of Householder transformations

$$H = I - \sigma \mathbf{v}\mathbf{v}^*.$$

The requirement of $H$ being unitary leads to a condition

$$|\sigma|^2 \|\mathbf{v}\|_2^2 = 2\mathrm{Re}(\sigma)$$

See https://www.netlib.org/lapack/lug/node128.html. In LAPACK, the vector is assumed to satisfy $v_1 = 1$. It is also assumed that in the real arithmetic $1 \le \sigma \le 2$ while in the complex arithmetic $1 \le \mathrm{Re}(\sigma) \le 2$ and $|\sigma - 1| \le 1$, expect when $\sigma = 0$ for the special case of $H = I$. This sudden jump of definition is not good in ML. The good news is the lower bound of $\sigma$ seems simply to be a choice they made in LAPACK. The necessary bound is only $0 \le \sigma \le 2$ for real arithmetic or $|\sigma - 1| \le 1$ for complex arithmetic, which is derived from $v_1 = 1$. Another possible representation is from quantum computing, in which a unitary matrix is commonly decomposed into two-level unitary matrices. However, it requires $\binom{n}{2} = n(n-1)/2$ small matrices in decomposition, which is too deep for the ML training.

In summary, there are several obvious ways to learn a unitary matrix

- Use ML to approximate a Hermitian matrix $H$ and define a unitary matrix by $U = \exp(iH)$ through a numerical matrix exponential `expm`. The pros are using ML to approximate $H$ is a simple task and the computational cost is low. The cons are there is nothing to guarantee the matrix from `expm` is a unitary matrix and `expm` needs some extra properties of $H$ to guarantee convergence.
- Use ML to approximate a skew-Hermitian matrix $S$ and define a unitary matrix through $U = (I + S)(I - S)^{-1}$ The pros are the matrix is always unitary and using ML to approximate $S$ is a simple task. The cons are one matrix inversion is needed and there is an extra requirement of $I - S$ being nonsingular.
- Use ML to approximate unitary matrices directly through

  $$U = H_1 H_2 ... H_k$$

  where each $H_i$ is defined

  $$H_i = I - \sigma_i \mathbf{v}_i \mathbf{v}_i^*$$

  with $v_1 = 1$ and $|\sigma - 1| \leq 1$. Here ML is used to approximate vectors $\mathbf{v}_i$ and $\sigma_i$. This seems a well defined ML problem but it may need many layers to converge (at most $n$).
- The disadvantage of the above representation is it requires to solve some constraints related to $\sigma$. The introduction of $\sigma$ is really for the QR factorization and therefore it may not be necessary in the ML learning. A better representation of Householder matrices may be

  $$H_i = I - 2\mathbf{v}_i \mathbf{v}_i^* / (\mathbf{v}_i^* \mathbf{v}_i)$$

  The disadvantage of the above approach is each $H_i$ comes with a parameter of $2n$ and in principle, it needs $n$ transformations which gives a total of $2n^2$ parameters. However the dimension of the unitary matrix group is only $n^2$, so it is not optimal.

Another approach is commonly used in quantum computing, see the description in [6]. An arbitrary unitary transformation $U$ can be composed from elementary unitary transformations in two-dimensional subspaces. Denote a two-level matrix as $E^{(i,j)}(\phi, \psi, \chi)$ and its nonzero items are

$$E_{kk}^{(i,j)} = 1 \qquad k = 1 \cdots N, \qquad k \neq i, j \tag{2.14}$$

$$E_{ii}^{(i,j)} = \cos\phi \exp(i\psi) \tag{2.15}$$

$$E_{ij}^{(i,j)} = \sin\phi \exp(i\psi) \tag{2.16}$$

$$E_{ji}^{(i,j)} = -\sin\phi \exp(-i\chi) \tag{2.17}$$

$$E_{jj}^{(i,j)} = \cos\phi \exp(-i\chi) \tag{2.18}$$

The above element unitary transformation can be used to construct the following composite rotations

$$E_1 = E^{(1,2)}(\phi_{12}, \psi_{12}, \chi_{12}) \tag{2.19}$$

$$E_2 = E^{(2,3)}(\phi_{23}, \psi_{23}, 0)\, E^{(1,3)}(\phi_{13}, \psi_{13}, \chi_{13}) \tag{2.20}$$

$$E_3 = E^{(3,4)}(\phi_{34}, \psi_{34}, 0)\, E^{(2,4)}(\phi_{24}, \psi_{24}, 0)\, E^{(1,4)}(\phi_{14}, \psi_{14}, \chi_{14}) \tag{2.21}$$

$$\cdots \tag{2.22}$$

$$E_{N-1} = E^{(N-1,N)}(\phi_{N-1,N}, \psi_{N-1,N}, 0) E^{(N-2,N)}(\phi_{N-2,N}, \psi_{N-2,N}, 0) \cdots E^{(1,N)}(\phi_{1,N}, \psi_{1,N}, \chi_{1,N}) \tag{2.23}$$

Then the unitary matrix is defined as

$$U = \exp(i\alpha) E_1 E_2 \cdots E_{N-1}$$

This decomposition consists of $N^2$ parameters. There are two advantages of this decompositions: (1) it needs fewer parameters (it is optimal) (2) this decomposition is used to generate random unitary matrix and thus it is easy to initialize a random matrix using this formula. The disadvantage is $E_i$ is not a dense matrix in general.

## 2.5 Symmetric unitary matrix

Now consider the case of symmetric unitary matrix. We collect some results related those matrices.

**Theorem 2.5.1.** The matrix $U$ is a symmetric unitary matrix if and only if there exists a unitary matrix $W$ such that

$$U = WW^T$$

Note the above factorization is not unique. Let Q be a real orthogonal matrix such that $OO^T = I$, then

$$U = WW^T = (WO)(WO)^T.$$

In fact, the set of symmetric unitary matrices $\mathcal{O}$ is isomorphic to the left coset space of the orthnomal matrix set $O(N)$ in the unitary matrix group $U(N)$,

$$\mathcal{O} \cong U(N)/O(N).$$

# Chapter 3

# Symplectic networks

## 3.1 Previous work on HenonNet

In the previous work, we have designed and developed a novel network, HenonNet, a special kind of symplectic neural networks to study Hamiltonian systems. There are two key ideas in this work.

The first idea is an approximation theorem of HenonNet for arbitrary symplectic diffeomorphism, which is proven in [7],

**Theorem 3.1.1.** (Symplectic universal approximation) Given a $C^r$ symplectic diffeomorphism $\mathcal{F} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n \times \mathbb{R}^n$, a compact region $U \subset \mathbb{R}^n \times \mathbb{R}^n$, and $\epsilon > 0$, there exists

- a smooth function $V : \mathbb{R}^n \to \mathbb{R}$
- a vector $\eta \in \mathbb{R}^n$
- a positive integer $N$

such that $|H[V, \eta]^{4N} - \mathcal{F}|_{C^r(U)} < \epsilon$.

The work [7] does not consider machine learning, instead it constructs a special kind of Henon maps (its composition is denoted as $H[V, \eta]^{4N}$ here) which can be used to approximate any symplectic diffeomorphism arbitrary close. However, in the theorem, the potential function $V$, vector $\eta$ and integer $N$ are not known a priori. This is actually ideal for ML, since we can learn $V$ and $\eta$ using ML while $N$ can be used as a hyperparameter. Therefore, we construct a neural network consistent with the special maps defined in the approximation theorem. The significant advantage of this network is it is symplectic by design, which can be interpreted as *we embed physics into the network*. This is different from many existing works in applying ML to math/physics problems where physics is only embedded through a loss function. We found embedding physics into the network architecture can dramatically improve the approximation power of the network.

The second key idea is we apply the network to learning a Poincare map. Since a strong approximation theorem is established for the network, we conducted a very aggressive task, i.e., the network is used to learn a flow map with a huge time step (note it typically needs O(1000) time steps if a conventional RK integrator is used to evolve the same large time step). Thanks to the large time step approximation, after training, the network can generate a Poincare plot an order of magnitude faster than any conventional time integrator. Here the comparision is measured on CPUs for a fair comparison, and the improvement is even more significant if it is on GPUs.

Compared to the state-of-art works in which a regression or ML method is used to learn flow maps, our work is first to learn a network of very large time step (10-50 times larger than other methods) and it is first to learn both chaotic and regular regions.

## 3.2 Parameterized HenonNet

Recently, with some works, we found that the approximation theorem in [7] can be extended to parameterized cases, i.e.,

**Theorem 3.2.1.** (Parameterized symplectic universal approximation) Given a $C^r$ symplectic diffeomorphism $\mathcal{F}_\mu : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n \times \mathbb{R}^n$ with smooth $\mu$-dependence, a compact region $U \subset \mathbb{R}^n \times \mathbb{R}^n$, and $\epsilon > 0$, there exists

- a smooth function $V_\mu : \mathbb{R}^n \to \mathbb{R}$
- a vector $\eta_\mu \in \mathbb{R}^n$
- a positive integer $N$

such that $|H[V_\mu, \eta_\mu]^{4N} - \mathcal{F}_\mu|_{C^r(U)} < \epsilon$ uniformly in $\mu$.

We propose two types of NN architecture to generalize:

- **parametrized HénonNet**: replace $V_W(y)$ in the previous network structure with $V_W(y, \mu)$. The approximation can be proved by our theorem. A simple version has been implemented and tested.
- **meta HénonNet**: feed-forward NN that maps parameters $\mu$ into the space of HénonNets. This is a more general version than the parametrized HénonNet. It hopefully can improve its approximation power.

We further propose the following ideas based on the above theorem.

### 3.2.1 Idea 1: Learn Poincare/flow maps for a family of B fields

One criticism of the previous work is one has to learn one field each time, which is not very efficient if one wants to study a family of B fields such as $B_\lambda = B_0 + g(\lambda)$.

The new approximation theorem makes it possible to learn such a family, for instance, when $g(\lambda)$ is a small perturbation. In practice, this is mostly useful since physicists like to study some perturbation with respect to a known field. A more aggressive approach is to learn the discrete representation of all the possible B field. For instance, using a Fourier expansion in $\phi$ of a cylindrical coordinate, we may consider the family of $B$ as

$$B = B(r, z) + \sum_{n=1}^{N} \left(a_n \cos(n\phi) + b_n \sin(n\phi)\right).$$

In this case, the parameter would be all the possible $\{(a_n, b_n)\}_N$. Note the resulting learned map can be viewed as an operator from the parameter space $\{(a_n, b_n)\}_N$ towards its flow maps.

**Potential risk** The second task is certainly very aggressive. We propose to learn an uncountable infinite number of fields/Hamiltonian systems at the same time. The training could be very challenging, or it is likely impossible to capture a very localized structure among the parameter space.

### 3.2.2 Idea 2: ROM with HenonNet

One significant challenge in ROM is it is generally hard to achieve any sort of numerical stability. A simple example would be a shallow-water or wave equation. It is known that both equations have the underlying Hamiltonian structure, and thus the high-fidelity model guarantees long-time stability due to the energy conservation. This however is challenging to achieve in its reduced order model. Even though the training data is stable, the reduced model will quickly diverge when one evolve moderately many time steps. The underlying reason is the reduced order model loses a symplectic structure. Recently, Jan Hesthave et al. proposed a symplectic ROM approach by carefully construct reduced bases using the conventional ROM

approach in [8], which received a lot of credits in the ROM community. The low-rank structure of certain parameterized Hamiltonian has been also shown in [8].

Inspired by two works: (a) Nathan Kutz's Koopman operator approximation [9] and (b) Jan Hesthaven's work of symplectic ROM [8], we can achieve a **symplectic ROM** by using HénonNet. See Figure 3.1 for the architecture. Note the symplectic property and ROM can be easily achieved through the ML approach.

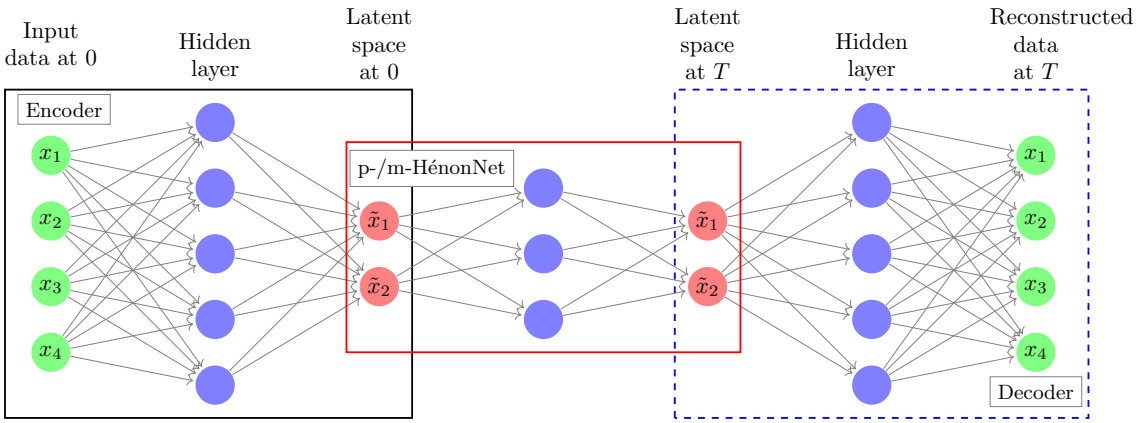


Figure 3.1: Symplectic ROM with HenonNet

**Potential risk** Although two key parts of symplectic and ROM can be easily achieved, it is not clear how accurate the network can learn the high-fidelity model. This however can be used as a criticism to most of ROM method if an approximation theorem with ROM is not established.

### 3.2.3 Idea 3: using HénonNet to study bifurcation

A very recent work of [10] shows a symplectic integrator preserves many bifurcation structure of Hamiltonian boundary value problem. This inspires us to use the HenonNet to study bifurcation structures in the system.

For instance, consider a bifurcation Hamiltonian problem with respect to $C$

$$H(p, q; C) = \frac{1}{2}p^2 + C\exp(q),$$

with a Dirichlet boundary condition $q(0) = q(1) = q_0$ and assume one wants to understand bifurcation with respect to $C$. In conventional approaches, it would first reformulate into an algebraic problem

$$F(p, q; C) = 0.$$

Then a method of numerical continuation is used to solve the problem in the space of $(p, C)$. So this is generally a problem of size 2D+1D+time.

If an approximation to the flow map $\mathcal{F}(p, q; C)$ from 0 to 1 is trained through HenonNet, we can directly solved the bifurcation digram using

$$\mathcal{F}_2(p, q(0); C) = q(1).$$

in the space of $(p, C)$. So this effectively solves the problem in a 2D space, which can be potentially significantly faster than the conventional approach. The study in [10] indicates symplectic NN can expose qualitatively accurate bifurcation results.

**Potential risk** It is going to be challenging for the symplectic NN to expose quantitatively accurate results.

### 3.2.4 Idea 4: use HénonNet as a discrepancy model

Assume we know a good analytical approximation to certain Hamiltonian flow maps, denoted as, $\mathcal{F}_a(x, y, \eta_0)$. HénonNet $\mathcal{F}(x, y, \eta)$ can be used as a discrepancy model, efficiently correcting the known model.

For instance, the network may be designed as

$$F_{true}(x, y, \eta) \approx \mathcal{F}(x, y, \eta) \circ \mathcal{F}_a(x, y, \eta_0)$$

Due to the structure of HenonNet, the analytical formulation of $\mathcal{F}_a$ will be kept in the network. Therefore, the discrepancy model is likely going to be efficient thanks to the ResNet structure of HenonNet.

### 3.2.5 Idea 5: using PINN approach to learn flow maps

Our previous work uses a supervised learning with a collection of training data pair $\{(p, q)_i, (\hat{p}, \hat{q})_i\}$. This can be avoided by using the physics-informed neural network (PINN) approach.

The key idea is to note the Hamiltonian flow map $\mathcal{F}(x, y, t)$ is smooth/continuous with respect to time. Time can be simply treated as a parameter of flow maps. Using a HénonNet to approximate the "parameterized" flow map, we can use a PINN approach based on the loss

$$Loss := MSE(\,\mathcal{F}(\cdot, 0),\, \mathbf{I}\,) + MSE(\,\partial_t \mathcal{F}(x, y, t),\, S_N \nabla H)$$

The two terms in the loss correspond to its initial condition and ODEs, respectively.

## 3.3 HenonNet generalization

We also propose a few approaches to improving HenonNet training or extend the applications of HenonNet.

### 3.3.1 Non-canonical systems

In practical applications, the Hamiltonian system is typically described in a non-canonical coordinate without knowing the transformation to the canonical coordinate. For such a system, it is very challenging to deploy symplectic integrators directly.

The idea is to use HénonNet and learned Darboux transformation to integrate non-canonical symplectic integrators. It uses a network structure similar with the autoencoder idea, except the "autoencoder" is used to represent a coordinate transformation from a non-canonical space to an unknown canonical space. This Darboux transformation is learned by invertible neural network from normalized flow [11].

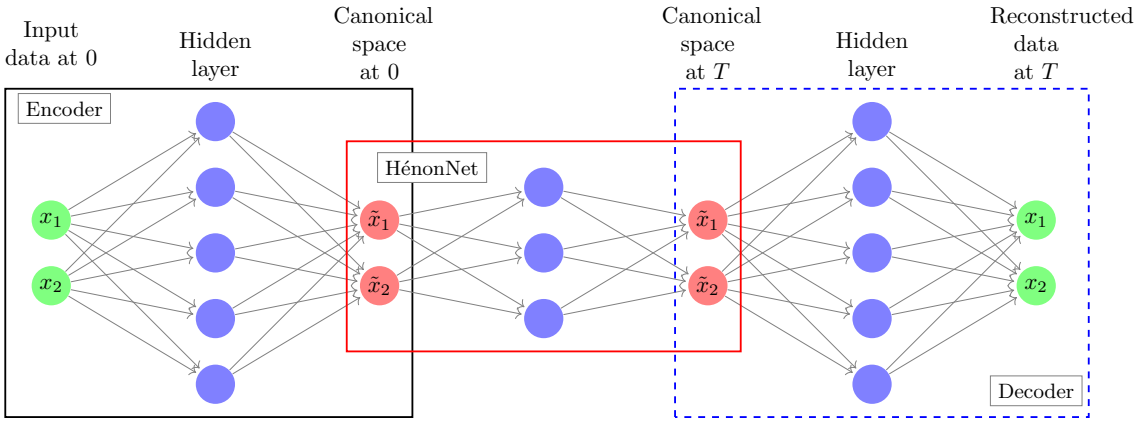


Figure 3.2: Symplectic network for non-canonical systems

The outcome would be a non-canonical symplectic integrator/flow map. Note the dimension of the canonical space is not a hyperparameter in this case. Guiding center dynamics would be immediate application. We have some preliminary results using this idea.

### 3.3.2 PDEs with underlying Hamiltonian structure

The HenonNet can be applied to certain PDEs with underlying Hamiltonian structure. PDEs are infinite dimensional Hamiltonian systems, and thus it is natural to first reduce the dimensions to finite dimensions and then apply HenonNet.

Consider a simple model problem

$$\partial_{tt} u = \Delta u$$

with a periodic boundary condition. Apply a spectral method

$$\partial_{tt} \hat{u}_k = -k^2 u_k$$

The system has a discrete Hamiltonian defined as

$$H := 1/2 \sum_k (\dot{\hat{u}}_k^2 - k^2 \hat{u}_k^2)$$

Then we use HenonNet to learn the dynamics in the Fourier space. The above idea can be applied to other discretization as long as there is a discrete Hamiltonian in the discretization. Vlasov systems however need more thoughts.

### 3.3.3 Splitting method

Another extension of the HenonNet is through the splitting method. For a general introduction of splitting methods, see [12].

For instance, consider a dissipative Hamiltonian system satisfies

$$\frac{\partial H}{dt} = -cp^T p$$

We could use a simple splitting scheme with a flow map for time t

$$e^{-ct} \mathcal{F}(x, y)$$

The above network is *conformal symplectic*, i.e., it satisfies

$$\Omega(t) = e^{-ct} \Omega(0)$$

We can also extend it to a general splitting method

$$e^{-ct_N} HL^{(N)} \circ \cdots \circ e^{-ct_1} HL^{(1)}.$$

where $\sum_N t_N = t$ (coefficient can be also optimzied). The idea is also applicable to other splitting systems such as fast slow splitting.

The most significant limitation of HenonNet is it is only applicable in a dissipative free system. The splitting approach is an attempt to generalize the HenonNet to a dissipative system. The above dissipative system is still too limited and it is still unclear how to extend the network to a more general dissipative system. However, the idea in the generalization through splitting methods is interesting.

### 3.3.4 Parallel-in-layer training

The network can be train using parallel-in-layer (model distribution). Thanks to the composition property of symplectic maps, we can train each Henon layer to approximate certain portion of the time-advanced map $[t_i, t_{i+1}]$. We can even follow a half-V cycle in the multigrid as an approach to improve the final time-advance map. We have used a simpler version in the previous Poincare map work, and an improvement in training was observed.

### 3.3.5 Learn periodic problems

We have some ideas to build periodic architecture (time, space or parameter) into the HenonNet. It requires a small modification of the network by adding an extra layer of sin and cos into the potential function. The potential application is time periodic non-autonomous systems or spatial periodic problems such as the standard maps.

### 3.3.6 Quantum HénonNets

A HénonNet comprises a composition of a (user-specified) number of Hénon layers. Each Hénon layer is an explicit symplectic mapping $HL = \mathcal{H} \circ \mathcal{H} \circ \mathcal{H} \circ \mathcal{H}$, where the Hénon map $\mathcal{H}(\boldsymbol{q}, \boldsymbol{p}) = (\overline{\boldsymbol{q}}, \overline{\boldsymbol{p}})$ is given by

$$\overline{\boldsymbol{q}} = \boldsymbol{p} + \boldsymbol{\eta} \tag{3.1}$$

$$\overline{\boldsymbol{p}} = -\boldsymbol{q} + \nabla V(\boldsymbol{p}), \tag{3.2}$$

for some scalar potential $V(\boldsymbol{q})$ and bias vector $\boldsymbol{\eta}$.

As explained earlier in Section 2.1, any unitary matrix, and in particular any $S$-matrix is automatically a (linear) symplectic map. Since HénonNets can approximate any symplectic mapping, they may therefore be used to learn any $S$-matrix, while exactly preserving the Hamiltonian structure inherent to quantum mechanics. However, HénonNets do not commute with quantum phase rotations $\psi \mapsto e^{i\phi}\psi$, and therefore do not manifestly preserve unitarity. (Unitarity of quantum dynamics may be understood as the conservation law associated with phase invariance of the quantum Hamiltonian $\langle \psi, \hat{H}\psi \rangle$.) It is therefore natural to wonder if HénonNets can be modified in some way to preserve both unitarity and Hamiltonian structure. In fact, it is enough to seek network architectures that preserve unitarity only; we saw in Section 2.1 that unitary matrices are automatically symplectic.

We may develop a manifestly-unitary network architecture by constructing a quantum analogue of the HénonNet architecture as follows. Consider first the classical physics interpretation of a single Hénon map $\mathcal{H}$. For simplicity, set $\boldsymbol{\eta} = 0$. In order to obtain $(\overline{\boldsymbol{q}}, \overline{\boldsymbol{p}}) = \mathcal{H}(\boldsymbol{q}, \boldsymbol{p})$ from $(\boldsymbol{q}, \boldsymbol{p})$, first, the position and momentum are rotated into one another by 90 degrees using the mapping $f(\boldsymbol{q}, \boldsymbol{p}) = (\boldsymbol{p}, -\boldsymbol{q})$. Next one flows along the Hamiltonian dynamics with Hamilton function $H(\boldsymbol{q}, \boldsymbol{p}) = -V(\boldsymbol{q})$ for 1 second, i.e. one applies the mapping $d(\boldsymbol{q}, \boldsymbol{p}) = (\boldsymbol{q}, \boldsymbol{p} + \nabla V(\boldsymbol{q}))$. The net result is

$$\mathcal{H}(\boldsymbol{q}, \boldsymbol{p}) = d(f(\boldsymbol{q}, \boldsymbol{p})) = d(\boldsymbol{p}, -\boldsymbol{q}) = (\boldsymbol{p}, -\boldsymbol{q} + \nabla V(\boldsymbol{p})), \tag{3.3}$$

which recovers the above formula for $\mathcal{H}$ with $\boldsymbol{\eta} = 0$, as claimed.

Now consider the quantum analogue of evaluating $\mathcal{H}$.

- The quantum mechanical method for rotating position into momentum is given by the Fourier transform $\mathcal{F}$, which is a unitary operator on Hilbert space. If the Hilbert space is finite dimensional, then $\mathcal{F}$ must be given by the discrete Fourier transform, which is still unitary. Note that both the mapping $f$ and $\mathcal{F}$ give the identity after being applied four times, i.e. $f \circ f \circ f \circ f = \mathrm{id}$ and $\mathcal{F}^4 = \mathbb{I}$.
- The quantum analogue of the Hamiltonian $-V(\boldsymbol{q})$ is given simply by replacing position with the position operator to obtain $-V(\hat{\boldsymbol{q}})$. In units where $\hbar = 1$, the one-second propagator for this Hamiltonian is just the diagonal operator (in position space) $D = \exp(iV(\hat{\boldsymbol{q}}))$.

It follows that the quantum analogue of the Hénon map $\mathcal{H} = d \circ f$ is given by

$$\mathcal{H}_Q = D\mathcal{F}, \tag{3.4}$$

where $\mathcal{F}$ is the (discrete) Fourier transform and $D$ is a diagonal unitary map. We may therefore define a quantum Hénon layer according to

$$HL_Q = \mathcal{H}_Q\mathcal{H}_Q\mathcal{H}_Q\mathcal{H}_Q = D\mathcal{F}D\mathcal{F}D\mathcal{F}D\mathcal{F}, \tag{3.5}$$

where the diagonal unitary matrix $D$ is a trainable parameter. A Quantum HénonNet, or **QHénonNet**, is then defined as a composition of Quantum Hénon layers $HL_Q^{(k)}$. The weights of a QHénonNet may be identified with the networks collection of diagonal unitary matrices $D^{(k)}$.

Remarkably, Ref. [**?**] proves that any unitary matrix may be expressed as a finite composition of Quantum Hénon layers. Therefore QHénonNets are unitary universal approximators.

We may incorporate parameters in the QHénonNet framework by mimicking parameterized HénonNets. In particular, in the diagonal unitary matrix $D = \mathrm{diag}(e^{i\phi_1}, \ldots, e^{i\phi_n})$ we replace the vector of phases $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_n)$ with a feed-forward neural network $\boldsymbol{\Phi}_{\boldsymbol{W}}(\boldsymbol{\mu}) = (\phi_1(\boldsymbol{\mu}; \boldsymbol{W}), \ldots, \phi_n(\boldsymbol{\mu}; \boldsymbol{W}))$. Here $\boldsymbol{\mu}$ is a vector of parameters and $\boldsymbol{W}$ denotes the weights of the feed-forward network $\boldsymbol{\Phi}_{\boldsymbol{W}}$. In this manner, we obtain a neural network architecture that is capable of representing the map from a scattering model's parameter vector $\boldsymbol{\mu}$ to the associated $S$-matrix $S(\boldsymbol{\mu})$, while preserving unitarity exactly.

### 3.3.7 Unitary reduced-order models with parameters

Large unitary matrices $U : \mathcal{H} \to \mathcal{H}$ may be challenging to learn. Therefore it is natural to wonder if the most important features of $U$ can be encoded in a smaller unitary matrix $\mathfrak{u}_0 : \mathfrak{h}_0 \to \mathfrak{h}_0$. Unitary reduced-order models (ROMs) provide a natural framework for studying this question.

To understand unitary ROMs, it is useful to introduce the notion of a *reducible unitary operator*. A unitary operator $U : \mathcal{H} \to \mathcal{H}$ on a Hilbert space $\mathcal{H}$ is **reducible** if there is a linear subspace $\mathfrak{h} \subset \mathcal{H}$ such that $U(\mathfrak{h}) = \mathfrak{h}$ and $U \mid \mathfrak{h}^\perp = \mathrm{id}_{\mathfrak{h}^\perp}$. Thus, a reducible unitary operator $U$ admits an invariant subspace whose orthogonal complement comprises fixed points for $U$. Given a reducible unitary operator $U$, the restriction of $U$ to $\mathfrak{h}$ defines a linear operator $\mathfrak{u} = U \mid \mathfrak{h} : \mathfrak{h} \to \mathfrak{h}$ that is unitary with respect to the induced inner product $\langle \cdot, \cdot \rangle_{\mathfrak{h}}$ on $\mathfrak{h}$. We say $\mathfrak{u}$ is the **reduction** of the reducible unitary operator $U$. As the following proposition shows, the pair $(\mathfrak{h}, \mathfrak{u})$ completely determines $U$.

**Proposition 3.3.1.** There is a bijective mapping $R$ that sends reducible unitary operators $U : \mathcal{H} \to \mathcal{H}$ into the space of pairs $(\mathfrak{h}, \mathfrak{u})$, where $\mathfrak{h} \subset \mathcal{H}$ is a linear subspace and $\mathfrak{u} : \mathfrak{h} \to \mathfrak{h}$ is unitary.

*Proof.* The comments above already establish the existence of a mapping $R$ from reducible unitary operators $U$ to pairs $(\mathfrak{h}, \mathfrak{u})$. We will establish this map's injectivity and surjectivity.

*Injectivity:* Suppose the unitary operators $U, U'$ determine the same pair $(\mathfrak{h}, \mathfrak{u})$, i.e. $R(U) = R(U')$. Since $\mathfrak{h}^\perp$ comprises fixed points for both $U$ and $U'$, $U\psi = U'\psi$ whenever $\psi \in \mathfrak{h}^\perp$. Since $U$ and $U'$ each restrict to $\mathfrak{u}$ on $\mathfrak{h}$, we also have $U\psi = U'\psi$ whenever $\psi \in \mathfrak{h}$. Since $\mathcal{H} = \mathfrak{h} \oplus \mathfrak{h}^\perp$, we conclude $U\psi = U'\psi$ for all $\psi \in \mathcal{H}$. In other words $U = U'$.

*Surjectivity:* Fix $(\mathfrak{h}, \mathfrak{u})$, where $\mathfrak{h} \subset \mathcal{H}$ is a linear subspace and $\mathfrak{u} : \mathfrak{h} \to \mathfrak{h}$ is unitary. We will construct a unitary $U : \mathcal{H} \to \mathcal{H}$ such that $R(U) = (\mathfrak{h}, \mathfrak{u})$. Let $\iota : \mathfrak{h} \to \mathcal{H}$ be the inclusion map. We may lift $\mathfrak{u} : \mathfrak{h} \to \mathfrak{h}$ to an operator $\widetilde{\mathfrak{u}} : \mathcal{H} \to \mathcal{H}$ using compositions of $\iota$ and its adjoint $\iota^\dagger$ according to $\widetilde{u} = \iota\, \mathfrak{u}\, \iota^\dagger$. We claim that

$$U = \widetilde{\mathfrak{u}} + \Pi_{\mathfrak{h}^\perp}, \tag{3.6}$$

where $\Pi_{\mathfrak{h}^\perp}$ is the orthogonal projection onto $\mathfrak{h}^\perp$, is unitary and satisfies $R(U) = (\mathfrak{h}, \mathfrak{u})$. To confirm unitarity, we fix $\psi \in \mathcal{H}$, set $(\psi_\parallel, \psi_\perp) = (\Pi_{\mathfrak{h}}\psi, \Pi_{\mathfrak{h}^\perp}\psi)$, and compute directly:

$$\langle U\psi, U\psi \rangle = \langle \widetilde{\mathfrak{u}}\psi, \widetilde{\mathfrak{u}}\psi \rangle + \langle \Pi_{\mathfrak{h}^\perp}\psi, \Pi_{\mathfrak{h}^\perp}\psi \rangle = \langle \psi, \iota\, \mathfrak{u}^\dagger\, \iota^\dagger\, \iota\, \mathfrak{u}\, \iota^\dagger \psi \rangle + \langle \psi_\perp, \psi_\perp \rangle = \langle \psi, \Pi_{\mathfrak{h}}\psi \rangle + \langle \psi_\perp, \psi_\perp \rangle = \langle \psi, \psi \rangle. \tag{3.7}$$

Here we have used $\iota^\dagger\, \iota = \mathrm{id}_{\mathfrak{h}}$, $\iota\, \iota^\dagger = \Pi_{\mathfrak{h}}$, and the unitarity of $\mathfrak{u}$. To confirm $R(U) = (\mathfrak{h}, \mathfrak{u})$, we note that $U$ restricts to $\mathfrak{u}$ on $\mathfrak{h}$ because $\Pi_{\mathfrak{h}^\perp}\psi = 0$ and $\iota^\dagger\psi = \psi$ whenever $\psi \in \mathfrak{h}$. □

A **unitary ROM** for a unitary operator $\mathcal{U} : \mathcal{H} \to \mathcal{H}$ is a reducible unitary operator $U$ that is close to $\mathcal{U}$ in terms of some conventional operator norm such as $||\mathcal{U} - U||^2 = \sup_{||\psi||=1} ||(\mathcal{U} - U)\psi||^2$ and whose corresponding invariant subspace $\mathfrak{h}$ is low-dimensional. Note that a unitary ROM for $\mathcal{U}$ is not the same as a low-rank approximation, since unitary matrices are always invertible, and therefore have maximal rank. Nevertheless, unitary ROM and low-rank approximation share the common goal of compressing the information contained in a large matrix.

One way to compute a unitary ROM $U$ for a unitary matrix $\mathcal{U}$ on an $N$-dimensional Hilbert space is to minimize $||\mathcal{U} - U||^2$ subject to the constraint $\dim \mathfrak{h} = n$, where $n$ is some integer less than $N$. This is

an optimization problem on the space of pairs $(\mathfrak{h}, \mathfrak{u})$ with $\dim \mathfrak{h} = n$ and $\mathfrak{u} : \mathfrak{h} \to \mathfrak{h}$ unitary. The following theorem offers an equivalent formulation of the optimization problem in terms of rectangular matrices with orthogonal columns.

**Definition 3.3.1.** Let $(\mathfrak{h}_0, \langle\cdot,\cdot\rangle_0)$ and $(\mathcal{H}, \langle\cdot,\cdot\rangle)$ be Hilbert spaces with dimensions $n$ and $N$, respectively. Assume $n \leq N$. A **linear isometric embedding** is a linear map $w : \mathfrak{h}_0 \to \mathcal{H}$ that preserves lengths and angles. In other words, for each $\psi_0 \in \mathfrak{h}_0$ we have $\langle w\psi_0, w\psi_0\rangle = \langle \psi_0, \psi_0\rangle_0$.

**Remark 3.3.1.** If $\mathfrak{h}_0 = \mathbb{C}^n$ and $\mathcal{H} = \mathbb{C}^N$ then a linear isometric embedding $w$ is merely a rectangular $N \times n$ matrix with complex entries whose columns are orthogonal, linearly-independent unit vectors.

**Theorem 3.3.1.** Let $\mathcal{U}$ be a unitary matrix on a Hilbert space $\mathcal{H}$ of dimension $N$. Let $n \leq N$ be an integer and fix a second Hilbert space $\mathfrak{h}_0$ of dimension $n$. The following are equivalent.

(1) The unitary matrix $U$ minimizes $||\mathcal{U}-U||^2$ among the class of reducible unitary matrices with $\dim \mathfrak{h} = n$.

(2) The unitary matrix $U$ minimizes $||\mathcal{U} - U||^2$ among the class of reducible unitary matrices of the form

$$U = w\,\mathfrak{u}_0\,w^\dagger + (\mathrm{id}_{\mathcal{H}} - w\,w^\dagger), \tag{3.8}$$

where $\mathfrak{u}_0 : \mathfrak{h}_0 \to \mathfrak{h}_0$ is unitary and $w : \mathfrak{h}_0 \to \mathcal{H}$ is a linear isometric embedding.

*Proof.* We must show that any reducible unitary matrix $U$ with $\dim \mathfrak{h} = n$ is equal to $w\,\mathfrak{u}_0\,w^\dagger + \Pi_{\mathrm{im}\,w^\perp}$ for some linear isometric embedding $w : \mathfrak{h}_0 \to \mathcal{H}$ and some unitary matrix $\mathfrak{u}_0 : \mathfrak{h}_0 \to \mathfrak{h}_0$. First note that since $\dim \mathfrak{h} = \dim \mathfrak{h}_0$ there exists a (non-unique) linear isometric embedding $w$ with $\mathrm{im}\, w = \mathfrak{h}$. So we may define a linear map $\mathfrak{u}_0 : \mathfrak{h}_0 \to \mathfrak{h}_0$ using the formula $\mathfrak{u}_0 = w^\dagger\,U\,w$. This mapping is unitary since

$$\langle \mathfrak{u}_0\psi_0, \mathfrak{u}_0\psi_0\rangle_0 = \langle \psi_0, w^\dagger\,U^\dagger\,w\,w^\dagger\,U\,w\psi_0\rangle_0 = \langle \psi_0, w^\dagger\,U^\dagger\,\Pi_{\mathfrak{h}}\,U\,w\psi_0\rangle_0 = \langle \psi_0 w^\dagger\,w\psi_0\rangle_0 = \langle \psi_0, \psi_0\rangle_0, \tag{3.9}$$

where we have used $U\,w\psi_0 \in \mathfrak{h}$, $w\,w^\dagger = \Pi_{\mathfrak{h}}$, and $w^\dagger\,w = \mathrm{id}_{\mathfrak{h}_0}$. Moreover, we have the identity

$$w\,\mathfrak{u}_0\,w^\dagger = w\,w^\dagger\,U\,w\,w^\dagger = \Pi_{\mathfrak{h}}\,U\,\Pi_{\mathfrak{h}} = U\,\Pi_{\mathfrak{h}}. \tag{3.10}$$

Therefore

$$U\psi = U\Pi_{\mathfrak{h}}\psi + U\Pi_{\mathfrak{h}^\perp}\psi = w\,\mathfrak{u}_0\,w^\dagger\,\psi + \Pi_{\mathfrak{h}^\perp}\psi. \tag{3.11}$$

The result now follows from

$$\Pi_{\mathfrak{h}^\perp} = \mathrm{id}_{\mathcal{H}} - \Pi_{\mathfrak{h}} = \mathrm{id}_{\mathcal{H}} - w\,w^\dagger. \tag{3.12}$$

□

# Chapter 4

# Application: atomic physics (Mark and Nathan)

## 4.1 Relations Between Scattering Matrices and Cross Sections

In the Optical Theorem the total cross sections $\sigma^{\mathrm{Tot}}(E)$ at the incident electron energy $E$ is related to the elastic scattering amplitude $f^{\mathrm{Elast}}(E,\theta)$ (in momentum space representation pg 105 Landau) via:

$$\sigma^{\mathrm{Tot}}(E) \;=\; \frac{4\pi}{k}\mathrm{Im}\left[f^{\mathrm{Elast}}(E,\theta=0)\right], \tag{4.1}$$

where $k^2/2 = E$. The scattering amplitude is related to the complex $T$-matrix $T(E)$ and the cross section $\sigma_{l,u}(E)$ with the following:

$$f(E,\theta) \;\propto\; T(E) \tag{4.2}$$
$$f_L(E,\theta) \;\propto\; T_L(E) \tag{4.3}$$
$$\sigma_{l,u}(E) \;\propto\; |f_{l,u}(E,\theta)|^2 \;\propto\; |T_{l,u}(E)|^2. \tag{4.4}$$

Here the subscript $L$ refers to a partial-wave of the quantity, and the subscripts $l$ and $u$ refer "lower" and "upper" states of the target e.g. for the reaction: $e^-(E) + Z(l) \rightarrow e^-(E') + Z(u)$.

The $T$-matrix is related to the $S$-matrix and $K$-matrix. Note the reactance $R$-matrix is also known as the $K$-matrix (see Landau Appendix of K-matrix). In partial wave form we have the following relations (pg 105, 114 121, 122 and 123 Landau):

$$R_L(E) \;=\; T_L(E) + i\rho_T R_L(E) T_L(E) \tag{4.5}$$
$$\;=\; \frac{T_L(E)}{1 - i\rho_T R_L(E)} \;=\; -\frac{\tan(\delta_L)}{\rho_T} \tag{4.6}$$
$$T_L(E) \;=\; \frac{R_L(E)}{1 + i\rho_T R_L(E)} \;=\; -\frac{\exp(i\delta_L)\sin(\delta_L)}{\rho_T} \tag{4.7}$$
$$S_L \;=\; 1 - 2i\rho_T T_L(E) \;=\; \exp(2i\delta_L) \tag{4.8}$$

where $\delta_L$ is the scattering phase shift of the asymptotic wave function form and $\rho_T$ is the density of states factor (Landau pg 114). It is important to note that the $R$-matrix is real and the $S$-matrix has the useful property $S^\dagger S = 1$.

# Chapter 5

# Application: nuclear physics (Amy)

Often the types of problems that we're looking to solve involve a projectile scattering off of a medium mass or heavy target (typically neutrons or protons interacting with a nucleus of $A > 20$). There are some *ab initio* methods to solve the exact A-body problem, but these methods are limited to light mass nuclei, and even methods that can extend into the medium and heavy mass region of the nuclear chart are either limited to closed-shell nuclei (e.g. doubly magic) or are only used to solve for the energy level in the nucleus - as opposed to solving the projectile-target scattering problem. Therefore, the common technique is to freeze degrees of freedom in the nucleus and solve a two- or three-body problem instead of the full A-body problem.

To describe this interaction in the two-body system, we use phenomenological optical potentials,

$$U(R) = V(R) + i(W(R) + W_s(R)) + V_{so}(R). \tag{5.1}$$

(For charged projectiles, a Coulomb term is also included in the above equation.) These potentials typically have volume, surface, and spin-orbit terms which are parametrized as Woods-Saxons or derivatives of Woods-Saxons. In addition, to take into account flux that is lost, due to channels that are not explicitly included in the model, the potential is complex. The volume term is given as

$$V(R) = -\frac{V_R}{1 + \exp\left[(R - R_R)/a_R\right]}, \tag{5.2}$$

for the real part, and

$$W(R) = -\frac{W_R}{1 + \exp\left[(R - R_W)/a_W\right]}, \tag{5.3}$$

for the imaginary part. The surface term is usually completely imaginary

$$W_s(R) = -4a_s \frac{d}{dR} \frac{W_s}{1 + \exp\left[(R - R_s)/a_s\right]}. \tag{5.4}$$

As the incident energy of the projectile increases, the imaginary volume term becomes more important and the surface term becomes less important (there is an interplay between the two imaginary terms). Finally, the spin-orbit potential is typically also the derivative of a Woods-Saxon,

$$V_{so} = \left(\frac{\hbar}{m_\pi c}\right)^2 \frac{2\mathbf{L}\cdot\mathbf{s}}{R} \frac{d}{dR} \frac{V_{so}}{1 + \exp\left[(R - R_{so})/a_{so}\right]}. \tag{5.5}$$

Here, the potential is spherical, and the wave function is cylindrically symmetric, having the asymptotic form (with the projectile incident along the $z$-axis with momentum $\mathbf{k}$)

$$\psi^{\text{asym}}(R, \theta)\ e^{ikz} + f(\theta)\frac{e^{ikR}}{R}, \tag{5.6}$$

where $f(\theta)$ is the scattering amplitude which captures the modification of the outgoing spherical wave by the interaction with the potential. To solve for the scattering amplitude, the asymptotic form of the wave function is matched with the solution of the time-independent Schrödinger equation,

$$\left[T + U(R) - E\right]\psi(R, \theta) = 0, \tag{5.7}$$

which is written using a partial wave expansion

$$\psi(R,\theta) = \sum_{L=0}^{\infty}(2L+1)i^L P_L(\cos\theta)\frac{1}{kR}\chi_L(R), \tag{5.8}$$

where $P_L(\cos\theta)$ are the Legendre Polynomials. The Legendre Polynomials are eigenfunction of $\mathbf{L}^2$, each with an eigenvalue of $L(L+1)$. So the radial equation for each $L$ value

$$\left[-\frac{\hbar^2}{2\mu}\left(\frac{d^2}{dR^2} - \frac{L(L+1)}{R^2}\right) + U(R) - E\right]\chi_L(R) = 0, \tag{5.9}$$

can be solved independently. The equations are solved through numerical integration, $\chi_L = B_L u_L(R)$. The numerical solution gets matched to the wave function outside of the range of the nuclear interaction,

$$\chi_L(R > R_{\text{int}}) \to A_L\left[H_L^-(kR) - S_L H_L^+(kR)\right], \tag{5.10}$$

where $S_L$ is the S-matrix element for each L value, and $H^{\pm}(kR)$ are the Hankel function. The S-matrix elements are calculated numerically by equating the logarithmic derivatives of the interior and asymptotic wave functions at a point beyond the nuclear interaction, $a$,

$$R_L = \frac{1}{a}\frac{\chi_L(a)}{\chi_L'(a)} = \frac{1}{a}\frac{u_L(a)}{u_L'(a)}. \tag{5.11}$$

The scattering amplitude is then

$$f(\theta) = \frac{1}{2ik}\sum_{L=0}^{\infty}(2L+1)P_L(\cos\theta)(S_L - 1). \tag{5.12}$$

The differential cross section is then

$$\frac{d\sigma}{d\Omega}\left|f(\theta)\right|^2, \tag{5.13}$$

and the elastic, total, and reaction cross sections can be calculated from the S-matrix elements.

When the target is deformed, the deformation needs to be taken into account in solving the scattering problem (the optical potentials discussed up to this point only account for spherical targets). The main way to take the deformation into account is by expanding the optical potential in terms of spherical harmonics,

$$V(R,\theta',\phi') \approx U(R) - U'(R)\sum_{q\mu} d_{q\mu}Y_q^{\mu}(\theta',\phi'), \tag{5.14}$$

where $Y_q^{\mu}$ are the spherical harmonics, $d_{q\mu}$ is the deformation length, and $U(R)$ is the diagonal optical potential as previously described. For a nucleus that can be described with just an axial deformation, the potential can then be written

$$U(R,\hat{\xi},\hat{\mathbf{R}}) = U_0(R)Y_0^0(\hat{\mathbf{R}}) + \beta_2 U_2(R)Y_2^0(\hat{\mathbf{R}})Y_2^0(\xi), \tag{5.15}$$

where $\hat{\xi}$ is the internal coordinate of the target, $\hat{\mathbf{R}} = (\theta,\phi)$, and $\beta_2$ is the quadrupole deformation (related to $d_2$ through $\beta_2 = d_2/R_0$). Solving the scattering problem with these coupling potentials is somewhat more computationally expensive than solving the single-channel scattering problem (especially as higher order deformations are included). For example, using the potential in Eq. (5.15) - which is equivalent to coupling the ground state of the nucleus to the first $2^+$ state - the wave function becomes

$$\psi(R,\hat{\xi}) = \chi_0(R)\phi_0(\hat{\xi}) + \chi_2(R)\phi_2(\hat{\xi}), \tag{5.16}$$

where $\chi_i$ is the scattering wave function and $\phi_i$ is the wave function of the internal state of the system, satisfying the eigenvalue equation

$$H_i(\xi)\phi_i(\xi) = \epsilon_i\phi_i(\xi), \tag{5.17}$$

with eigenenergies $\epsilon_i$.

In general, an arbitrary number of coupled channels can be solved (of course, it becomes computationally expensive to do so, even with some of the approximations that can be made in the solution),

$$\left[T_L(R) + U_{\alpha\alpha} - E_{pt}\right] \chi_\alpha(R) + \sum_{\alpha' \neq \alpha} U_{\alpha\alpha'} \chi_{\alpha'}(R) = 0, \tag{5.18}$$

where $E_{pt}$ is the reaction energy minus the excitation energies of the internal states. The coupling potentials are defined as

$$U_{ij}(R) = \langle Y_{L_i}^{M_i}(\hat{\mathbf{R}}) \chi_i(\hat{\xi}) | U(R, \hat{\xi}, \hat{\mathbf{R}}) | Y_{L_j}^{M_j}(\hat{\mathbf{R}}) \chi_j(\hat{\xi}) \rangle. \tag{5.19}$$

The most direct method to solved the coupled equations is through direct numerical integration. However, solving in this manner can lead to numerical instabilities when the centrifugal barrier is large, so basis expansions are also used. In addition, another approximation can be made, the distorted-wave Born approximation. In the two-level example, this involves removing the couplings from the $i = 0$ solution, but including the couplings in the $i = 2$ solution (e.g., if we equate this to coupling the elastic and first inelastic states, the inelastic channel includes couplings from the elastic channel, but the elastic channel does not include couplings from the inelastic). Including the backward and forward couplings can become more important as the level spacing in the target nucleus becomes more dense - and the nucleus becomes more deformed.

Connecting these concepts to some of the current work in the T-2 group, in our deterministic and Monte Carlo fission codes, in order to calculate the neutron emission from the wide range of fission fragments that are created during the fission process, we use global optical models, which parametrize the depths, radii, and diffusenesses of the Woods-Saxons as functions of the reaction energy as well as the mass and charge of the target. These parametrizations are fit to reaction observables (such as differential, elastic, and total/reaction cross sections and polarization data) of available data - which is typically close to stability. However, the fission fragment are significantly far from stability, where there is limited measured information about each nucleus, and therefore, the potentials are extrapolated from the region within which they were optimized. The quality of these extrapolations is difficult to determine. (Possibly with experimental facilities like the Facility for Rare Isotope Beams at Michigan State University coming online soon, we may get more data on these isotopes.)

In addition, because the fission fragments are far from stability, they are probably also highly deformed (at least the light fragments - the heavy fragments tend to be closer to a magic number and therefore more spherical). Therefore, the approximation of a spherical potential may not be very good. However, solving the full coupled channel calculation is too slow to put in our already slow fission models. Having a faster solver would make a study like this feasible - not to mention, that the potentials would need to be re-optimized using the coupled channels solution, since the global models only solve the single channel problem (again typically due to computational challenges, since the newest global potential was constructed in 2003).

# Bibliography


[1] Xiangxiong Zhang and Chi-Wang Shu. On maximum-principle-satisfying high order schemes for scalar conservation laws. *Journal of Computational Physics*, 229(9):3091–3120, 2010.

[2] Yingda Cheng, Andrew J Christlieb, and Xinghui Zhong. Energy-conserving discontinuous galerkin methods for the vlasov–maxwell system. *Journal of Computational Physics*, 279:145–173, 2014.

[3] Philip J Morrison. Structure and structure-preserving algorithms for plasma physics. *Physics of Plasmas*, 24(5):055502, 2017.

[4] Ernst Hairer, Christian Lubich, and Gerhard Wanner. *Geometric numerical integration: structure-preserving algorithms for ordinary differential equations*, volume 31. Springer Science & Business Media, 2006.

[5] James Meiss. Hamiltonian systems. *Scholarpedia*, 2(8):1943, 2007.

[6] Karol Zyczkowski and Marek Kus. Random unitary matrices. *Journal of Physics A: Mathematical and General*, 27(12):4235, 1994.

[7] D. Turaev. Polynomial approximations of symplectic dynamics and richness of chaos in non-hpyerbolic area-preserving maps. *Nonlinearity*, 16:123–135, 2003.

[8] Babak Maboudi Afkham and Jan S Hesthaven. Structure preserving model reduction of parametric hamiltonian systems. *SIAM Journal on Scientific Computing*, 39(6):A2616–A2644, 2017.

[9] Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature communications*, 9(1):1–10, 2018.

[10] Robert I McLachlan and Christian Offen. Preservation of bifurcations of hamiltonian boundary value problems under discretisation. *Foundations of Computational Mathematics*, pages 1–38, 2020.

[11] Ivan Kobyzev, Simon Prince, and Marcus Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[12] Robert I McLachlan and G Reinout W Quispel. Splitting methods. *Acta Numerica*, 11:341, 2002.