Neural Networks for Dissipative Physics Using Morse-Feshbach Lagrangian

Veera Sundararaghavan[*]
Department of Aerospace Engineering
University of Michigan
Ann Arbor, MI
veeras@umich.edu

Megna N Shah and Jeff P Simmons
Materials and Manufacturing Directorate
Air Force Research Laboratory
Wright Patterson Air Force Base, OH
megna.shah.1@us.af.mil
jeff.simmons.3@afrl.af.mil

Abstract

There is a growing attention given to utilizing Lagrangian and Hamiltonian mechanics with network training in order to incorporate physics into the network. Most commonly, conservative systems with zero frictional losses are modeled, which do not accurately represent physical reality. This work addresses systems with dissipation using a novel neural network formulation of the Morse–Feshbach Lagrangian. The Morse-Feshbach Lagrangian models dissipative dynamics by doubling the number of dimensions of the system in order to create a 'mirror' latent representation that would counterbalance the dissipation of the observable system, making it a conservative system. We start with their formal approach by redefining a new Dissipative Lagrangian, such that the unknown matrices in the Euler-Lagrange's equations arise as partial derivatives of the Lagrangian with respect to only the observables. We then train a network from simulated training data for dissipative systems. As a model system, we choose a mechanical system with frictional dissipation and show that the approach is able to accurately capture dissipative dynamics. The approach is quite general and can be used to represent other dissipative phenomena such as Fickian diffusion.

1 Introduction

Lagrangian mechanics recently has attracted growing interest for describing physics via machine learning [1, 2, 3]. While the Lagrangian formulation has been very successful for describing systems in which quantities are conserved, real-world systems often contain dissipation. Such systems include frictional or diffusional losses, the latter applying to thermal, viscous, or chemical transport via Fick's law. This work presents the results for a mechanical system with frictional losses and shows the dynamics to be reversible by virtue of the fact that the trajectory in a lifted phase space is deterministic.

1.1 Related Work

Cranmer, *et al.* [1] and Lutter *et al.* [3] developed a generalized way of fitting a Lagrangian to a dataset and applied this to systems where quantities (e.g. energy) were conserved. Similar techniques exist in the context of learning Hamiltonians [4, 5]. However, these techniques are only for conserved systems without energy loss. Recently there has been significant interest in modeling real world dissipative systems using this concept. Sosanya and Greydanus developed 'dissipative hamiltonian neural networks' [6] which describes dissipative systems using a Hamiltonian and a dissipative term. A similar method for Lagrangian neural networks [2] has been recently developed. However, these methods are specific to the case of Rayleigh dissipation and do not have wide applicability to other physics such as diffusion.

[*] Corresponding author: Prof. Sundararaghavan, Email: veeras@umich.edu, Tel: 734-615-7242

Machine Learning and the Physical Sciences Workshop, NeurIPS 2024.

For dissipative systems, a more general theory builds on the approach proposed by Morse and Feshbach[7] for treating dissipative systems with a Lagrangian approach. They start with the differential equation for an oscillator with friction and propose a 'purely formal expression' for the Lagrangian. The key idea is to double the number of coordinates [8] so that, if energy is lost through dissipation, it is transferred to a 'mirror image' (latent) space. In this way, the energy of the doubled dimension system remains constant.

1.2 Contribution

Our new contributions include: (1) development of a neural network based on Morse and Feshbach's formal Lagrangian expression while increasing its generality, (2) Proposing a new 'Dissipative Lagrangian'($\mathcal{D}$) that can capture evolution purely in terms of observables, and (3) using a network to learn the form of the dissipative Lagrangian in the way of Cranmer, *et al.*[1]

All codes will be made freely available at publication time.

2 Preliminary Concepts

Morse and Feshbach started with the dissipative harmonic oscillator, governed by:

$$M\ddot{x} + C\dot{x} + Kx = 0 \tag{1}$$

where $M \in \Re$ is a 'mass,' allowing for inertia of a system, $C \in \Re$ is a damping factor, $K \in \Re$ is a 'stiffness,' and $x \in \Re$ is a generalized displacement variable representing the change from an equilibrium value.

If $C \neq 0$, the system is not conservative and conventional Lagrangian fails. To address this, they proposed a 'purely formal expression:'

$$\mathcal{L} = M(\dot{x}\dot{\eta}) - \frac{1}{2}C(\eta\dot{x} - x\dot{\eta}) - Kx\eta \tag{2}$$

where $\mathcal{L}$ is the Morse–Feshbach Lagrangian and $\eta \in \Re$ defines displacements of a 'mirror-image oscillator' with negative friction. This reproduces the original equations of motion via the pair of Euler-Lagrange equations:

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\eta}} - \frac{\partial \mathcal{L}}{\partial \eta} = 0 \qquad\qquad \frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{x}} - \frac{\partial \mathcal{L}}{\partial x} = 0 \tag{3}$$

Substituting in the Lagrangian in Equation 2, the first equation provides equation of motion for the real system (Equation 1) and the second evolves the mirror system with negative damping. Hence:

$$M\ddot{x} + C\dot{x} + Kx = 0 \qquad\qquad M\ddot{\eta} - C\dot{\eta} + K\eta = 0 \tag{4}$$

are the governing equations in the observable and mirror-image space, respectively.

3 Lagrangians for Multi-dimensional Dissipative Problems

We extend Morse and Feshbach's expression (Equation (2) to a field capable of representing image data containing N pixels. Replacing the scalar products with bilinear forms of matrices, we have:

$$\mathcal{L} = \sum_{ij}\left[\dot{\eta}_i M_{ij}\dot{x}_j - \frac{1}{2}(\dot{x}_i C_{ij}\eta_j - \dot{\eta}_i C_{ij} x_j) - \eta_i K_{ij} x_j\right] \tag{5}$$

where $M_{ij}, C_{ij}, K_{ij} \in \Re^{N\times N}$ and we have redefined $\eta_i, x_i \in \Re^N$, $i \in [1, N]$.

We only use observables for training. We introduce a new dissipative Lagrangian ($\mathcal{D}$):

$$\mathcal{D} = \sum_{ij}\left[\frac{1}{2}\dot{x}_i M_{ij}\dot{x}_j + \frac{1}{2}\dot{x}_i C_{ij} x_j + \frac{1}{2}x_i K_{ij} x_j\right] \tag{6}$$

The Lagrangian $\mathcal{L}$ can then be rewritten as:

$$\mathcal{L} = \sum_{i}\left[\dot{\eta}_i \frac{\partial \mathcal{D}}{\partial \dot{x}_i} - \eta_i \frac{\partial \mathcal{D}}{\partial x_i}\right] \tag{7}$$

Thus, the following equalities are satisfied:

$$\frac{\partial \mathcal{L}}{\partial \eta_i} = -\frac{\partial \mathcal{D}}{\partial x_i}, \quad \frac{\partial \mathcal{L}}{\partial \dot{\eta}_i} = \frac{\partial \mathcal{D}}{\partial \dot{x}_i} \tag{8}$$

Substituting this into the Euler-Lagrange equations, we obtain a vector form of Equation (3):

$$\frac{d}{dt}\frac{\partial \mathcal{D}}{\partial \dot{x}_i} + \frac{\partial \mathcal{D}}{\partial x_i} = 0 \qquad\qquad \frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{x}_i} - \frac{\partial \mathcal{L}}{\partial x_i} = 0 \tag{9}$$

Unlike the original equation proposed by Morse and Feshbach, the first equation that models the dynamics of the real system is now only related to the observables in the real system. The linear differential operator of the first Euler-Lagrange equation now relates to the adjoint of that in the second Euler-Lagrange equation. The matrices M, K and the symmetric part of C can be retrieved from Equation 6 as:

$$M_{ij} = \frac{\partial^2 \mathcal{D}}{\partial \dot{x}_i \partial \dot{x}_j} \qquad K_{ij} = \frac{\partial^2 \mathcal{D}}{\partial x_i \partial x_j} \qquad C_{sym} = \frac{1}{2}\left(C_{ij} + C_{ji}\right) = \frac{\partial^2 \mathcal{D}}{\partial \dot{x}_i \partial x_j} + \frac{\partial^2 \mathcal{D}}{\partial x_i \partial \dot{x}_j} \tag{10}$$

While the above formulation is for classical mechanics, the approach is general enough to model other equations. For example, Fickian diffusion of concentration field c described by the discretized equation $\dot{c}_i + \sum_j K_{ij} c_j = 0$ can be represented using Morse-Feshbach Lagrangian given by:

$$\mathcal{L} = \sum_i \frac{1}{2}(\dot{\eta}_i c_i - \eta_i \dot{c}_i) - \sum_{ij} \eta_i K_{ij} c_j \tag{11}$$

The auxiliary variable, η, can be understood as an 'undiffuser' here, so that while the observable, c, dissipates, η 'un-dissipates,' allowing complete recovery of the state of the system at every moment. If we define a dissipative Lagrangian $\mathcal{D} = \sum_i \frac{1}{2}\dot{c}_i c_i + \sum_{ij} \frac{1}{2} c_i K_{ij} c_j$, the matrix K follows as: $K_{ij} = \frac{\partial^2 \mathcal{D}}{\partial c_i \partial c_j}$.

4 Neural network architecture

The network used (Figure 1) was a fully connected network, consisting of two hidden layers of 200 neurons each and a final output layer of 1 neuron that was trained using the Adam optimizer for 1000 epochs per batch. A batch size of 1000 was used for all datasets.

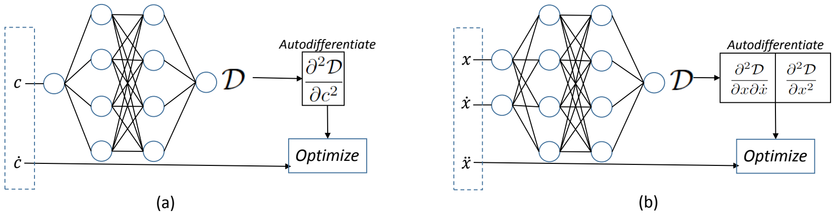

c
ċ
$\mathcal{D}$
Autodifferentiate
$\frac{\partial^2 \mathcal{D}}{\partial c^2}$
Optimize
(a)
x
ẋ
ẍ
$\mathcal{D}$
Autodifferentiate
$\frac{\partial^2 \mathcal{D}}{\partial x \partial \dot{x}}$
$\frac{\partial^2 \mathcal{D}}{\partial x^2}$
Optimize
(b)

Figure 1: Neural network architectures for (a) diffusion problem and (b) the dissipative mechanics problems

For the dissipative mechanics problems (Fig 1b), two initial condition vectors (position and velocity) were used. In this case, the input layer took in both x and $\dot{x}$ at time t as shown in Figure 1(b). Each training set contained three vectors $(x, \dot{x}, \ddot{x})$ at a given time step. The acceleration data was only used in the loss function. The unknown stiffness and damping matrices were obtained via autodifferentiation of the output $\mathcal{D}$ and used in the following loss function:

$$L_{DLNN} = || \sum_j \frac{\partial^2 \mathcal{D}}{\partial \dot{x}_i \partial \dot{x}_j} \ddot{x}_{data,j} + \sum_j \left(\frac{\partial^2 \mathcal{D}}{\partial \dot{x}_i \partial x_j} + \frac{\partial^2 \mathcal{D}}{\partial x_i \partial \dot{x}_j} \right) \dot{x}_{data,j} + \sum_j \frac{\partial^2 \mathcal{D}}{\partial x_i \partial x_j} x_{data,j} || \tag{12}$$

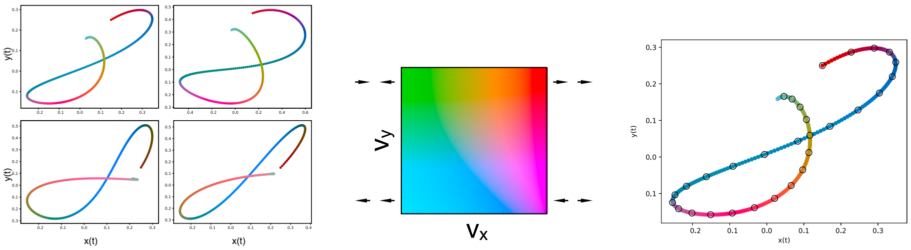
During inference, the input positions and velocities are used to compute the system matrices via autodifferentiation of $\mathcal{D}$. The acceleration is inferred using Eq. 1, and time-stepping is used to propagate the solution. A similar network for modelling Fickian diffusion is shown in Fig. 1a.

Parameter	Value
K_{11}	1
K_{22}	1
K_{12}	-0.4
C_{11}	0.1
C_{12}	0.1
C_{22}	0.2
M	$\mathbb{1}_{2\times 2}$

Table 1: Mass–spring–damper system parameters.

Problem	This work	HNN [6]	LNN [2]	BaselineNN
Mechanics	1.06×10^{-3}	0.97×10^{-3}	1.18×10^{-3}	6.76×10^{-3}
Diffusion	1.08×10^{-3}	NA	NA	8.21×10^{-3}

Table 2: Maximum error comparisons between the proposed method and current techniques. Single oscillator (M=K=C=1) system for mechanics and ten pixel Fickian diffusion (K is the Lehmer matrix). HNN and LNN models are not applicable to diffusion problems.



(a) (b) (c)

Figure 2: (a) Few training set examples. Each set consists of an array of 300 points with coordinates $[x(t), y(t), v_x(t), v_y(t)]$, the positions and velocities of each of the masses. $x(t)$ and $y(t)$ are plotted as positions as shown. $v_x(t)$ and $v_y(t)$ are plotted as colors from the CIE LAB color model[10] with luminosity (L) channel varying from 30 to 70 from initial time to end time. $v_x(t)$ is represented by the a-channel and $v_y(t)$ by the b-channel.(b) Legend of velocities vs. color representation, (c) Comparison between predicted and ground truth values for a test trajectory. Large dark circles illustrate ground truth result. For clarity, only every tenth point is marked. In the computation, the L_2 norm was used for all 300 points.

5 Experiments

The first test case considered a simple problem of a mass–spring–damper system with parameters given in Table 1. Training data contained simulated trajectories, based on time integration using an explicit Runge-Kutta (4,5) formula, the Dormand-Prince pair [9], with initial positions and velocities taken from a square grid formed by all permutations of $x_i, \dot{x}_i \in [0.2, 0.4], i \in [1, 2]$. Example trajectories are shown in Figure 2(a). The test data contained 625 new extrapolatory trajectories $x_i, \dot{x}_i \in [0.15, 0.25, 0.3, 0.35, 0.45]$.

6 Results

Figure 2 (c) shows a trajectory of the test set predicted by the DLNN compared to the ground truth solution. The mean RMS error for the test set of all 625 trajectories, as compared to ground truth, was 1.54×10^{-3}, representing an ability to accurately predict the physical behavior of the system.. In Table 2, we compare the maximum error in the extrapolatory mode for the current model with the model implementations for Refs [2, 6] and a baseline neural network that is directly trained to predict the acceleration from the position and velocity data. The present model is comparable to the published models and is superior to the baseline NN, but is also broadly applicable to other equations such as Fickian diffusion. The errors measure the maximum RMS error of the result compared to the ground truth in an extrapolatory case where the initial conditions were halved compared to the training set.

7 Conclusions

We have developed a novel Lagrangian neural network model of dissipative systems. This approach is based on the Morse and Feshbach[7] method of doubling coordinates to lift the problem to a space where the system is conservative. The method is quite general and can be easily extended to other physical equations (including diffusion, viscous flow, electromagnetics, and Schrodinger's equation) significantly broadening our understanding of physics using observed data.

References

[1] Miles Cranmer, Sam Greydanus, Stephan Hoyer, Peter Battaglia, David Spergel, and Shirley Ho. Lagrangian neural networks. arXiv preprint arXiv:2003.04630, 2020.

[2] Shanshan Xiao, Jiawei Zhang, and Yifa Tang. Generalized lagrangian neural networks. arXiv preprint arXiv:2401.03728, 2024.

[3] Michael Lutter, Christian Ritter, and Jan Peters. Deep lagrangian networks: Using physics as model prior for deep learning. arXiv preprint arXiv:1907.04490, 2019.

[4] Samuel Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. Advances in neural information processing systems, 32, 2019.

[5] Peter Toth, Danilo Jimenez Rezende, Andrew Jaegle, Sébastien Racanière, Aleksandar Botev, and Irina Higgins. Hamiltonian generative networks. arXiv preprint arXiv:1909.13789, 2019.

[6] Andrew Sosanya and Sam Greydanus. Dissipative hamiltonian neural networks: Learning dissipative and conservative dynamics separately. arXiv preprint arXiv:2201.10085, 2022.

[7] Philip M Morse and Herman Feshbach. Methods of theoretical physics. American Journal of Physics, 22(6):410–413, 1954.

[8] Harry Bateman. On dissipative systems and related variational principles. Physical Review, 38(4):815, 1931.

[9] John R Dormand and Peter J Prince. A family of embedded runge-kutta formulae. Journal of computational and applied mathematics, 6(1):19–26, 1980.

[10] CIE Standard et al. Colorimetry-part 4: Cie 1976 l* a* b* colour space. International Standard, pages 2019–06, 2007.