Uncovering the Lagrangian from observations of trajectories

Yakov Berchenko-Kogan

June 1, 2011

Abstract

We approach the problem of automatically modeling a mechanical system from data about its dynamics, using a method motivated by variational integrators. We write the discrete Lagrangian as a quadratic polynomial with varying coefficients, and then use the discrete Euler-Lagrange equations to numerically solve for the values of these coefficients near the data points. This method correctly modeled the Lagrangian of a simple harmonic oscillator and a simple pendulum, even with significant measurement noise added to the trajectories.

1 Introduction

Much research has been done to automatically recover the rules that underlie large sets of data. For example, Saul and Roweis [3] created an algorithm that, given a large set of points that all lie on a low-dimensional submanifold embedded in some high-dimensional space, finds a low-dimensional parametrization of the points on the submanifold. In the field of dynamical systems, Schmidt and Lipson [4] created an algorithm that uses evolutionary computation to find meaningful conserved quantities and invariant equations of a system from observations of the system dynamics, including Hamiltonians and Lagrangians. Unfortunately, their approach is currently limited to finding simple expressions, and thus cannot yet be applied to complex systems.

The method presented in this paper for discovering the Lagrangian of a dynamical system is motivated by the theory of variational integrators. Variational integrators simulate a system with known Lagrangian $L(q, \dot{q})$ by writing a discrete Lagrangian L_d, a function of pairs of points in the configuration space, where

$$L_d(x, y) = \tau L\left(\frac{x+y}{2}, \frac{y-x}{\tau}\right),$$

where τ is the time step used in the simulation. Then, the discrete motion of the system is described by the discrete Euler-Lagrange equations

$$D_2 L_d(x, y) + D_1 L_d(y, z) = 0,$$

where x, y, and z are consecutive points on a trajectory. Solving the discrete Euler-Lagrange equations for z yields an update rule for the variational integrator. For a more detailed introduction to variational integrators, see [5]. Variational integrators capture the global behavior of a system extremely well, including the conservation of energy and momenta conserved in the original system, which has applications in many fields, such as fluid mechanics [1], [2]. It is hoped that methods for modeling systems based on these ideas will share some of these useful properties.

In order to recover the Lagrangian, we again use the discrete Euler-Lagrange equations, but instead of solving them for z, we solve them for L_d. More precisely, we write the discrete Lagrangian $L_d(x, y)$ as a quadratic polynomial with coefficients that vary in x and y, and then we use the discrete Euler-Lagrange equations written for several nearby triplets (x_i, y_i, z_i) to numerically solve for the values of these coefficients locally. Of course, since many Lagrangians are equivalent in the sense that they yield identical equations of motion, it is impossible to solve for all of the coefficients. However, it is possible to solve for the Lagrangian up to this equivalence, which is enough to make predictions about new trajectories of the system.

This approach for deriving the Lagrangian of a system from observations of its trajectories would eventually provide a new way to automatically model complex behaviors, which would impact both research into these behaviors and industries that rely on these models. Numerically inferring a Lagrangian from captured data of a system would allow one to predict the system's behavior, even for seemingly complex systems such as a butterfly's wing and global weather patterns, for which accurate modeling still eludes us today.

2 Recovering a Lagrangian in one dimension

For now, we will work with one-dimensional systems. Given a data set, we wish to find a Lagrangian L such that for each triple of consecutive points (x, y, z) on a trajectory, we have the discrete Euler-Lagrange equations

$$D_2L(x, y) + D_1L(y, z) = 0.$$

On a small neighborhood, we would like to approximate the Lagrangian as a quadratic $L(x, y) \approx ax^2 + 2bxy + cy^2 + dx + ey + f$. However, since the coefficients of the quadratic vary with x and y, it is more appropriate to write

$$L(x, y) = a(x, y)x^2 + 2b(x, y)xy + c(x, y)y^2 + d(x, y)x + e(x, y)y + f(x, y), \tag{1}$$

where a, b, c, d, e, and f are real-valued functions of x and y. On a small neighborhood, the change in a, b, c, d, e, and f should be negligible, a requirement that we will make more precise later.

We first introduce notation that will be helpful when generalizing to higher dimensional systems. Namely, we assume that we have functions

$$\begin{aligned} H &\colon \mathbb{R} \times \mathbb{R} \to \{\text{quadratic forms in two variables}\}, \\ J &\colon \mathbb{R} \times \mathbb{R} \to \{\text{linear functionals in two variables}\}, \text{ and} \\ K &\colon \mathbb{R} \times \mathbb{R} \to \mathbb{R}, \end{aligned}$$

such that $L(x, y) = H(x, y)(x, y)+J(x, y)(x, y)+K(x, y)$ and such that, on a small neighborhood, the change in the functions H, J, and K is negligible.

We can write the equation for the Lagrangian in matrix form

$$L(x, y) = \begin{pmatrix} x & y \end{pmatrix} H(x, y) \begin{pmatrix} x \\ y \end{pmatrix} + J(x, y) \begin{pmatrix} x \\ y \end{pmatrix} + K(x, y).$$

This equation is the same as equation (1) with the substitutions $H = \left(\begin{smallmatrix} a & b \\ b & c \end{smallmatrix}\right)$, $J = \left(\begin{smallmatrix} d & e \end{smallmatrix}\right)$, and $K = f$. We compute

$$\begin{aligned} D_1L(x, y) = \begin{pmatrix} 1 & 0 \end{pmatrix} H(x, y) \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} x & y \end{pmatrix} H(x, y) \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} x & y \end{pmatrix} D_1H(x, y) \begin{pmatrix} x \\ y \end{pmatrix} \\ + J(x, y) \begin{pmatrix} 1 \\ 0 \end{pmatrix} + D_1J(x, y) \begin{pmatrix} x \\ y \end{pmatrix} + D_1K(x, y). \end{aligned}$$

Since $H(x, y)$ is a symmetric matrix, we can simply this expression to

$$D_1L(x, y) = 2 \begin{pmatrix} 1 & 0 \end{pmatrix} H(x, y) \begin{pmatrix} x \\ y \end{pmatrix} + J(x, y) \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} x & y \end{pmatrix} D_1H(x, y) \begin{pmatrix} x \\ y \end{pmatrix} + D_1J(x, y) \begin{pmatrix} x \\ y \end{pmatrix} + D_1K(x, y).$$

Likewise,

$$D_2L(x, y) = 2 \begin{pmatrix} 0 & 1 \end{pmatrix} H(x, y) \begin{pmatrix} x \\ y \end{pmatrix} + J(x, y) \begin{pmatrix} 0 \\ 1 \end{pmatrix} + \begin{pmatrix} x & y \end{pmatrix} D_2H(x, y) \begin{pmatrix} x \\ y \end{pmatrix} + D_2J(x, y) \begin{pmatrix} x \\ y \end{pmatrix} + D_2K(x, y).$$

We can now make our assumption that H, J, and K vary slowly more precise. Namely, we require that, for all pairs (x, y) of consecutive points *in the data set*, we have

$$\begin{pmatrix} x & y \end{pmatrix} D_1 H(x,y) \begin{pmatrix} x \\ y \end{pmatrix} + D_1 J(x,y) \begin{pmatrix} x \\ y \end{pmatrix} + D_1 K(x,y) \ll 2 \begin{pmatrix} 1 & 0 \end{pmatrix} H(x,y) \begin{pmatrix} x \\ y \end{pmatrix} + J(x,y) \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \text{ and}$$
$$\begin{pmatrix} x & y \end{pmatrix} D_2 H(x,y) \begin{pmatrix} x \\ y \end{pmatrix} + D_2 J(x,y) \begin{pmatrix} x \\ y \end{pmatrix} + D_2 K(x,y) \ll 2 \begin{pmatrix} 0 & 1 \end{pmatrix} H(x,y) \begin{pmatrix} x \\ y \end{pmatrix} + J(x,y) \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

The interpretation of this condition is that if the time step in our data set is too large, the data will not be fine enough to resolve changes in the nature of the Lagrangian, and therefore recovering the Lagrangian of the system from this data set will be impossible.

Making the assumption above, we have

$$D_1 L(x,y) \approx 2 \begin{pmatrix} 1 & 0 \end{pmatrix} H(x,y) \begin{pmatrix} x \\ y \end{pmatrix} + J(x,y) \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \text{ and}$$
$$D_2 L(x,y) \approx 2 \begin{pmatrix} 0 & 1 \end{pmatrix} H(x,y) \begin{pmatrix} x \\ y \end{pmatrix} + J(x,y) \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

Assuming further that if (x, y, z) are consecutive points on a trajectory then the coefficients $H = \left(\begin{smallmatrix} a & b \\ b & c \end{smallmatrix}\right)$, $J = \left(\begin{smallmatrix} d & e \end{smallmatrix}\right)$, and $K = f$ are approximately equal when evaluated at (x, y) and when evaluated at (y, z), we can compute

$$\begin{aligned} 0 = D_2 L(x,y) + D_1 L(y,z) &\approx (2bx + 2cy + e)|_{(x,y)} + (2ay + 2bz + d)|_{(y,z)} \\ &\approx 2(a+c)y + 2b(x+z) + (d+e). \quad (2) \end{aligned}$$

We obtain the values of x, y, and z from our data, and we would like to use them to compute the values of a, b, c, d, e, and f. Clearly, we cannot solve for six unknowns with only one equation. However, if we strengthen our assumption that a, b, c, d, e, and f are roughly constant not only between the pairs (x, y) and (y, z) but also for a few nearby consecutive pairs (x_i, y_i), and (y_i, z_i), we can obtain as many equations as we need. Still, no matter how many equations we have, we will not be able to compute f since it does not appear in the equations, we will be able to compute $a + c$ and $d + e$ but not a, c, d, and e individually, and any solution we obtain could be scaled to obtain another solution. There is no way around this, since the Lagrangian for a system is not unique. In particular, if a system has Lagrangian L, then we obtain the same equations of motion with the Lagrangian

$$L'(x,y) = \alpha L(x,y) + \beta(y^2 - x^2) + \gamma(y - x) + \delta,$$

for any choice of real parameters α, β, γ, and δ, which correspond to scaling, changing a and b while fixing $a + b$, changing d and e while fixing $d + e$, and changing f, respectively.

However, knowing $(a + c, b, d + e)$ up to scaling is all the information we need about the Lagrangian to be able to compute new trajectories, so the fact that we cannot know more about the Lagrangian is not troubling. In order to solve for $(a+c, b, d+e)$ up to scaling, we need two equations of the form of equation (2). A natural choice is to use four consecutive points (w, x, y, z) along a trajectory, and write equation (2) for the triplets (w, x, y) and (x, y, z).

Because real data will have error, it may be appropriate to use more than the necessary number of triplets and then minimize the sum of squares of $D_2 L(x, y) + D_1 L(y, z)$ instead of solving $D_2 L(x, y) + D_1 L(y, z) = 0$. However, for now, we will compute $(a + c, b, d + e)$ up to scaling by assuming $a + c \neq 0$ and computing $\frac{b}{a+c}$ and $\frac{d+e}{a+c}$. We can then write the following system of equations for four consecutive points (w, x, y, z) along a trajectory.

$$2y + 2\left(\frac{b}{a+c}\right)(x+z) + \left(\frac{d+e}{a+c}\right) = 0,$$
$$2x + 2\left(\frac{b}{a+c}\right)(w+y) + \left(\frac{d+e}{a+c}\right) = 0.$$

As long as $x + z \neq w + y$, we can solve this system to obtain

$$\frac{b}{a+c} = \frac{x-y}{x+z-w-y}, \qquad \frac{d+e}{a+c} = 2 \cdot \frac{y(y+w) - x(x+z)}{x+z-w-y}. \tag{3}$$

We now apply these equations to some simple one-dimensional systems.

3 The Simple Harmonic Oscillator

3.1 Exact values of the parameters

Normalizing the mass and spring constant to one, the Lagrangian of the system is

$$L(q, \dot{q}) = \tfrac{1}{2}\left(\dot{q}^2 - q^2\right).$$

Given a time step τ, we can write the discrete Lagrangian as

$$L(x,y) = \frac{\tau}{2}\left(\left(\frac{y-x}{\tau}\right)^2 - \left(\frac{x+y}{2}\right)^2\right) = \left(\frac{1}{2\tau} - \frac{\tau}{8}\right)(x^2+y^2) - \left(\frac{1}{\tau} + \frac{\tau}{4}\right)xy.$$

Recall that we approximate the Lagrangian as

$$L(x,y) = ax^2 + 2bxy + cy^2 + dx + ey + f.$$

In this particular case, the Lagrangian is quadratic, so the approximation is exact, and we see that

$$a = c = \frac{1}{2\tau} - \frac{\tau}{8}, \qquad b = -\frac{1}{2\tau} - \frac{\tau}{8}, \qquad d = e = f = 0.$$

In particular, since the Lagrangian is quadratic, a, b, c, d, e, and f are global constants that do not depend on x and y. Normalizing, we observe that

$$\frac{b}{a+c} = -\frac{1}{2} \cdot \frac{4+\tau^2}{4-\tau^2}, \qquad \frac{d+e}{a+c} = 0.$$

3.2 Parameters computed from trajectories

Next, we can recover $\frac{b}{a+c}$ and $\frac{d+e}{a+c}$ from the trajectories themselves. A general trajectory of this system is $q(t) = A(\sin t + \phi)$. For simplicity, we will set $A = 1$ and $\phi = 0$, but the following manipulations will work in general. Four consecutive points along the trajectory have the form

$$\begin{aligned}
w &= \sin\left(t - \tfrac{3\tau}{2}\right) = \sin t \cos\tfrac{3\tau}{2} - \cos t \sin\tfrac{3\tau}{2}, \\
x &= \sin\left(t - \tfrac{\tau}{2}\right) = \sin t \cos\tfrac{\tau}{2} - \cos t \sin\tfrac{\tau}{2}, \\
y &= \sin\left(t + \tfrac{\tau}{2}\right) = \sin t \cos\tfrac{\tau}{2} + \cos t \sin\tfrac{\tau}{2}, \\
z &= \sin\left(t + \tfrac{3\tau}{2}\right) = \sin t \cos\tfrac{3\tau}{2} + \cos t \sin\tfrac{3\tau}{2}.
\end{aligned}$$

We then compute

$$\begin{aligned}
x + z - w - y &= 2\cos t\left(\sin\tfrac{3\tau}{2} - \sin\tfrac{\tau}{2}\right) = 4\cos t \cos\tau \sin\tfrac{\tau}{2}, \\
x - y &= -2\cos t \sin\tfrac{\tau}{2},
\end{aligned}$$

and we compute

$$\begin{aligned}
y(y+w) - x(x+z) &= (y^2 - x^2) + (yw - xz) \\
&= (4\sin t\cos t \sin\tfrac{\tau}{2}\cos\tfrac{\tau}{2}) - (2\sin t\cos t\sin\tfrac{3\tau}{2}\cos\tfrac{\tau}{2} - 2\sin t\cos t\sin\tfrac{\tau}{2}\cos\tfrac{3\tau}{2}) \\
&= 2\sin t\cos t\left(\sin\left(2\cdot\tfrac{\tau}{2}\right) - \sin\left(\tfrac{3\tau}{2} - \tfrac{\tau}{2}\right)\right) = 0.
\end{aligned}$$

Therefore, the values of $\frac{b}{a+c}$ and $\frac{d+e}{a+c}$ computed from the trajectories are

$$\frac{b}{a+c} = \frac{x-y}{x+z-w-y} = -\frac{1}{2\cos\tau}, \qquad \frac{d+e}{a+c} = 2\frac{y(w+y)-x(x+z)}{x+z-w-y} = 0.$$

Note that

$$-\frac{1}{2\cos\tau} = -\frac{1}{2}\cdot\frac{4+\tau^2}{4-\tau^2} + o(\tau^2).$$

Hence, the Lagrangian parameters computed from the trajectories match the exact values to second order in the time step τ. Next, we move on to a slightly more complex example.

4 The Simple Pendulum

4.1 Exact values of the parameters

Normalizing the gravitational constant and the mass and length of the pendulum to one, the Lagrangian of the simple pendulum is

$$L(\theta,\dot\theta) = \tfrac{1}{2}\dot\theta^2 - (1-\cos\theta).$$

Again, we write the discrete Lagrangian with time step τ.

$$L(x,y) = \tau\left(\frac{1}{2}\left(\frac{y-x}{\tau}\right)^2 - \left(1-\cos\left(\frac{x+y}{2}\right)\right)\right).$$

We compute

$$D_1L(x,y) = \tau\left(\frac{x-y}{\tau^2} - \frac{1}{2}\sin\left(\frac{x+y}{2}\right)\right) \qquad D_2L(x,y) = \tau\left(\frac{y-x}{\tau^2} - \frac{1}{2}\sin\left(\frac{x+y}{2}\right)\right)$$
$$D_1D_1L(x,y) = \tau\left(\frac{1}{\tau^2} - \frac{1}{4}\cos\left(\frac{x+y}{2}\right)\right) \qquad D_2D_2L(x,y) = \tau\left(\frac{1}{\tau^2} - \frac{1}{4}\cos\left(\frac{x+y}{2}\right)\right)$$
$$D_1D_2L(x,y) = \tau\left(-\frac{1}{\tau^2} - \frac{1}{4}\cos\left(\frac{x+y}{2}\right)\right)$$

Using a second order Taylor approximation at an arbitrary point (x,y), we see that

$$\begin{aligned}
L(x,y) &\approx L(x_0,y_0) + D_1L(x_0,y_0)(x-x_0) + D_2L(x_0,y_0)(y-y_0)\\
&\quad + \tfrac{1}{2}D_1D_1L(x_0,y_0)(x-x_0)^2 + \tfrac{1}{2}D_2D_2L(x_0,y_0)(y-y_0^2) + D_1D_2L(x_0,y_0)(x-x_0)(y-y_0)\\
&= \tfrac{1}{2}D_1D_1L(x_0,y_0)x^2 + D_1D_2L(x_0,y_0)xy + \tfrac{1}{2}D_2D_2L(x_0,y_0)y^2\\
&\quad + (D_1L(x_0,y_0) - D_1D_1L(x_0,y_0)x_0 - D_1D_2L(x_0,y_0)y_0)x\\
&\quad + (D_2L(x_0,y_0) - D_2D_2L(x_0,y_0)y_0 - D_1D_2L(x_0,y_0)x_0)y\\
&\quad + L(x_0,y_0) - D_1L(x_0,y_0)x_0 - D_2L(x_0,y_0)y_0\\
&\qquad + \tfrac{1}{2}D_1D_1L(x_0,y_0)x_0^2 + D_1D_2L(x_0,y_0)x_0y_0 + \tfrac{1}{2}D_2D_2L(x_0,y_0)y_0^2\\
&= ax^2 + 2bxy + cy^2 + dx + ey + f.
\end{aligned}$$

Hence, at (x_0,y_0), we compute

$$a = c = \frac{1}{2\tau} - \frac{\tau}{8}\cos\left(\frac{x_0+y_0}{2}\right), \qquad b = -\frac{1}{2\tau} - \frac{\tau}{8}\cos\left(\frac{x_0+y_0}{2}\right),$$

and we compute

$$d = \frac{x_0 - y_0}{\tau} - \frac{\tau}{2}\sin\left(\frac{x_0+y_0}{2}\right) - \frac{x_0}{\tau} + \frac{\tau x_0}{4}\cos\left(\frac{x_0+y_0}{2}\right) + \frac{y_0}{\tau} + \frac{\tau y_0}{4}\cos\left(\frac{x_0+y_0}{2}\right)$$
$$= -\frac{\tau}{2}\sin\left(\frac{x_0+y_0}{2}\right) + \frac{\tau}{4}(x_0+y_0)\cos\left(\frac{x_0+y_0}{2}\right).$$

Likewise,

$$e = \frac{y_0 - x_0}{\tau} - \frac{\tau}{2}\sin\left(\frac{x_0+y_0}{2}\right) - \frac{y_0}{\tau} + \frac{\tau y_0}{4}\cos\left(\frac{x_0+y_0}{2}\right) + \frac{x_0}{\tau} + \frac{\tau x_0}{4}\cos\left(\frac{x_0+y_0}{2}\right)$$
$$= -\frac{\tau}{2}\sin\left(\frac{x_0+y_0}{2}\right) + \frac{\tau}{4}(x_0+y_0)\cos\left(\frac{x_0+y_0}{2}\right).$$

Therefore,

$$\frac{b}{a+c} = -\frac{1}{2}\cdot\frac{4+\tau^2\cos\left(\frac{x_0+y_0}{2}\right)}{4-\tau^2\cos\left(\frac{x_0+y_0}{2}\right)}, \qquad \frac{d+e}{a+c} = 4\tau^2\cdot\frac{\frac{x_0+y_0}{2}\cdot\cos\left(\frac{x_0+y_0}{2}\right) - \sin\left(\frac{x_0+y_0}{2}\right)}{4-\tau^2\cos\left(\frac{x_0+y_0}{2}\right)}. \tag{4}$$

4.2 Parameters Computed from Trajectories

Using various trajectories of the pendulum computed using Matlab's `ode45` and `ode23` integrators, we applied the formulas (3) to compute $\frac{b}{a+c}$ and $\frac{d+e}{a+c}$. The Lagrangian parameters computed from the trajectories agreed approximately with the values that we expected from (4), as can be seen in Figure 1.

One would expect uncertainties in the computed parameters when the denominator of the expressions in equations (3) is near zero, which occurs when the velocity of the system is near zero. However, the parameters computed from the trajectories had additional unexpected spikes, whose locations depended on the integration method used, as seen in Figure 2. I suspect that these spikes are due to changes in the timestep used by the integrator or to beats between the timestep used by the integrator and the timestep at which I sampled the simulated trajectory. I suspect that such isolated spikes would not be present if the parameters were computed from data with random normally distributed error.

Adding random noise to the simulated trajectories as shown in Figure 3 showed that the calculation of the parameters is very sensitive to noise. In addition, decreasing the time step made the parameters even more sensitive to noise, which is unsurprising since four consecutive points provide less information about the system if they are closer together. These observations suggest that more points should be used to estimate the parameters, which I will investigate next.

5 A new choice of parameters

5.1 Localization

One problem with the parameters used above is that $d+e$ does not respect the 2π-translational symmetry of the pendulum. We solve this problem by creating new parameters to replace $d+e$. We begin by localizing our equation for the Lagrangian at a point p, defining new parameters J_p and K_p. We write

$$L(x,y) = \begin{pmatrix} x & y \end{pmatrix} H(x,y) \begin{pmatrix} x \\ y \end{pmatrix} + J(x,y)\begin{pmatrix} x \\ y \end{pmatrix} + K(x,y)$$
$$= \begin{pmatrix} x-p & y-p \end{pmatrix} H(x,y) \begin{pmatrix} x-p \\ y-p \end{pmatrix} + J_p(x,y)\begin{pmatrix} x-p \\ y-p \end{pmatrix} + K_p(x,y).$$

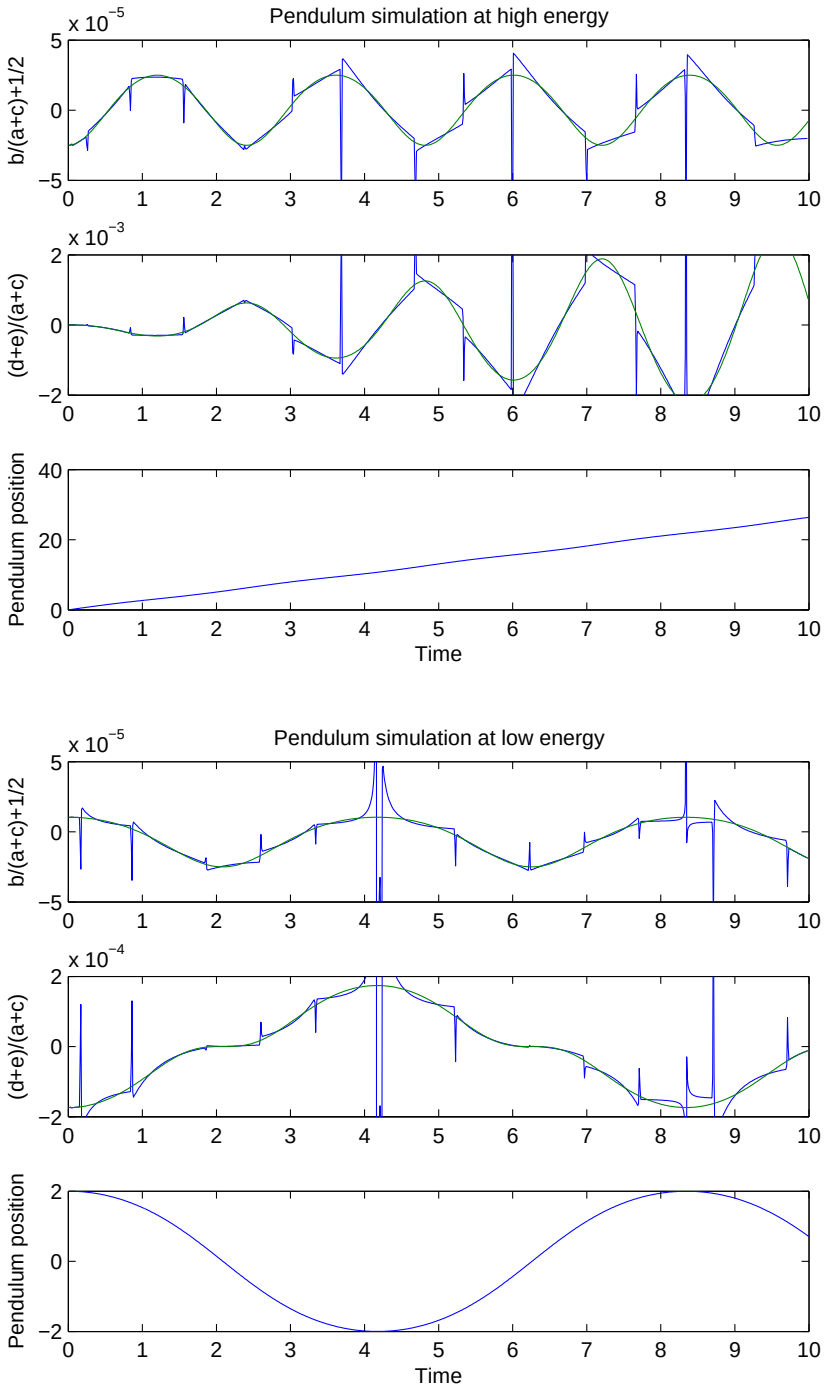


Figure 1: The Lagrangian parameters estimated from the pendulum trajectory at both high and low energy. The values of the parameters that we expect are in green, and the values of the parameters computed from the trajectories are in blue.

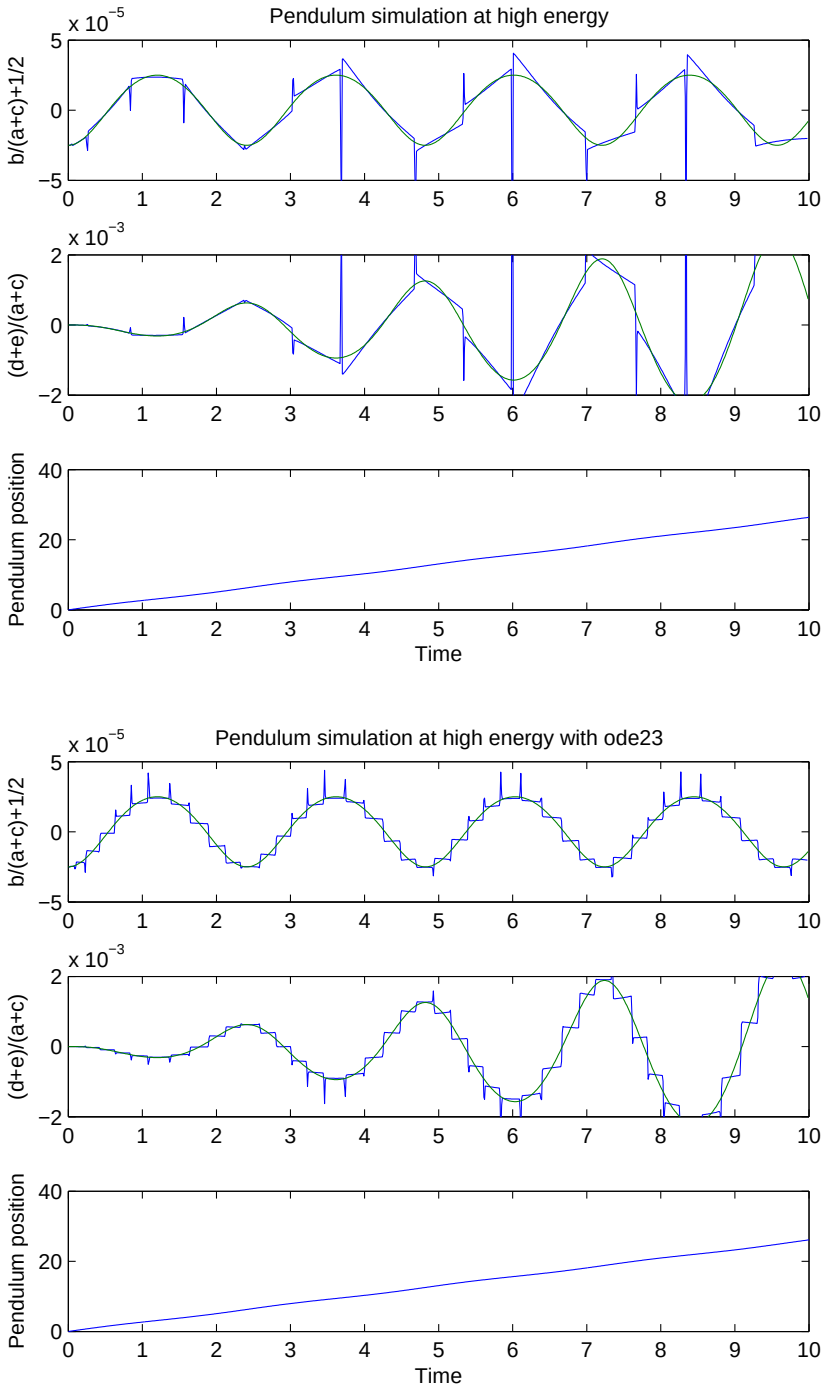


Figure 2: A comparison of the high energy plot from Figure 1 with the same plot, except with `ode23` used instead of `ode45` to simulate the pendulum. As can be seen, the locations of the sporadic spikes depend on the integration method.

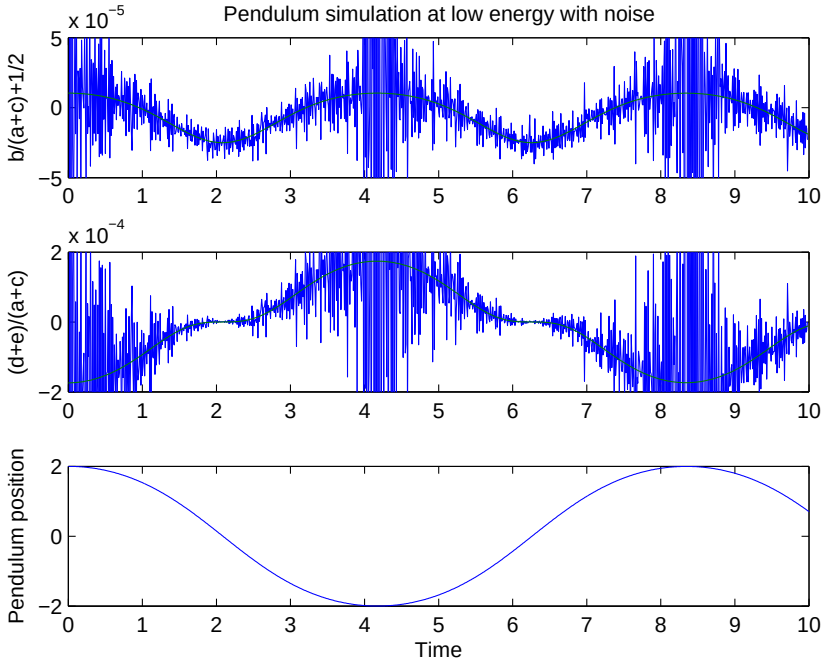


Figure 3: Adding noise with standard deviation $1 \cdot 10^{-7}$ to the pendulum position resulted in significant noise in the parameters computed from the trajectory compared to Figure 1.

One can compute that the new parameters are related to the old ones by the equations

$$J_p(x,y) = 2\begin{pmatrix} p & p \end{pmatrix} H(x,y) + J(x,y),$$
$$K_p(x,y) = \begin{pmatrix} p & p \end{pmatrix} H(x,y) \begin{pmatrix} p \\ p \end{pmatrix} + J(x,y) \begin{pmatrix} p \\ p \end{pmatrix} + K(x,y).$$

Under the same assumptions as earlier, we neglect the change in H, J_p, and K_p to compute

$$D_1 L(x,y) \approx 2\begin{pmatrix} 1 & 0 \end{pmatrix} H(x,y) \begin{pmatrix} x-p \\ y-p \end{pmatrix} + J_p(x,y) \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \text{ and}$$
$$D_2 L(x,y) \approx 2\begin{pmatrix} 0 & 1 \end{pmatrix} H(x,y) \begin{pmatrix} x-p \\ y-p \end{pmatrix} + J_p(x,y) \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

Like before, we assume that H, J_p, and K are approximately constant for a triplet of consecutive points (x,y,z) on a trajectory, and we use the discrete Euler-Lagrange equations to compute

$$0 = D_1 L(y,z) + D_2 L(x,y) = 2\begin{pmatrix} 1 & 0 \end{pmatrix} H \begin{pmatrix} y-p \\ z-p \end{pmatrix} + J_p \begin{pmatrix} 1 \\ 0 \end{pmatrix} + 2\begin{pmatrix} 0 & 1 \end{pmatrix} H \begin{pmatrix} x-p \\ y-p \end{pmatrix} + J_p \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{5}$$

Writing $H = \left(\begin{smallmatrix} a & b \\ b & c \end{smallmatrix}\right)$ and $J_p = \left(\begin{smallmatrix} d_p & e_p \end{smallmatrix}\right)$, we write the above equation as

$$\begin{aligned} 0 = 2a(y-p) + 2b(z-p) + d_p + 2b(x-p) + 2c(y-p) + e_p \\ = 2(a+c)(y-p) + 2b(x-p+z-p) + (d_p + e_p). \end{aligned}$$

Earlier, we used the analogous equation (2) to estimate $a(x_0,y_0)+c(x_0,y_0)$, $b(x_0,y_0)$, and $d(x_0,y_0)+e(x_0,y_0)$. Now, we will use equation (5) to estimate $a(x_0,y_0)+c(x_0,y_0)$, $b(x_0,y_0)$, and $d_p(x_0,y_0)+e_p(x_0,y_0)$, where $p = \frac{1}{2}(x_0+y_0)$. Note that these new parameters are related to the old ones by the equation

$$d_{(x_0+y_0)/2} + e_{(x_0+y_0)/2} = (x_0+y_0)(a+c+2b) + (d+e).$$

5.2 Scaling

The last step in defining the new parameters is to address the issue that $d_p + e_p$ and $a+c$ have different units. The choice of renormalization is motivated by the Taylor approximation of the Lagrangian

$$\begin{aligned} L(x,y) &= \begin{pmatrix} x-p & y-p \end{pmatrix} \begin{pmatrix} a & b \\ b & c \end{pmatrix} \begin{pmatrix} x-p \\ y-p \end{pmatrix} + \begin{pmatrix} d_p & e_p \end{pmatrix} \begin{pmatrix} x-p \\ y-p \end{pmatrix} + K_p(x,y) \\ &\approx \frac{1}{2} \begin{pmatrix} x-p & y-p \end{pmatrix} \begin{pmatrix} D_1D_1L & D_1D_2L \\ D_2D_1L & D_2D_2L \end{pmatrix} \begin{pmatrix} x-p \\ y-p \end{pmatrix} + \begin{pmatrix} D_1L & D_2L \end{pmatrix} \begin{pmatrix} x-p \\ y-p \end{pmatrix} + L, \end{aligned}$$

along with the Taylor approximation of the discrete Euler-Lagrange equations

$$0 = D_1L(y,z) + D_2L(x,y) \approx D_1L(x,y) + D_1D_1L(x,y)(y-x) + D_2D_1L(x,y)(z-y) + D_2L(x,y).$$

The first equation suggests that $2a$, $2b$, and $2c$ are of the same order as the second derivatives of L, and that d_p and e_p are of the same order as the first derivatives of L. The second equation suggests that the first derivatives of L are of the same order as the second derivatives of L times $y_0 - x_0$. From here, we conclude that an appropriate choice of parameters is

$$A(x_0,y_0) := (a+c)\,\|y_0-x_0\|, \qquad B(x_0,y_0) := 2b\,\|y_0-x_0\|, \qquad D(x_0,y_0) := d_{(x_0+y_0)/2} + e_{(x_0+y_0)/2}.$$

With these parameters, the discrete Euler-Lagrange equations become

$$2A \cdot (y-p) + B \cdot (x-p+z-p) + D\,\|y_0-x_0\| = 0. \tag{6}$$

At a pair of points (x_0,y_0), we will estimate the value of the parameters A, B, and D by writing equation (6) for several nearby consecutive triplets (x,y,z) and finding the values of A, B, and D that minimize the error in these equations.

5.3 Using Arbitrarily Many Data Points

We wish to estimate the Lagrangian parameters A, B, and D at a pair (x_0, y_0). To do so, we use a collection of n triplets (x_i, y_i, z_i) of consecutive points in our data set, such that the (x_i, y_i) and (y_i, z_i) are near (x_0, y_0). We construct an $n \times 3$ matrix M whose ith row is

$$\begin{pmatrix} 2y_i - (x_0 + y_0) & x_i + z_i - (x_0 + y_0) & \|y_0 - x_0\| \end{pmatrix}$$

In Section 6, we will scale these rows so that points further from (x_0, y_0) are weighted less. It may also be appropriate to scale the rows in order to compensate for a nonuniform distribution of data points. However, for now, we will do without scaling. Observe that equation (6) becomes $M \begin{pmatrix} A \\ B \\ D \end{pmatrix} = 0$. Of course, we are unlikely to be able to find a solution for n equations with 3 unknowns, so we let ϵ denote the error $M \begin{pmatrix} A \\ B \\ D \end{pmatrix} = \epsilon$, and we will minimize $\frac{\|\epsilon\|^2}{A^2+B^2+D^2}$. To do this, we choose $\begin{pmatrix} A \\ B \\ D \end{pmatrix}$ to be an eigenvector corresponding to the least eigenvalue of $M^T M$.

5.4 The Simple Pendulum Revisited

Using our earlier calculations for the pendulum, we see that the values of B/A, and D/A computed from a second order Taylor approximation to the Lagrangian are

$$\frac{B}{A} = -\frac{4 + \tau^2 \cos\left(\frac{x_0+y_0}{2}\right)}{4 - \tau^2 \cos\left(\frac{x_0+y_0}{2}\right)}, \qquad \frac{D}{A} = -\frac{4\tau^2}{\|y_0 - x_0\|} \cdot \frac{\sin\left(\frac{x_0+y_0}{2}\right)}{4 - \tau^2 \cos\left(\frac{x_0+y_0}{2}\right)}. \tag{7}$$

We estimate the parameters A, B, and D from simulated trajectories of a pendulum using the method in Subsection 5.3. In Figure 4, we compare the values of $\frac{B}{A} + 1$ and $\frac{D}{A} \|y_0 - x_0\|$ computed from the simulated trajectories with the values we expect from Equations (7). The computed values track the expected values much more closely than in Figure 1. Moreover, the new parameters respect the 2π-translational symmetry of the pendulum, as desired. Unfortunately, a drawback of using many data points to smooth the values computed from the trajectories is that they are biased towards zero. A method for mitigating this bias is discussed in Section 6.

In Figure 5, we compute the parameters after adding noise to the simulated trajectories. Comparing Figures 5 and 3 and noting the noise added in Figure 5 is a thousand times larger, it is clear that using many data points dramatically reduces the sensitivity of the parameters to noise in the data.

6 Assigning Weights to the Data Points

In Section 5.4, we observed a bias in the values computed from the trajectories. One of the likely causes for this bias is that, when using many data points to estimate the Lagrangian at (x_0, y_0), we are forced to use data points far away from (x_0, y_0), where the Lagrangian may be different. A natural way of mitigating this effect is to assign weights to the data points based on their distance from (x_0, y_0). However, to do this, we must first create a meaningful definition of distance between a data point (x_i, y_i, z_i) and a pair (x_0, y_0).

We first define a notion of distance between two pairs (x_0, y_0) and (x, y). A natural choice is to define the square of the distance to be $\|x - x_0\|^2 + \|y - y_0\|^2$. Unfortunately, with this definition, changing the time step will change the relative distances between pairs that represent the same position-velocity pairs in phase space. To solve this problem, one could instead define the square of the distance to be $\left\|\frac{(x+y)-(x_0+y_0)}{2}\right\|^2 + \left\|\frac{(y-x)-(y_0-x_0)}{\tau}\right\|^2$, where τ is the time step. However, this definition has inconsistent units. To resolve this issue, we need to have some information about how sensitive the Lagrangian is to changes in velocity relative to changes in position. For the purposes of assigning weights, a characteristic time interval of the system τ_s is a good enough estimate of this ratio, though, given more information about the system, it might be good

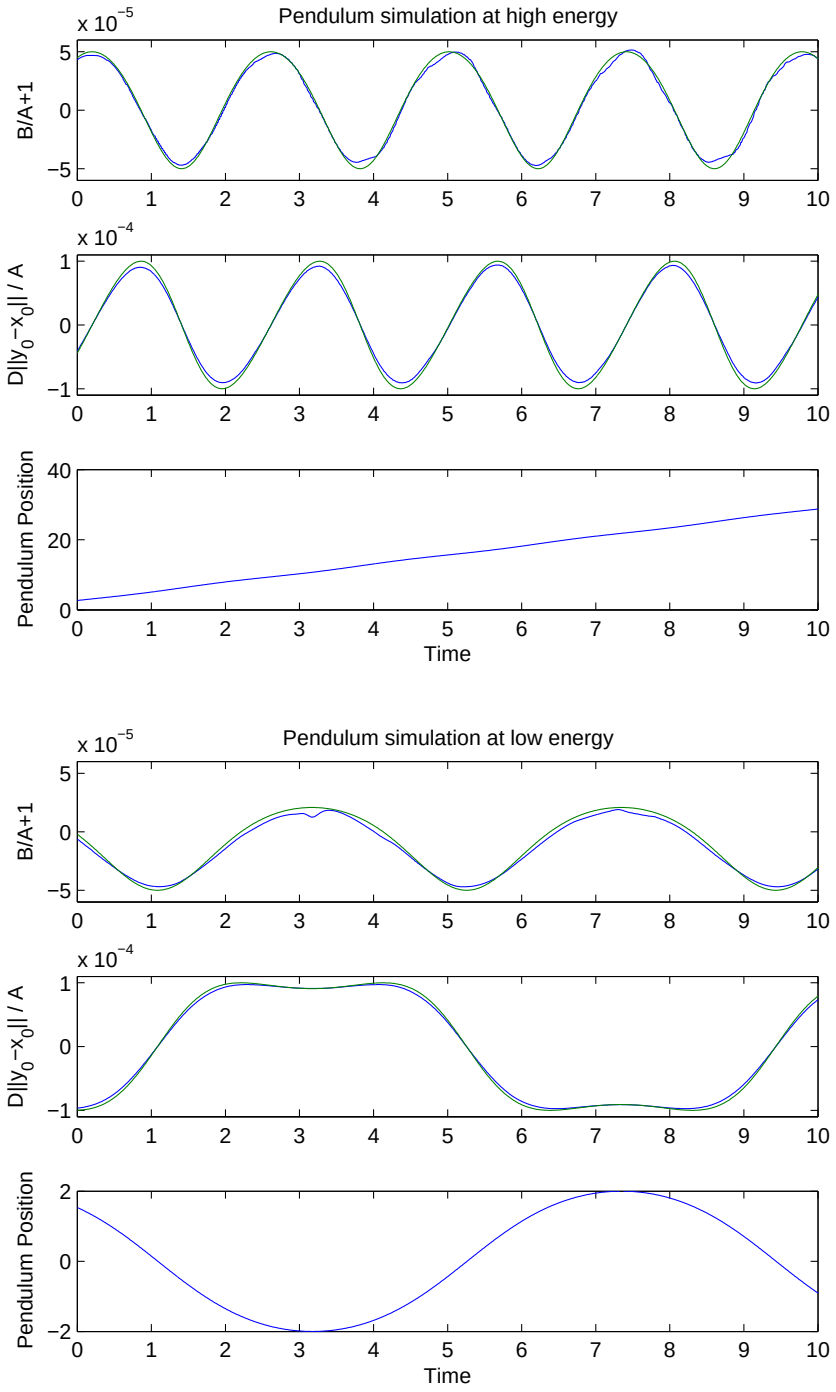


Figure 4: The Lagrangian parameters estimated from the pendulum trajectory at high and low energy using many data points to estimate the values. The values of the parameters that we expect are in green, and the values of the parameters computed from the trajectories are in blue.

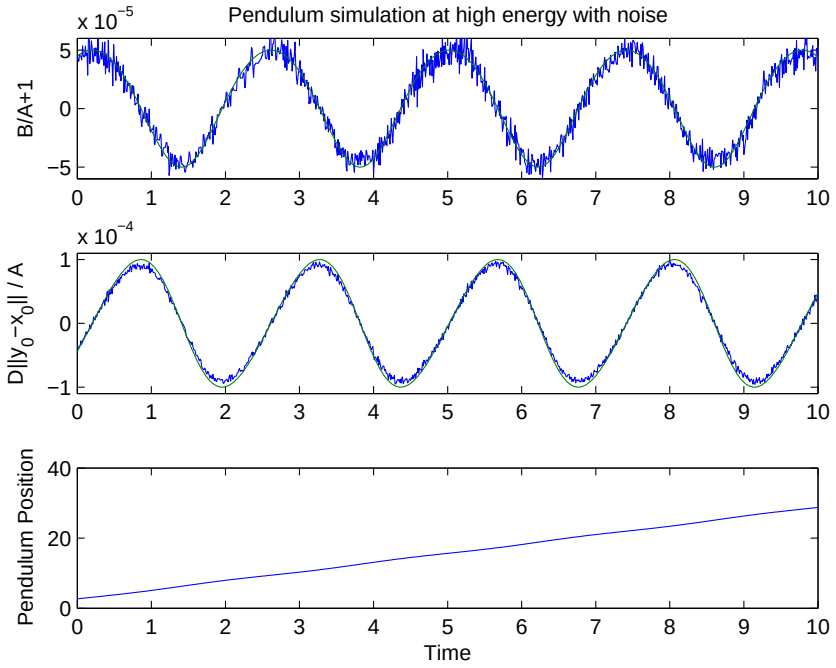


Figure 5: Adding noise with standard deviation $1 \cdot 10^{-4}$ to the pendulum position shows that using many data points to estimate the parameters computed from the trajectory dramatically reduces the sensitivity to noise.

to have τ_s depend on (x_0, y_0). Using this time interval, the distance between pairs of points that we will use is

$$\delta((x_0, y_0), (x, y))^2 = \frac{1}{4} \|(x + y) - (x_0 + y_0)\|^2 + \left(\frac{\tau_s}{\tau}\right)^2 \|(y - x) - (y_0 - x_0)\|^2 . \tag{8}$$

From here, a good notion of the distance between a triple (x_i, y_i, z_i) and a pair (x_0, y_0) is

$$\delta((x_0, y_0), (x_i, y_i, z_i))^2 = \delta((x_0, y_0), (x_i, y_i))^2 + \delta((x_0, y_0), (y_i, z_i))^2. \tag{9}$$

Using this notion of distance, we can modify the matrix used in Section 5.3 by scaling each row of the matrix by a Gaussian as a function of distance to (x_0, y_0). Namely, the ith row of the matrix becomes

$$w_i \begin{pmatrix} 2y_i - (x_0 + y_0) & x_i + z_i - (x_0 + y_0) & \|y_0 - x_0\| \end{pmatrix},$$

where

$$w_i = \exp\left(-\frac{1}{2\sigma^2}\delta((x_0, y_0), (x_i, y_i, z_i))^2\right)$$

and σ is a parameter. For actual computations, it makes sense to set a threshhold and include those data points (x_i, y_i, z_i) with $\exp\left(-\frac{1}{2\sigma^2}\delta((x_0, y_0), (x_i, y_i, z_i))^2\right)$ above the threshhold, and ignore the data points with smaller weights. A smaller σ means that fewer data points are used, which decreases the estimate's bias, whereas a larger σ means that more data points are used, which decreases the estimate's sensitivity to noise.

We applied this method to the simple pendulum, which improved the parameter estimates compared to Section 5.4. In Figure 6, we see that, at small σ, it was possible to significantly reduce the bias in the estimate. At large σ, it was possible to compensate for noise of the same order of magnitude as the difference between consecutive trajectory points, a hundred times larger than that in Figure 5 in Section 5.4. In Figure 7, we show the effect of varying σ on noise rejection and bias. Note that the noise added in Figure 7 is still ten times larger than the noise in Figure 5.

In addition, we discovered that letting the parameter τ_s be equal to the time step τ produced better results than setting τ_s to the characteristic time step of the system. It is possible that, in these cases, τ_s should be interpreted as the ratio between the uncertainty in position and the uncertainty in velocity. If velocity is determined by subtracting two consecutive position measurements and dividing by τ, as it is in our case, then the uncertainty in velocity will be larger than the uncertainty in position by a factor of $\frac{1}{\tau}$.

7 Higher Dimensional Systems

In the above sections, we had largely concerned ourselves only with one-dimensional systems. However, the methods we used for estimating Lagrangian parameters can be generalized to a general n-dimensional system. Like before, we write

$$L(x, y) = \begin{pmatrix} x^T & y^T \end{pmatrix} H(x, y) \begin{pmatrix} x \\ y \end{pmatrix} + J(x, y) \begin{pmatrix} x \\ y \end{pmatrix} + K(x, y).$$

Here, x and y are points in $\mathbb{R}^n$, and H, J, and K are functions from $\mathbb{R}^n \times \mathbb{R}^n$ to quadratic forms in $2n$ variables, linear functions in $2n$ variables, and scalars, respectively. We next transform J and K with respect to a point $p \in \mathbb{R}^n$ to produce J_p and K_p, where

$$\begin{aligned} J_p(x, y) &= 2 \begin{pmatrix} p^T & p^T \end{pmatrix} H(x, y) + J(x, y), \\ K_p(x, y) &= \begin{pmatrix} p^T & p^T \end{pmatrix} H(x, y) \begin{pmatrix} p \\ p \end{pmatrix} + J(x, y) \begin{pmatrix} p \\ p \end{pmatrix} + K(x, y). \end{aligned}$$

As a result, we have

$$L(x, y) = \begin{pmatrix} (x - p)^T & (y - p)^T \end{pmatrix} H(x, y) \begin{pmatrix} x - p \\ y - p \end{pmatrix} + J_p(x, y) \begin{pmatrix} x - p \\ y - p \end{pmatrix} + K_p(x, y).$$

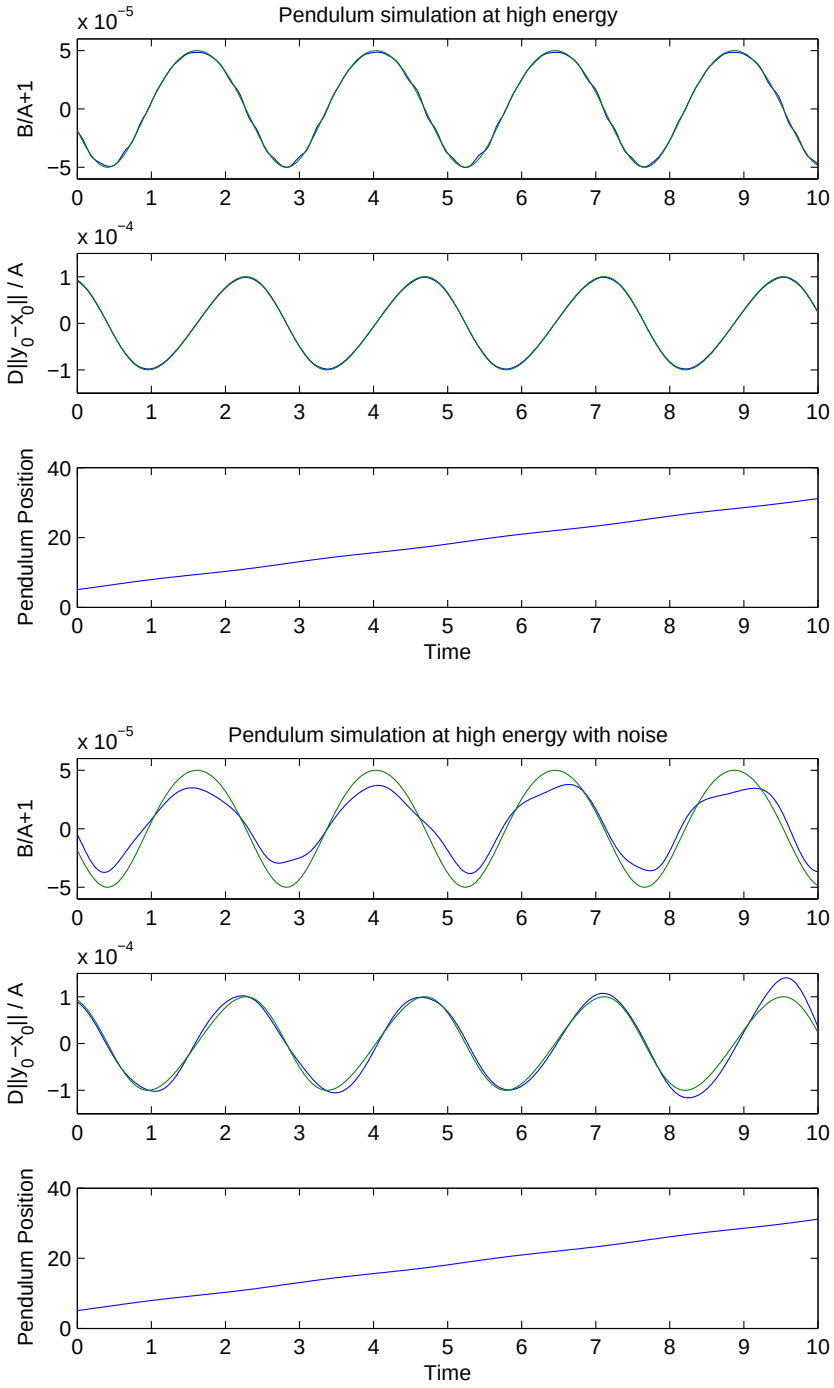
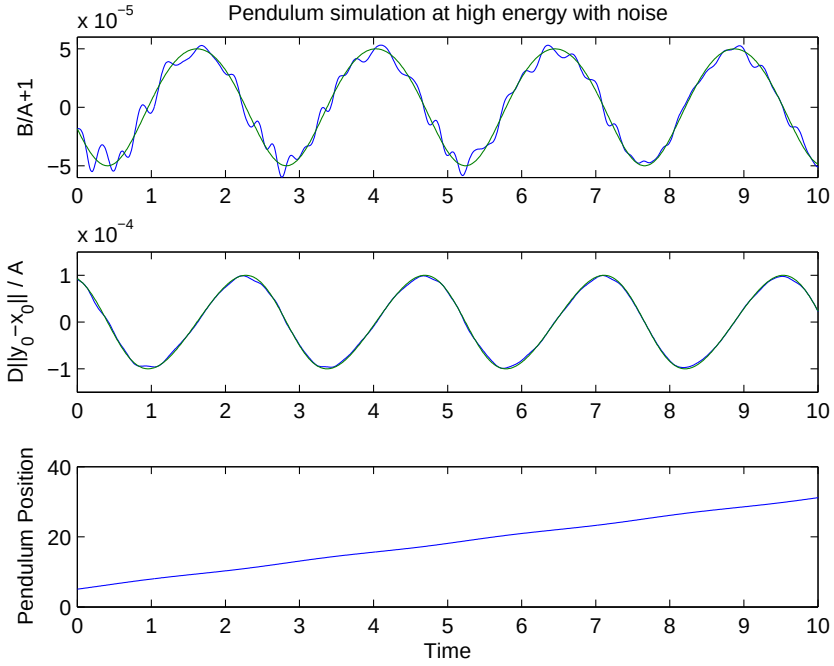


Figure 6: The Lagrangian parameters estimated from the pendulum trajectory at high energy. The values of the parameters that we expect are in green, and the values of the parameters computed from the trajectories are in blue. In the first plot, a small σ results in a near-perfect match between the computed and expected values. In the second plot, we added noise with standard deviation $1 \cdot 10^{-2}$. We increased σ to reject most of this noise at the expense of a larger bias in the estimate.



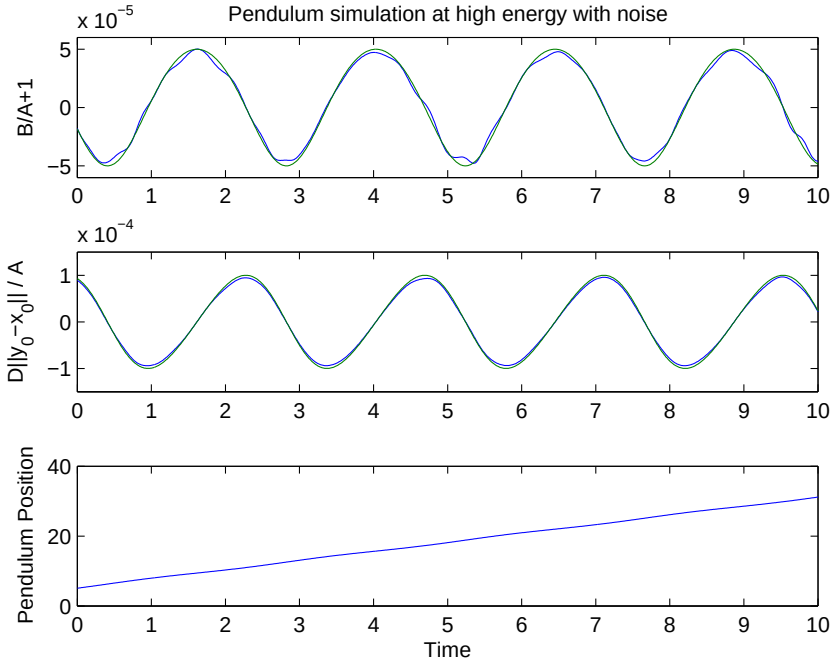


Figure 7: The Lagrangian parameters estimated from the pendulum trajectory with added noise with standard deviation $1 \cdot 10^{-3}$. The values of the parameters that we expect are in green, and the values of the parameters computed from the trajectories are in blue. The two plots are identical except for the choice of σ. In the second plot, the larger value of σ is enough to reject the noise, at the expense of a larger bias in the estimate.

We can then neglect the change in H, J_p, and K_p in order to write the discrete Euler-Lagrange equations for three consecutive points (x, y, z) on a trajectory.

$$\begin{aligned} 0 &= D_1 L(y,z) + D_2 L(x,y) \\ &\approx 2\begin{pmatrix}(y-p)^T & (z-p)^T\end{pmatrix} H \begin{pmatrix} I \\ 0 \end{pmatrix} + J_p \begin{pmatrix} I \\ 0 \end{pmatrix} + 2\begin{pmatrix}(x-p)^T & (y-p)^T\end{pmatrix} H \begin{pmatrix} 0 \\ I \end{pmatrix} + J_p \begin{pmatrix} 0 \\ I \end{pmatrix} \end{aligned}$$

We can write H in block form as $\left(\begin{smallmatrix} a & b \\ b^T & c \end{smallmatrix}\right)$, where a, b, and c are $n \times n$ matrices. Likewise, we can write J_p in block form $\left(\begin{smallmatrix} d_p & e_p \end{smallmatrix}\right)$, where d_p and e_p are row-vectors of length n. The above equation then simplifies to the row-vector equation

$$0 \approx 2(y-p)^T \cdot (a+c) + 2(x-p)^T \cdot b + (z-p)^T \cdot b^T + (d_p + e_p).$$

As before, for estimating the Lagrangian at a point (x_0, y_0), we set $p = \frac{1}{2}(x_0 + y_0)$ and rescale the parameters to produce

$$A(x_0,y_0) := (a+c)\,\|y_0 - x_0\|\,, \qquad B(x_0,y_0) := 2b\,\|y_0 - x_0\|\,, \qquad D(x_0,y_0) := d_p + e_p.$$

We can then rewrite the Euler-Lagrange equations as

$$0 \approx (2y - x_0 - y_0)^T \cdot A + \tfrac{1}{2}\,(2x - x_0 - y_0)^T \cdot B + \tfrac{1}{2}\,(2z - x_0 - y_0)^T \cdot B^T + \|y_0 - x_0\| \cdot D.$$

In coordinates, this equation becomes the equations

$$0 \approx \sum_{i=1}^{n} (2y - x_0 - y_0)_i \cdot A_{ij} + \tfrac{1}{2}\sum_{i=1}^{n} (x - x_0 - y_0)_i \cdot B_{ij} + \tfrac{1}{2}\sum_{i=1}^{n} (z - x_0 - y_0)_i \cdot B_{ji} + \|y_0 - x_0\| \cdot D_j \qquad (10)$$

for $1 \le j \le n$.

Note that since H is a symmetric matrix, both a and c are symmetric, so A is also symmetric. Therefore, we need to estimate a total of $\frac{1}{2}n(n+1) + n^2 + n = \frac{3}{2}n(n+1)$ parameters up to scaling. If we use N data points (x_k, y_k, z_k) to estimate these parameters, we have nN equations. Thus we construct an $nN \times \frac{3}{2}n(n+1)$ matrix M, where the entries are the coefficients of the parameters in the n equations (10) for each data point, multiplied by a weight w_k computed the same way as before with

$$w_k = \exp\left(-\frac{1}{2\sigma^2}\delta((x_0,y_0),(x_k,y_k,z_k))^2\right),$$

where δ is the same distance function as in (8) and (9) in Section 6. As in Section 5.3, we estimate the values of the entries of A, B, and D up to scaling by computing the eigenvector corresponding to the least eigenvalue of the matrix $M^T M$.

8 Future Directions

The method outlined in this paper accurately reconstructed the Lagrangian of the simple pendulum, even when a substantial amount of measurement noise was added to the trajectories, so it is likely to be a promising way for uncovering the Lagrangian of a system from observations. However, there are still several unexplored directions in developing this method further.

One important thing to try is to apply this method to real data. Even after all the improvements to the method, the estimated parameters still depend slightly on the method of integration used to simulate the pendulum. It is possible that some artifacts of the estimated parameters will disappear when real data is used. Conversely, it is possible that real data will present new problems that simulated trajectories with added noise fail to capture.

In this paper, only one trajectory was used to estimate the Lagrangian. However, the method is general enough to be used with data about several trajectories at once, as long as they are near the point of phase space where one wishes to estimate the Lagrangian. Using many trajectories will change the distribution of data points in phase space, which could affect how this method is best applied, particularly with respect to the choice of weights for the data points.

The ultimate goal of recovering the Lagrangian from trajectories of a system is to use the estimated Lagrangian to predict new trajectories of the system. The accuracy of these new trajectories compared to other extrapolation methods would be a good test of the effectiveness of this method.

References

[1] Evan S. Gawlik, Patrick Mullen, Dmitry Pavlov, Jerrold E. Marsden, and Mathieu Desbrun, *Geometric, variational discretization of continuum theories*, 2010.

[2] Dmitry Pavlov, Patrick Mullen, Yiying Tong, Eva Kanso, Jerrold E. Marsden, and Mathieu Desbrun, *Structure-preserving discretization of incompressible fluids*, 2009, to appear in Physica D.

[3] Lawrence K. Saul and Sam T. Roweis, *An introduction to locally linear embedding*, Tech. report, 2000.

[4] Michael Schmidt and Hod Lipson, *Distilling free-form natural laws from experimental data*, Science **324** (2009), no. 5923, 81–85.

[5] Ari Stern and Mathieu Desbrun, *Discrete geometric mechanics for variational time integrators*, Discrete Differential Geometry: An Applied Introduction, 2006, pp. 75–80.