MA 591 - HOMEWORK 01

Hayden Outlaw

NCSU Applied Mathematics, 12 Sep 2025

---

1 Deep ReLU Neural Networks

Let $\text{Id} : \mathbb{R}^n \to \mathbb{R}^n$ be the identity map such that $\text{Id}(\boldsymbol{x}) = \boldsymbol{x}$ for all $\boldsymbol{x} \in \mathbb{R}^n$. In class, we explicitly constructed a two-layer ReLU neural network (NN) of width two that exactly represents Id for a special case of $n = 1$. Specifically, it can be checked that:

$$u_{NN}(\boldsymbol{x}) = W^{(2)}\phi\left(W^{(1)}\boldsymbol{x} + b^{(1)}\right) + b^{(2)} = \boldsymbol{x} = \text{Id}(\boldsymbol{x})$$

where $\phi(x) = \text{ReLU}(x) = \max\{x, 0\}$,

$$W^{(1)} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, b^{(1)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, W^{(2)} = \begin{bmatrix} 1 & -1 \end{bmatrix}, b^{(2)} = 0$$

1.1 Extending Width of Identity Map

Let $\boldsymbol{x} \in \mathbb{R}^n$. We maintain $b^{(1)} = \boldsymbol{0} \in \mathbb{R}^{2n}, b^{(2)} = \boldsymbol{0} \in \mathbb{R}^n$ are vectors of appropriately sized zeros.

Define $W^{(1)} \in \mathbb{R}^{2n \times n}$:

$$W^{(1)} = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ & & & & 1 \\ -1 & & & & \\ & -1 & & & \\ & & \ddots & & \\ & & & -1 & \\ & & & & -1 \end{bmatrix} = \begin{bmatrix} I_n \\ -I_n \end{bmatrix}$$

Then, expressed in block form, $W^{(1)}\boldsymbol{x} + b^{(1)} = \begin{bmatrix} \boldsymbol{x} \\ -\boldsymbol{x} \end{bmatrix}$. If we similarly define, in block form,

$$W^{(2)} = \begin{bmatrix} 1 & & & & & -1 & & & & \\ & 1 & & & & & -1 & & & \\ & & \ddots & & & & & \ddots & & \\ & & & 1 & & & & & -1 & \\ & & & & 1 & & & & & -1 \end{bmatrix} = \begin{bmatrix} I_n & -I_n \end{bmatrix}$$

We can then evaluate the output of the composed weight matrices, using $\phi = \mathrm{ReLU}(x)$ as:

$$
\begin{aligned}
W^{(2)}\phi\left(W^{(1)}\boldsymbol{x} + b^{(1)}\right) + b^{(2)} &= \begin{bmatrix} I_n & -I_n \end{bmatrix} \begin{bmatrix} \phi(\boldsymbol{x}) \\ \phi(-\boldsymbol{x}) \end{bmatrix} \\
&= \phi(\boldsymbol{x}) - \phi(-\boldsymbol{x}) \\
&= \begin{bmatrix} \phi(x_1) - \phi(-x_1) \\ \phi(x_2) - \phi(-x_2) \vdots \\ \phi(x_n) - \phi(-x_n) \end{bmatrix} \\
&= \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \\
&= \boldsymbol{x}
\end{aligned}
$$

which demonstrates that this is a NN of width $2n$ equivalent to $\mathrm{Id}(x)$, demonstrating the identity map is also a learnable transformation for vectors in $\mathbb{R}^n$.

1.2 Extending Depth of Identity Map

Using the definitions in 1.1, we define $u_{Id}(x) = W^{(2)}\phi\left(W^{(1)}\boldsymbol{x} + b^{(1)}\right) + b^{(2)}$, which we know satisfies $u_{Id}(\boldsymbol{x}) = \boldsymbol{x}$, and is a 2-layer ReLU NN of width $2n$ that is equal to Id.
We can generalize from the 2-layer to 3-layer case by composing the network operations:

$$
\begin{aligned}
u_{Id}(u_{Id}(\boldsymbol{x})) &= W^{(2)}\phi\left(W^{(1)}\left[u_{Id}(\boldsymbol{x})\right] + b^{(1)}\right) + b^{(2)} \\
&= W^{(2)}\phi\left(W^{(1)}\left[W^{(2)}\phi\left(W^{(1)}\boldsymbol{x} + b^{(1)}\right) + b^{(2)}\right]\right) \\
&= W^{(2)}\phi\left(W^{(1)}\left[W^{(2)}\phi\left(W^{(1)}\boldsymbol{x}\right)\right]\right)
\end{aligned}
$$

Examining directly:

$$
\begin{aligned}
W^{(1)}W^{(2)} &= \begin{bmatrix} I_n \\ -I_n \end{bmatrix} \begin{bmatrix} I_n & -I_n \end{bmatrix} \\
&= \begin{bmatrix} I_n & -I_n \\ -I_n & I_n \end{bmatrix} \in \mathbb{R}^{2n \times 2n}
\end{aligned}
$$

Given that we can express this term as one matrix, this is a valid 3-layer NN, which we know via composition of linear operators, to be equivalent to $\mathrm{Id}(\mathrm{Id}(\boldsymbol{x})) = \mathrm{Id}(\boldsymbol{x}) = \boldsymbol{x}$. Inductively, if we wanted to extend this to L layers, you could do L repeated compositions of u_{Id} to define a valid L layer network that remains equivalent to Id.

1.3 Depth of Composed Networks

Let $u_{NN} : \mathbb{R}^n \to \mathbb{R}^m$ and $v_{NN} : \mathbb{R}^m \to \mathbb{R}^r$ be ReLU NNs of depth L_1 and L_2 respectively. Using the same mechanism illustrated in 1.2, we can show $v_{NN} \circ u_{NN}$ is a ReLU network of depth $L_1 + L_2 - 1$.

$v_{NN} \circ u_{NN}$ is a valid composition, as the output space of u_{NN} is the same as the input space of v_{NN}. By passing the output from u_{NN} to v_{NN}, it trivially defines a new ReLU NN - but what is the total depth of the composed network? If we define weight and bias terms of u_{NN} as $W_u^{(i)}, b_u^{(i)}$ at layer i, and the weight and bias terms of v_{NN} as $W_v^{(i)}, b_v^{(i)}$ at layer i, we have:

$$
\begin{aligned}
v_{NN}(u_{NN}(\boldsymbol{x})) &= v_{NN}\left(W_u^{(L_1)}\phi(W_u^{(L_1-1)}[\ldots] + b_u^{(L_1-1)}) + b_u^{(L_1)}\right) \\
&= W_v^{(L_2)}\phi\left(W_v^{(L_2-1)}\left(\ldots\ldots W_v^{(1)}\phi\left(W_u^{(L_1)}\phi(W_u^{(L_1-1)}[\ldots] + b_u^{(L_1-1)}) + b_u^{(L_1)}\right) + b_v^{(1)}\right)\ldots\cdots + b_v^{(L_2-1)}\right) + b_v^{(L_2)}
\end{aligned}
$$

The outer L_2 layers from v_{NN} and the inner L_1 layers for u_{NN} are directly identifiable. However, we can reduce the total depth by 1 by distributing the innermost operation of v_{NN} to the outermost operation of u_{NN}.
More specifically:

$$
W_v^{(1)}\phi\left(W_u^{(L_1)}\boldsymbol{x} + b_u^{(L_1)}\right) + b_v^{(1)} = W_v^{(1)}\phi\left(W_u^{(L_1)}\boldsymbol{x}\right) + \phi\left(b_u^{(L_1)}\right) + b_v^{(1)}
$$

This is really one layer, with bias operation $\phi\left(b_u^{(L_1)}\right) + b_v^{(1)}$ and weight operation $W_v^{(1)}\phi\left(W_u^{(L_1)}\boldsymbol{x}\right)$, because ϕ as an activation function is applied element-wise. Therefore, the true number of layers in the network is $L_1 + L_2 - 1$ once accounting for compacting the layer where the two networks are actually composed.

1.4 Reparameterizing ReLU Networks

Let $u_{NN} : \mathbb{R}^n \to \mathbb{R}$ be a L-layer ReLU network of at least width $2n$. We can show that there exists a deeper ReLU network v_{NN} of width L that is exact to u_{NN}.

We know this, as we parameterized the ReLU network u_{Id} of depth L in 1.1, and showed it can be equivalently of any length ≥ 2 in 1.2, guaranteeing the existence of ReLU NN $u_{Id} : \mathbb{R}^n \to \mathbb{R}^n$ such that $u_{Id}(\boldsymbol{x}) = \boldsymbol{x}$. To create a longer ReLU NN that is exact to u_{NN}, we can define v_{NN} to be $v_{NN} := u_{Id} \circ u_{NN}$.

We know $v_{NN}(\boldsymbol{x}) = u_{NN}(u_{Id}(\boldsymbol{x})) = u_{NN}(\boldsymbol{x}) \; \forall \; \boldsymbol{x} \in \mathbb{R}^n$, so $v_{NN} \equiv u_{NN}$. As shown in 1.3, if u_{Id} is of length $L \geq 2$, the total length of v_{NN} is $L + K - 1 \geq L + 2 - 1 > L$, so v_{NN} is of greater depth than u_{NN}.

2 Structure-preserving Numerical Integration

Consider a dynamical system that describes the movement of a simple pendulum. It is well known that the differential equation that governs the motion of a simple pendulum is given by:

$$\frac{d^2\theta}{dt} + \frac{g}{l}\sin\theta = 0$$

where g is the magnitude of the gravitational field, l is the length of the rod, and θ is the angle from the vertical axis to the pendulum. This description can be written as a Hamiltonian system.

Without loss of generality, let $l = 1, g = 1, m = 1$. Define $q := \frac{\theta}{l}$ be the generalized coordinate with respect to the angle of the system, and consider $R = \begin{bmatrix}\sin(q) & \cos(q)\end{bmatrix}^\top$ be the position vector of the pendulum. Since the total energy is the sum of the kinetic and potential energies, we have:

$$E = \frac{1}{2}\dot{q}^2 - g\cos(q)$$

The Hamiltonian for a conservative system is the total energy with $p = m\dot{q}$, which yields:

$$H = \frac{1}{2}p^2 - \cos(q)$$

The Hamiltonian dynamics are then described by:

$$\begin{cases} \dot{p} = -\frac{\partial}{\partial q}H(p,q) \\ \dot{q} = \frac{\partial}{\partial p}H(p,q) \end{cases}$$

2.1 Concise Hamiltonian Formulation

Let $x = \begin{bmatrix}p & q\end{bmatrix}^\top$. We can describe the matrix differential equation as:

$$\begin{aligned}
\dot{x} &= \begin{bmatrix}\dot{p} \\ \dot{q}\end{bmatrix} \\
&= \begin{bmatrix}-\frac{\partial}{\partial q}H(x) \\ \frac{\partial}{\partial p}H(x)\end{bmatrix} \\
&= \begin{bmatrix}0 & -1 \\ 1 & 0\end{bmatrix}\begin{bmatrix}\frac{\partial H(x)}{\partial p} & \frac{\partial H(x)}{\partial q}\end{bmatrix} \\
&= -\begin{bmatrix}0 & 1 \\ -1 & 0\end{bmatrix}\nabla H(x)
\end{aligned}$$

We can additionally show H is conserved in time, i.e., $\frac{d}{dt}H = 0$:

$$\begin{aligned}
\frac{d}{dt}H &= \frac{d}{dt}\left(\frac{1}{2}p^2\right) - \frac{d}{dt}\cos(q) \\
&= p\frac{dp}{dt} + \sin(q)\frac{dq}{dt} \\
&= m\dot{q}\frac{d}{dt}(m\dot{q}) + \sin(q)\dot{q} \qquad \text{as } (p = m\dot{q},\ m \text{ constant}) \\
&= m^2\dot{q}\ddot{q} + \sin(q)\dot{q} \\
&= m^2\left(\frac{\dot{\theta}}{l}\right)\left(\frac{\ddot{\theta}}{l}\right) + \sin\left(\frac{\theta}{l}\right)\left(\frac{\dot{\theta}}{l}\right) \qquad \text{as } (q = \theta/l,\ l \text{ constant}) \\
&= \dot{\theta}(\ddot{\theta} + \sin(\theta)) \qquad \text{as } (m = l = 1) \\
&= \dot{\theta}0 \qquad \text{(by governing equation)} \\
&= 0
\end{aligned}$$

2.2 Symplectic Jacobian of Evolution Operator

Let $\phi_t(x_0) = x(t), \phi_0(x_0) = x_0$ define trajectories of evolution of the system. We can then express:

$$\frac{d}{dt}x(t) = -J\nabla H(x(t)) \iff \frac{d}{dt}\phi_t(x_0) = -J\nabla H(\phi_t(x_0))$$

The objective is to show that $\Phi(t) := \frac{\partial \phi_t(x_0)}{\partial x_0}$ is a solution to $\dot{\Phi} = J^{-1}\nabla^2 H(\phi_t(x_0))\Phi$, i.e., Φ is the evolution operator for perturbations of the initial system.
We perturb the evolution operator by the initial condition:

$$\begin{aligned}
\frac{\partial}{\partial x_0}\frac{d}{dt}\phi_t(x_0) &= \frac{\partial}{\partial x_0}\left(-J\nabla H(\phi_t(x_0))\right) \\
\frac{d}{dt}\frac{\partial}{\partial x_0}\phi_t(x_0) &= \left(-J\nabla^2 H(\phi_t(x_0))\right)\frac{\partial \phi_t(x_0)}{\partial x_0} \qquad (J \text{ is constant wrt } x_0) \\
\dot{\Phi} &= \left(J^{-1}\nabla^2 H(\phi_t(x_0))\right)\Phi
\end{aligned}$$

Now, consider:

$$\begin{aligned}
\frac{d}{dt}\left((\nabla\phi_t)^\top J\nabla\phi_t\right) &= \left(\frac{d}{dt}(\nabla\phi_t)^\top J\nabla\phi_t\right) + \left((\nabla\phi_t)^\top J\frac{d}{dt}\nabla\phi_t\right) \\
&= \left(\left(\left(J^{-1}\nabla^2 H(\phi_t(x_0))\right)\Phi\right)^\top J\nabla\phi_t\right) + \left((\nabla\phi_t)^\top J\left(\left(J^{-1}\nabla^2 H(\phi_t(x_0))\right)\Phi\right)\right) \\
&= \left(\Phi^\top\nabla^2 H(\phi_t(x_0))^\top J^{-\top}J\nabla\phi_t\right) + \left((\nabla\phi_t)^\top JJ^{-1}\nabla^2 H(\phi_t(x_0))\Phi\right) \\
&= -\left(\Phi^\top\nabla^2 H(\phi_t(x_0))^\top\nabla\phi_t\right) + \left((\nabla\phi_t)^\top\nabla^2 H(\phi_t(x_0))\Phi\right) \qquad (\text{as } J^\top = -J, J^{-\top}J = -I) \\
&= -\left((\nabla\phi_t)^\top\nabla^2 H(\phi_t(x_0))^\top\nabla\phi_t\right) + \left((\nabla\phi_t)^\top\nabla^2 H(\phi_t(x_0))\nabla\phi_t\right) \quad (\text{by definition of evolution of perturbations, } Phi = \nabla \\
&= 0
\end{aligned}$$

If this quantity is conserved along trajectories, then $(\nabla\phi_t)^\top J\nabla\phi_t = (\nabla\phi_0)^\top J\nabla\phi_0$ for all t. Moreover, by the definition of the trajectory operator ϕ_0, we must have $\phi_0(x_0) = x_0$ (no trajectory at initial condition). We then have $(\nabla\phi_0)^\top J\nabla\phi_0 = x_0^\top Jx_0 = J$, and can conclude that:

$$(\nabla\phi_t)^\top J\nabla\phi_t = J\ \forall\ t \implies J \text{ is symplectic}$$

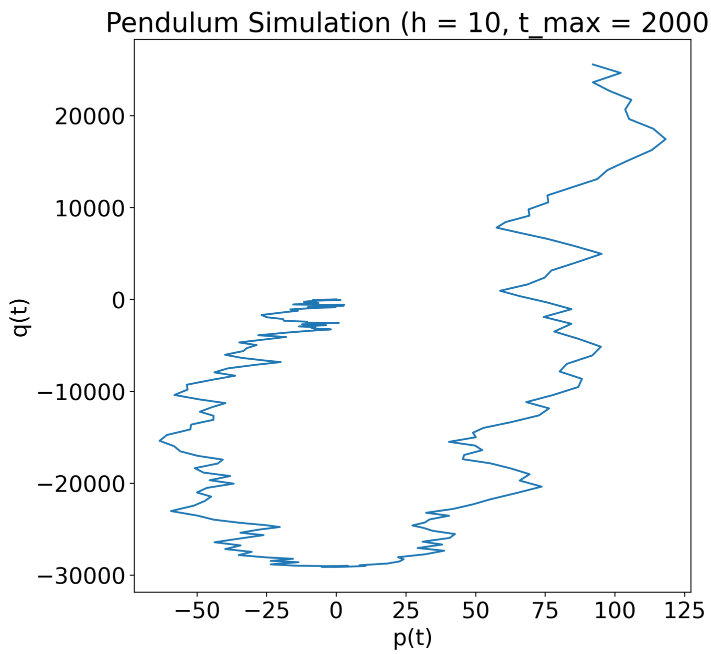
2.3 Semi-Implicit Euler Numerical Solution

Source Code Available At: `https://github.com/outlawhayden/ncsu-sciml/tree/main/hw1`

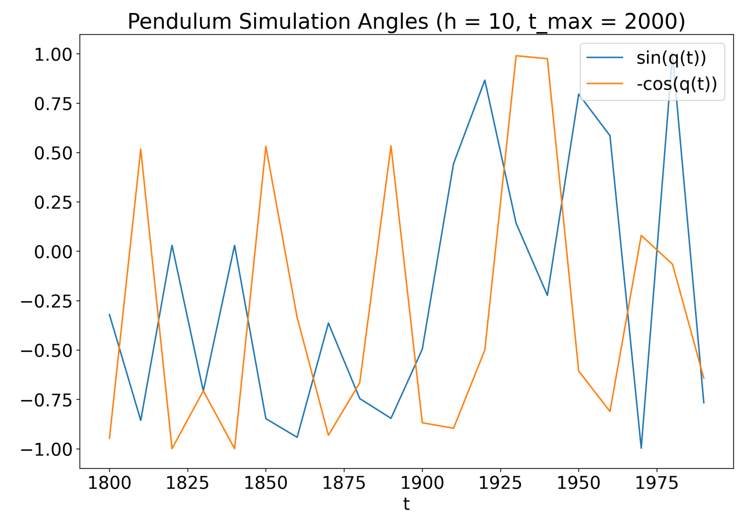
Given that $\dot{x} = \begin{bmatrix} \dot{p} \\ \dot{q} \end{bmatrix} = -J\nabla H = \begin{bmatrix} -\sin(q) \\ p \end{bmatrix}$, we can use the semi-implicit (or symplectic) Euler's method[1] to approximate the solution to the IVP in time:

$$\begin{aligned} x_{i+1} &= \begin{bmatrix} p_{i+1} \\ q_{i+1} \end{bmatrix} \\ &\approx \begin{bmatrix} p_i \\ q_i \end{bmatrix} + h \begin{bmatrix} -\sin(q_i) \\ p_{i+1} \end{bmatrix} \end{aligned}$$

where h is a fixed step forwards in time. Obviously, this is only a numerical simulation of the solution, and not the exact integration. How does the fidelity h affect the accuracy of the output?



(a) $p(t)$ vs $q(t), h = 10$

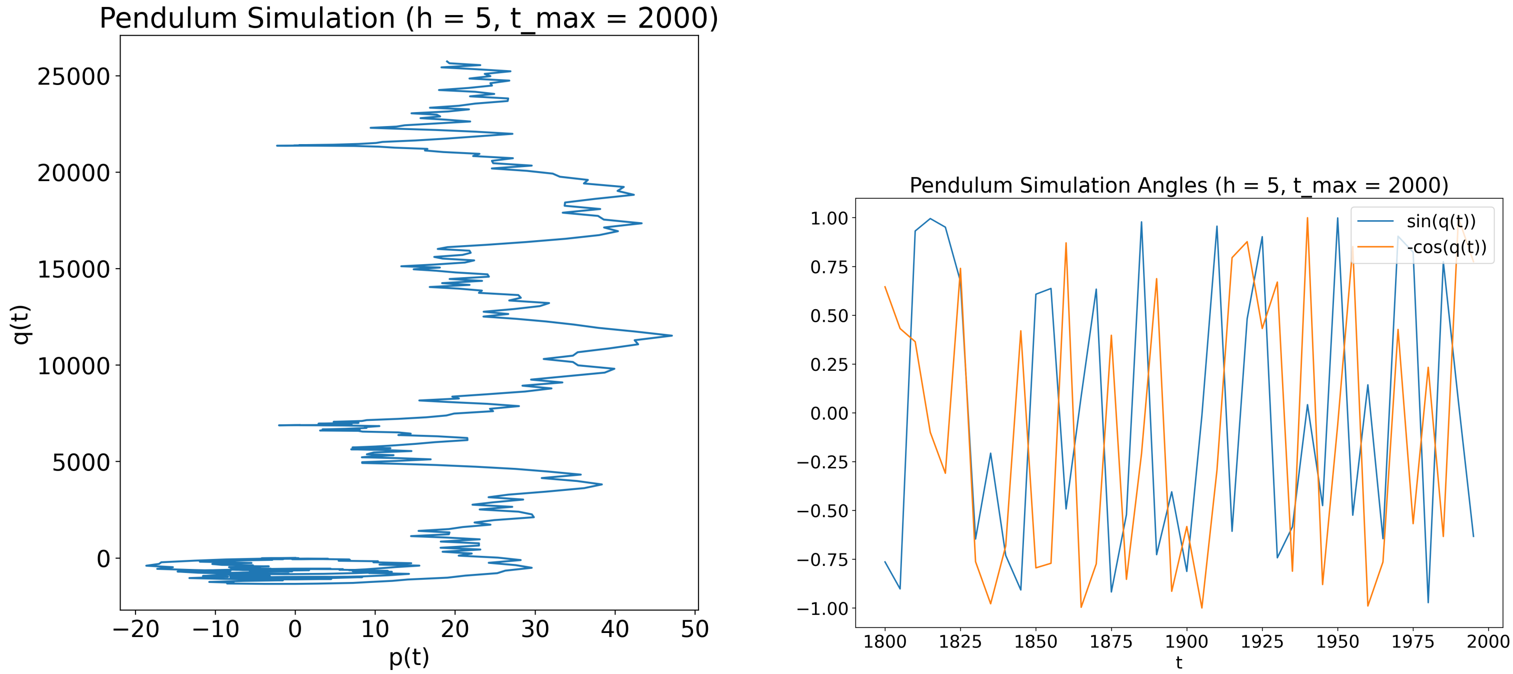


(b) Evolution of $\sin(q(t))$ and $-\cos(q(t))$, $h = 10$

Figure 1: Pendulum Simulation, $h = 10$.

Even though analytically we know the solution to be perfectly periodic, when the fidelity is $h = 10$, the system is clearly nonperiodic. Why? The pendulum system is well known to be sensitive to perturbation, and chaotic as a result. The numerical errors introduced between the symplectic Euler's method and the true integral value thus greatly reduce the accuracy of the system and degrade the final simulation as it evolves in time, as depicted in Figure 1.

[1] `https://en.wikipedia.org/wiki/Semi-implicit_Euler_method`

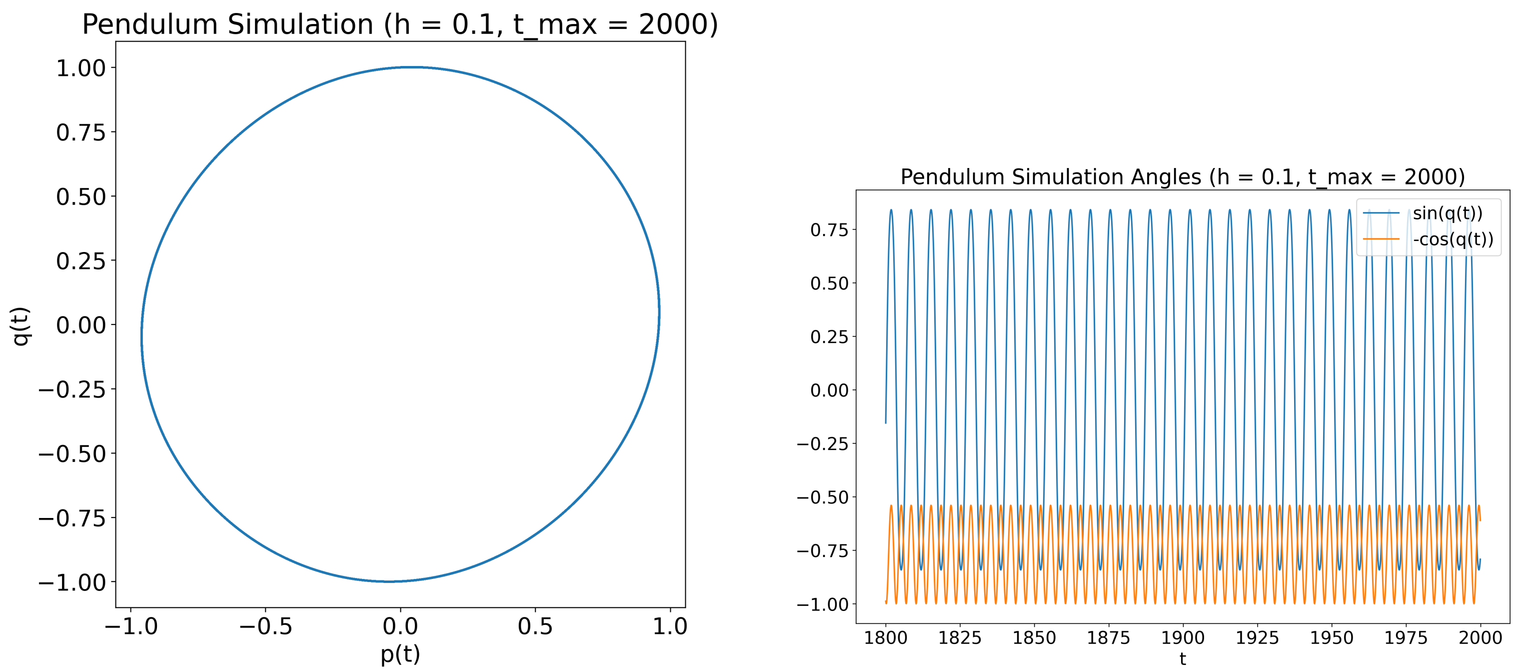


(a) $p(t)$ vs $q(t), h = 5$

(b) Evolution of $\sin(q(t))$ and $-\cos(q(t))$, $h = 5$

Figure 2: Pendulum Simulation, $h = 5$.

If we decrease the step size by a factor of 2 from $h = 10$ to $h = 5$, does the performance improve? As shown in Figure 2, the system is still ultimately chaotic and unstable, but it does perform better than the $h = 10$ case. Note the clustering of values in the bottom left of Figure 2a, before the system ultimately degrades.



(a) $p(t)$ vs $q(t), h = 0.1$

(b) Evolution of $\sin(q(t))$ and $-\cos(q(t))$, $h = 0.1$

Figure 3: Pendulum Simulation, $h = 0.1$.

However, as depicted in Figure 3, if we set $h = 0.1$ sufficiently small, we get exact periodic behavior. This is because this value of h, combined with the eigenvalues of the system for these parameters, lie within the absolute stability region for the method.[2]

[2]Region of A-stability Determined By Niiranen, 1999