Simulating the double pendulum using the Runge-Kutta method and the Adams-Bashforth method.

Mikael Nuutila

May 28, 2020

1 The double pendulum

1.1 Double pendulum

The double pendulum is a system of two pendulums, each consisting of a massless rigid rod, and a bob hanging from the rod. The first pendulum is fixed to a pivot, and the second pendulum is fixed to the bob of the first one. The system is relatively simple, and completely deterministic, but still it behaves in an erratic manner. This behaviour is an example of a phenomenon called chaos.

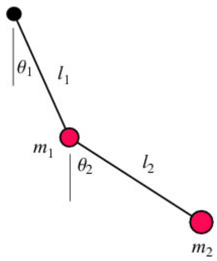


Figure 1: The double pendulum

1.2 Coordinates and equations of motion

The following derivations are from [1]. The double pendulum consists of two pendulums, the one fixed to the pivot has a length l_1 and it's bob has a weight m_1. The second pendulum has a length l_2 and mass m_2. The angles of the rods of pendulum 1 and 2 from vertical are θ_1 and θ_2 respectively. If the pivot point is at the origin, and the position of the first bob is (x_1, y_1), and the position of the second bob is (x_2, y_2), the euclidean coordinates can be expressed as functions of θ_1 and θ_2 (see figure 1):

$$
\begin{aligned}
x_1 &= l_1 \sin(\theta_1) \\
y_1 &= -l_1 \cos(\theta_1) \\
x_2 &= l_1 \sin(\theta_1) + l_2 \sin(\theta_2) \\
y_2 &= -l_1 \cos(\theta_1) - l_2 \cos(\theta_2)
\end{aligned}
$$

Derivating these by time we get

$$
\begin{aligned}
\dot{x}_1 &= l_1\dot{\theta}_1 \cos\theta_1 \\
\dot{y}_1 &= l_1\dot{\theta}_1 \sin\theta_1 \\
\dot{x}_2 &= l_1\dot{\theta}_1 \cos\theta_1 + l_2\dot{\theta}_2 \cos\theta_2 \\
\dot{y}_2 &= l_1\dot{\theta}_1 \sin\theta_1 + l_2\dot{\theta}_2 \sin\theta_2.
\end{aligned}
$$

The potential energy $U = m_1 g y_1 + m_2 g y_2$ can then be written in the form $U = -(m_1+m_2)l_1 g\cos(\theta_1) - m_2 l_2 \cos\theta_2$. The kinetic energy is as follows:

$$
\begin{aligned}
T &= \frac{1}{2}m_1 v_1^2 + \frac{1}{2}m_2 v_2^2 = \frac{1}{2}m_1(\dot{x}_1+\dot{y}_1)^2 + \frac{1}{2}m_2(\dot{x}_2+\dot{y}_2)^2 \\
&= \frac{1}{2}m_1 l_1^2\dot{\theta}_1^2 + \frac{1}{2}m_2(l_1^2\dot{\theta}_1^2 + l_2^2\dot{\theta}_2^2 + 2l_1 l_2\dot{\theta}_1\dot{\theta}_2\cos(\theta_1-\theta_2)).
\end{aligned}
$$

The Lagrangian of the system is then

$$
\begin{aligned}
\mathcal{L} = T - U &= \frac{1}{2}(m_1+m_2)l_1^2\dot{\theta}_1^2 + \frac{1}{2}m_2 l_2^2\dot{\theta}_2^2 + m_1 l_1 l_2\dot{\theta}_1\dot{\theta}_2\cos(\theta_1-\theta_2) \\
&\quad + (m_1+m_2)l_1 g\cos(\theta_1) + m_2 g l_2\cos(\theta_2).
\end{aligned}
$$

This can be used to get the Euler-Lagrangian equations $\frac{d}{dt}\left(\frac{\partial\mathcal{L}}{\partial\dot{\theta}_i}\right) - \frac{\partial\mathcal{L}}{\partial\theta_i} = 0$. These get the form

$$
\begin{aligned}
(m_1+m_2)l_1\ddot{\theta}_1 + m_2 l_2\ddot{\theta}_2\cos(\theta_1-\theta_2) + m_2 l_2\dot{\theta}_2^2\sin(\theta_1-\theta_2) + g(m_1+m_2)\sin(\theta_1) &= 0 \\
m_2 l_2\ddot{\theta}_2 + m_2 l_1\ddot{\theta}_1\cos(\theta_1-\theta_2) - m_2 l_1\dot{\theta}_1^2\sin(\theta_1-\theta_2) + m_2 g\sin(\theta_2) &= 0.
\end{aligned}
$$

According to [2] if these are solved for $\ddot{\theta}_1$ and $\ddot{\theta}_2$, and we substitute $z_1 = \dot{\theta}_1$ and $z_2 = \dot{\theta}_2$, we get the following system of equations:

$$
\begin{cases}
\dot{\theta}_1 = z_1 \\
\dot{\theta}_2 = z_2 \\
\dot{z}_1 = \frac{m_2 g\sin(\theta_2)\cos(\theta_1-\theta_2) - m_2 g\sin(\theta_1-\theta_2)(l_1 z_1^2\cos(\theta_1-\theta_2) + l_2 z_2^2) - (m_1+m_2)g\sin(\theta_1)}{l_1(m_1+m_2\sin(\theta_1-\theta_2)^2)} \\
\dot{z}_2 = \frac{(m_1+m_2)(l_1 z_1^2\sin(\theta_1-\theta_2) - g\sin(\theta_2) + g\sin(\theta_1)\cos(\theta_1-\theta_2)) + m_2 l_2 z_2^2\sin(\theta_1-\theta_2)\cos(\theta_1-\theta_2)}{l_2(m_1+m_2\sin(\theta_1-\theta_2)^2)}
\end{cases}
\tag{1}
$$

If we write $\mathbf{Q} = (\theta_1, \theta_2, z_1, z_2)^T$, this system of equations can be written in the form

$$\dot{\mathbf{Q}} = \mathbf{F}(\mathbf{Q}), \tag{2}$$

where $\mathbf{F} : \mathbb{R}^4 \to \mathbb{R}^4$ is the function specified in (1).

2 The numerical methods

2.1 The Runge-Kutta method

Assume that we have to solve the initial value problem $\frac{dy}{dt} = f(t, y),\ y(t_0) = y_0$. The approximation $\frac{dy(t_i)}{dt} = \frac{y_{i+1}-y_i}{\Delta t} = f(t_i, y_i)$, where $t_{i+1} = t_i + \Delta t$ leads to Euler's method $y_{i+1} = y_i + \Delta t f(t_i, y_i)$. This however is fairly inaccurate.

We can use slopes k_1, k_2, k_3 and k_4 at intermediate steps to reach a better approximation. The slopes are defined in the following way:

$$\begin{aligned}
k_1 &= f(t_i, y_i) \\
k_2 &= f(t_i + \frac{\Delta t}{2}, y_i + \Delta t \frac{k_1}{2}) \\
k_3 &= f(t_i + \frac{\Delta t}{2}, y_i + \Delta t \frac{k_2}{2}) \\
k_4 &= f(t_i + \Delta t, y_i + \Delta t k_3).
\end{aligned}$$

Now the next value for y is $y_{i+1} = y_i + \Delta t \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$. This method for updating the values y_i is called the *4th order Runge-Kutta method*. It is called *4th order* because the total accumulated error is of order $\mathcal{O}(\Delta t^4)$.

2.2 The Adams-Bashforth method

The Runge-Kutta method is more accurate than Euler's method, but it has a problem; four intermediate steps are calculated at each time-step, and these are not reused. This is computationally inefficient.

The *4th order Adams-Bashforth method* solves this weakness. New values for y_i are calculated by using the derivative at 4 previous time-steps. These

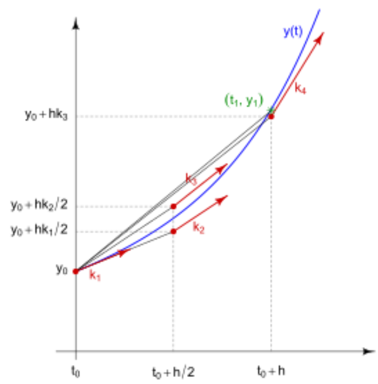


Figure 2: Illustration of the intermediate steps of the Runge-Kutta method.

derivatives can be stored, so at each time-step only one derivative needs to be calculated. The method is the following:

$$y_{i+1} = y_i + \Delta t \left(\frac{55}{24} f(t_i, y_i) - \frac{59}{24} f(t_{i-1}, y_{i-1}) + \frac{37}{24} f(t_{i-2}, y_{i-2}) - \frac{9}{24} f(t_{i-3}, y_{i-3}) \right).$$

The drawback is that now the initial condition y_0 is not enough, but values y_1, y_2, y_3 are needed. If these are not given, they have to be approximated for example by using Runge-Kutta method. This introduces error.

3 Simulating the double pendulum

3.1 Implementation

The equation (2) is of the form required in both the Runge-Kutta method and the Adams-Bashforth method. The python file `doublePendulum.py` contains method for simulating a double pendulum by using either the Runge-Kutta method or the Adams-Bashforth method.

Few key parameters are needed: `Q0,M,L,t` and `dt`. `Q0` is an `np.array` that contains the initial condition $(\theta_1(0), \theta_2(0), z_1(0), z_2(0))$. `M` and `L` are `np.arrays` containing (m_1, m_2) and (l_1, l_2) respectively. The time step Δt is stored in `dt`. The time interval is partitioned to points Δt apart from each other, these points are stored in `t`.

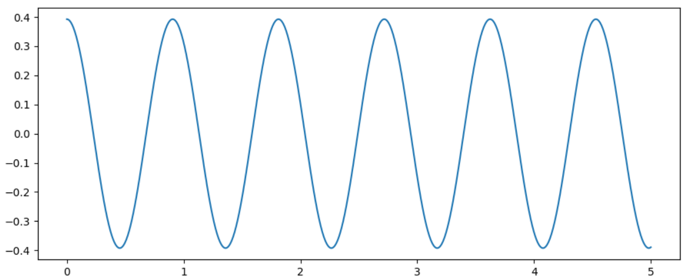


Figure 3: Graph of $\theta_1(t)$ of a single pendulum.
Initial angle $\theta_1(0) = \pi/8$.

The simulation can be started by calling `simulate_double_pendulum(Q0,M,L,t,dt,method)`, where `method` is a string specifying the method by which the simulation will be done, either `"RK4"` or `"AB4"`. The method call returns an `np.array` whose i:th column contains the state $\mathbf{Q}(t_i)$. These arrays are called *trajectories*.

After the simulation is done, and the trajectory has been saved, the pendulum can be animated by calling the method `animate_double_pendulum(list_of_trajectories,L,dt,fps)`. Here `list_of_trajectories` is a list containing trajectories. `fps` is a float specifying the framerate of the animation (with respect to `t`).

3.2 Results

In order to evaluate whether the simulation works, a single pendulum is tested. This can be done by setting the mass of the second pendulum to 0. The initial angle for θ_1 is set to $\pi/8$, small enough that the non-linearity of the system is negligible. This should result in a sinusoidal graph of $\theta_1(t)$. The graph is illustrated in figure 3. The graph looks sinusoidal, which is to be expected.

Another sanity check is to check whether the total energy of the system is conserved. Figure 4 shows that it is.

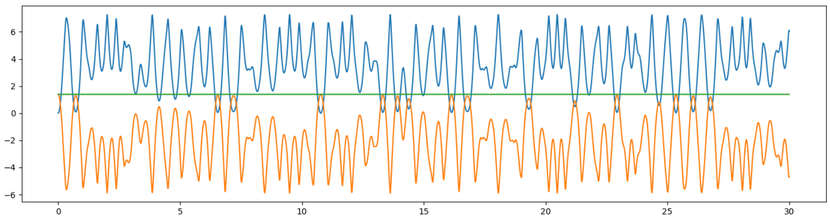


Figure 4: Plot of energies of a double pendulum. Both bobs have the same mass. Green is the total energy, blue is the kinetic energy, and red is the potential energy. Initial condition $\mathbf{Q}(0) = (\pi/2, 3\pi/4, 0, 0)$.

3.3 Differences between the Runge-Kutta and Adams-Bashforth methods

As expected the Adams-Bashforth method is significantly faster than the Runge-Kutta method. Starting from the initial condition $\mathbf{Q}(0) = (\pi/2, 3\pi/4, 0, 0)$ simulations of $10, 20, 30, 40, 50$ and 60 seconds are conducted. All simulations have $\Delta t = 0.0001$. The figure 5 shows the difference between the simulation times of both methods. It turns out that the Adams-Bashforth method is 3.048 times faster than the Runge-Kutta method.

Another difference between the methods is that when Δt is high, for example 0.01, the Adams-Bashforth method seems to result in fluctuating and diminishing energies. This is illustrated in figure 6.

3.4 Chaos and sensitivity to initial conditions

The double pendulum is chaotic in the sense that it is sensitive to changes in the initial condition. This means that if two initial conditions are different, the difference between the states starting from these initial conditions always gets larger than a specified number. In other words no matter how close the initial conditions are, given enough time the states will evolve into two completely different ones.

Sensitivity to initial conditions can be visualized by altering the initial

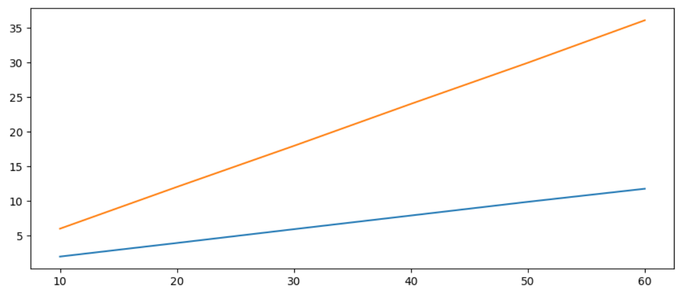
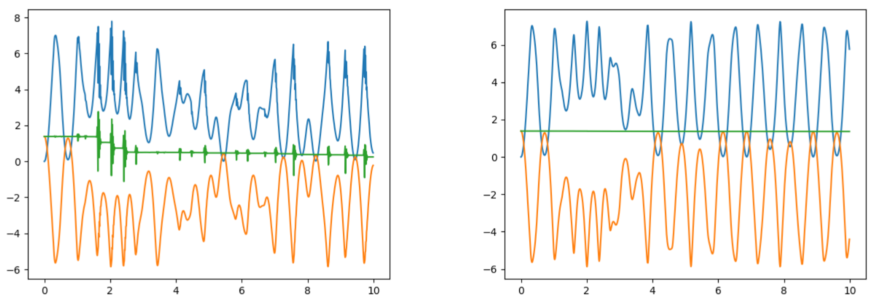


Figure 5: The computing time of both methods as a function of the length of the time interval. Blue graph is Adams-Bashforth, and red is Runge-Kutta.



(a) The Adams-Bashforth method

(b) The Runge-Kutta method

Figure 6: Energies when $\Delta t = 0.01$.

conditions and animating the pendulums, or by plotting different parameters of the systems. First start from state $Q(0) = (\pi/2, 3\pi/4, 0, 0)$. Then vary θ_1 and θ_2 by multiples of 0.00001, so that both angles get 11 different values. This is illustrated in figures 7 and 8.

Another way of producing different initial condition is to vary θ_1 by multiples of 0.00001 and z_1 by multiples of 0.04. The pairs (θ_1, z_1) can then be plotted, the resulsting images are illustrated in figure 9

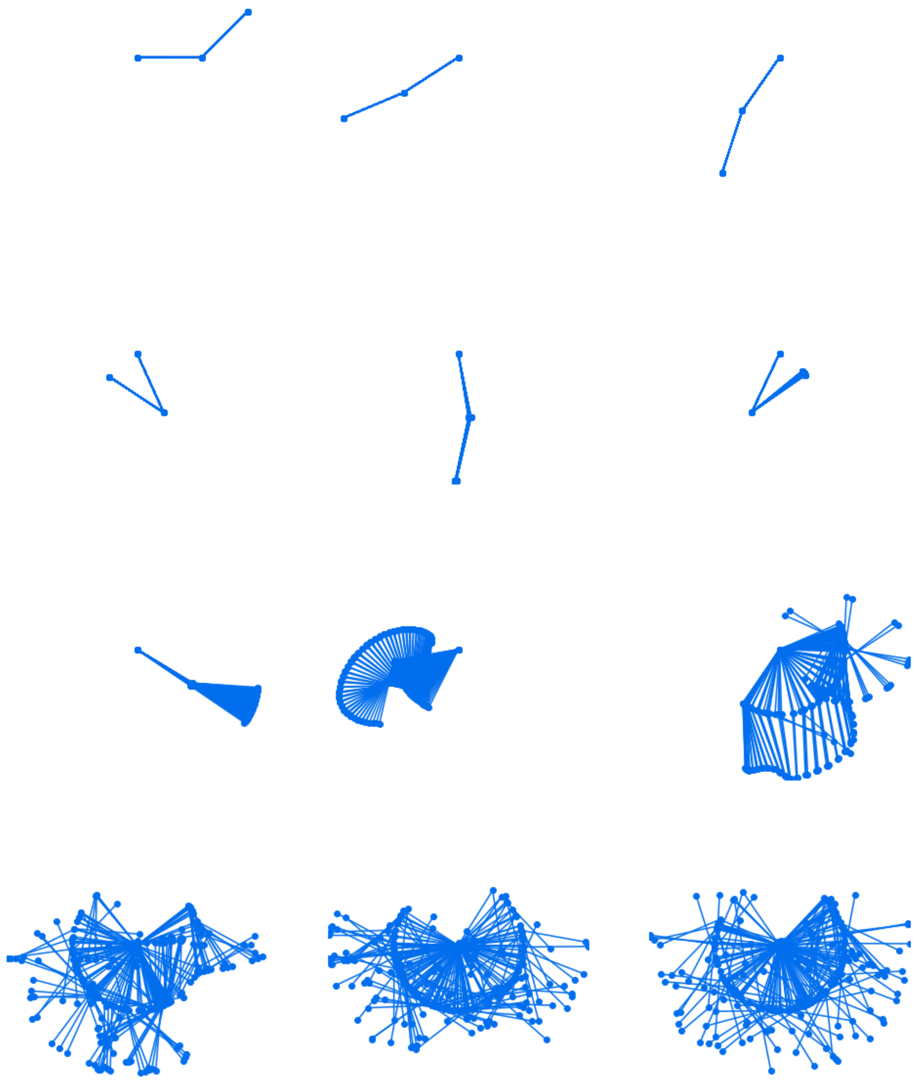


Figure 7: 121 pendulums starting from different initial conditions.

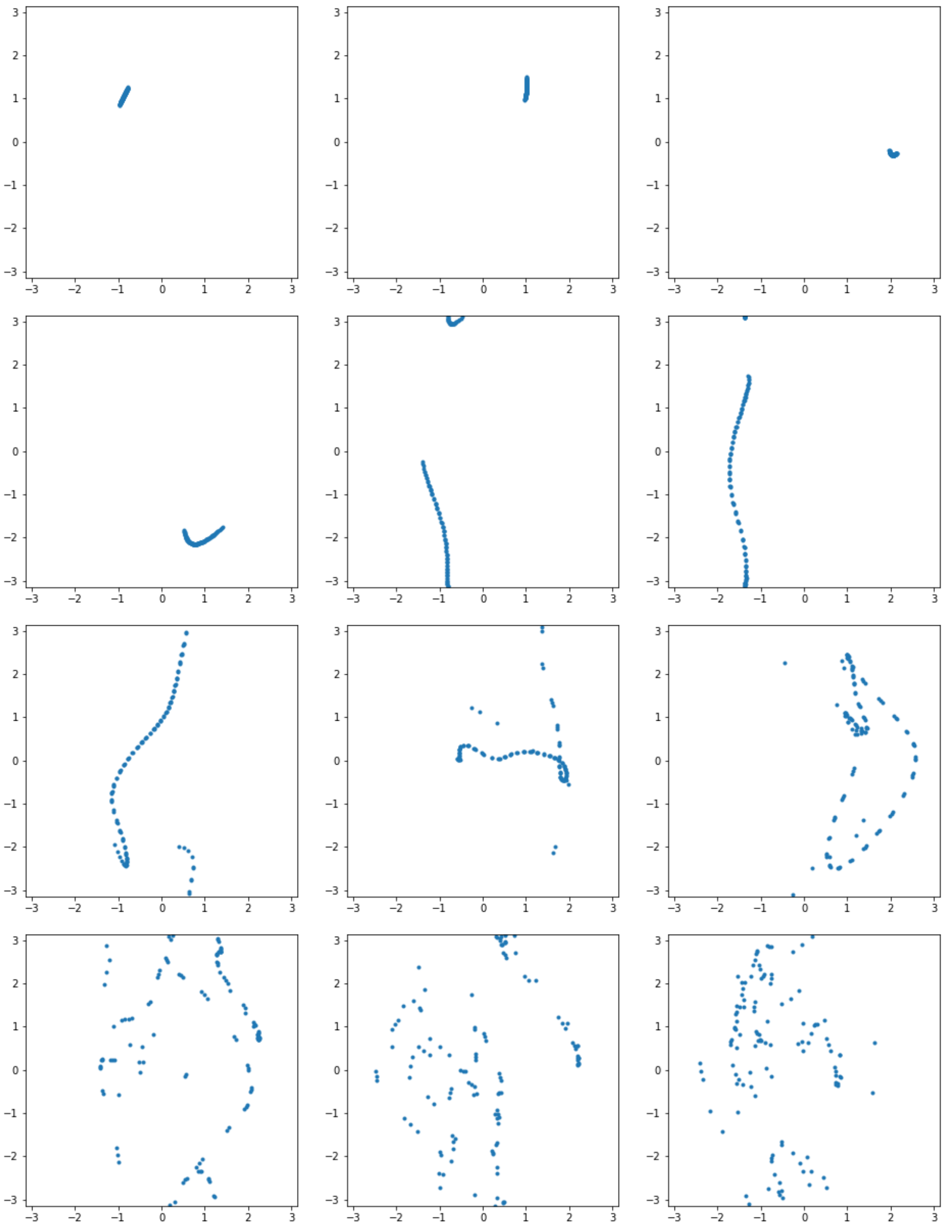


Figure 8: 121 pendulums starting from different initial conditions. Plot of (θ_1, θ_2) of each pendulum.

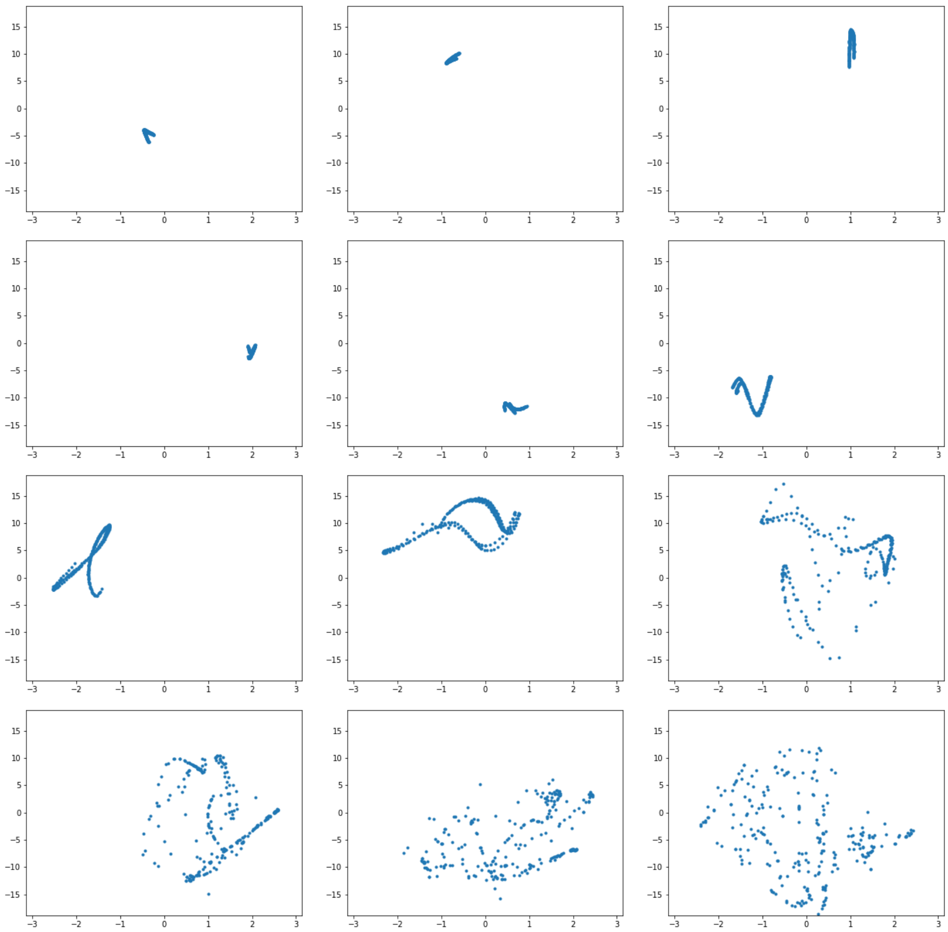


Figure 9: 169 pendulums starting from different initial conditions. Plot of (θ_1, z_1) of each pendulum.

References

[1] http://scienceworld.wolfram.com/physics/DoublePendulum.html

[2] https://scipython.com/blog/the-double-pendulum/