N-Pendulum

Roshan Tourani, 2008

Abstract

In this article we drive equations of motion for N-pendulum (in gravitational field and there is no other forces.), and study the numerical methods for solving these equations.

I. Theory of N-pendulum

Consider N connected pendulums with lengths $l_0, l_1, \cdots, l_{N-1}$ and masses $m_0, m_1, \cdots, m_{N-1}$. Position and velocity of n$^{\text{th}}$ mass, $\mathbf{r}_n, \mathbf{v}_n$, relative to the suspending point is given by,

$$\mathbf{r}_n = \left(\sum_{i=0}^{n} l_i \sin q_i\right)\hat{\mathbf{x}} - \left(\sum_{i=0}^{n} l_i \cos q_i\right)\hat{\mathbf{y}}, \tag{1}$$

$$\mathbf{v}_n = \left(\sum_{i=0}^{n} l_i \dot{q}_i \cos q_i\right)\hat{\mathbf{x}} + \left(\sum_{i=0}^{n} l_i \dot{q}_i \sin q_i\right)\hat{\mathbf{y}}, \tag{2}$$

where q_i is the angle between the i$^{\text{th}}$ pendulum and the vertical line. The kinetic and potential energy can be written as,

$$K = \sum_{n=0}^{N-1} \frac{1}{2} m_n \mathbf{v}_n . \mathbf{v}_n = \sum_{n=0}^{N-1} \frac{1}{2} m_n \sum_{i=0}^{n} \sum_{j=0}^{n} l_i l_j \dot{q}_i \dot{q}_j \cos(q_i - q_j), \tag{3}$$

$$U = U_0 + \sum_{n=0}^{N-1} m g \hat{\mathbf{y}} . \mathbf{r}_n = g \sum_{n=0}^{N-1} m_n \sum_{i=0}^{n} l_i (1 - \cos q_i), \tag{4}$$

and $\mathcal{L} = K - U$.
Now we can form equations of motion from Lagrange equations,

$$\frac{\partial \mathcal{L}}{\partial q_k} = -g l_k \sin q_k \sum_{n=k}^{N-1} m_n - \sum_{n=k}^{N-1} m_n \sum_{i=0}^{n} l_i l_k \dot{q}_i \dot{q}_k \sin(q_k - q_i), \tag{5}$$

$$\frac{\partial \mathcal{L}}{\partial \dot{q}_k} = \sum_{n=k}^{N-1} m_n \sum_{i=0}^{n} l_i l_k \dot{q}_i \cos(q_i - q_k), \tag{6}$$

$$\frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial \mathcal{L}}{\partial \dot{q}_k} = \sum_{n=k}^{N-1} m_n \sum_{i=0}^{n} l_i l_k \big[\ddot{q}_i \cos(q_k - q_i) - \dot{q}_i (\dot{q}_k - \dot{q}_i) \sin(q_k - q_i)\big], \tag{7}$$

finally equations of motion are given by,

$$g l_k \sin q_k \sum_{n=k}^{N-1} m_n + \sum_{n=k}^{N-1} m_n \sum_{i=0}^{n} l_i l_k \big[\ddot{q}_i \cos(q_k - q_i) + \dot{q}_i^2 \sin(q_k - q_i)\big] = 0, \quad k = 0, 1, \cdots, N-1, \tag{8}$$

we can write this equations as ($\mathbf{q} = (q_1, \cdots, q_N), \dot{\mathbf{q}} = (\dot{q}_1, \cdots, \dot{q}_N)$),

$$\boxed{\mathsf{M}\ddot{\mathbf{q}} = \mathbf{F}(\mathbf{q}, \dot{\mathbf{q}})}\,, \tag{9}$$

$$M_{ki} := \left(\sum_{n=\max(k,i)}^{N-1} m_n\right) l_k l_i \cos(q_k - q_i), \tag{10}$$

$$F_k := -g l_k \sin q_k \sum_{n=k}^{N-1} m_n - \sum_{n=k}^{N-1} m_n \sum_{i=0}^{n} l_i l_k \dot{q}_i^2 \sin(q_k - q_i). \tag{11}$$

Also generalized momentum, p_k, can be written as,

$$p_k := \frac{\partial \mathcal{L}}{\partial \dot{q}_k} = \sum_{n=k}^{N-1} m_n \sum_{i=0}^{n} l_i l_k \dot{q}_i \cos(q_i - q_k) = (\mathsf{M}\dot{\mathbf{q}})_k. \tag{12}$$

We can drive equations of motion from Hamiltonian by using equations,

$$\dot{p}_k = -\frac{\partial H}{\partial q_k},$$
$$\dot{q}_k = +\frac{\partial H}{\partial p_k}.$$

For this purpose we must write Hamiltonian in terms of q_i, p_i instead of $q_i, \dot{q}_i$,

$$\mathcal{H} = \frac{1}{2}\mathsf{M}\dot{\mathbf{q}}^T\dot{\mathbf{q}} + U = \frac{1}{2}\mathbf{p}^T\mathsf{M}^{-1}\mathbf{p} + U, \tag{13}$$

by this canonical Hamiltonian we can drive the equations of motion in terms of $(\mathbf{q}, \mathbf{p})$. Nevertheless we use equation (9) and also (12) for numerical computing. In fact we compute $(\mathbf{q}, \dot{\mathbf{q}})$ from equation (9) and in each time-step compute $\mathbf{p}$ from $\mathbf{p} = \mathsf{M}\dot{\mathbf{q}}$.

II. Numerical Method

The equations (8) are non-linear and can not be solved analytically. Here we describe a method for solving these equations numerically.
Take $\mathbf{s} := (\mathbf{q}, \dot{\mathbf{q}}) := (q_0, \cdots, q_{N-1}, \dot{q}_0, \cdots, \dot{q}_{N-1})$ as state. The equations of motion can be written as,

$$\mathsf{M}(\mathbf{q})\ \ddot{\mathbf{q}} = \mathbf{F}(\mathbf{q}, \dot{\mathbf{q}}) \tag{14}$$

$$\Rightarrow \quad \dot{\mathbf{s}} = \begin{bmatrix} \dot{\mathbf{q}} \\ \mathsf{M}^{-1}\mathbf{F}(\mathbf{q}, \dot{\mathbf{q}}) \end{bmatrix} = \mathbf{f}(\mathbf{s}) \tag{15}$$

which M is a state-dependent matrix and $\mathbf{F}$ is non-linear function given from (8).
Now we can use different algorithms for solving $\dot{\mathbf{s}} = \mathbf{f}(\mathbf{s})$. Euler-Richardson algorithm is good choice for this case, because the function $\mathbf{f}$ depends on velocity.