Nose-Hoover thermostat, Nose-Hoover chain, and stochastic thermostats for a classical harmonic oscillator in 1D

Scripts for plotting are available in the github repo: https://github.com/pyhope/thermostats

```
import numpy as np
from numpy.random import normal

def NH(dt, step, Q):
    kT = 1
    x, p, zeta, x_array, p_array = 1, 0, 0, np.zeros(step), np.zeros(step)
    for i in range(step):
        dxdt = p
        x_next = x + dxdt * dt
        dpdt = -x - zeta * p
        p_next = p + dpdt * dt
        dzdt = (p**2 - kT) / Q
        zeta_next = zeta + dzdt * dt
        x_array[i], p_array[i] = x_next, p_next
        zeta, x, p = zeta_next, x_next, p_next
    return x_array, p_array


def NHchain(dt, step, q, N):
    kT = 1
    Q = q * np.ones(N)
    x, p, x_array, p_array = 1, 0, np.zeros(step), np.zeros(step)
    pe, pe_next = np.zeros(N), np.zeros(N)
    for i in range(step):
        dxdt = p
        x_next = x + dxdt * dt
        p_next = p + (-x - pe[0] / Q[0] * p) * dt
        for k in range(N):
            if k == 0:
                dpdt = (p**2 - kT) - pe[1] / Q[1] * pe[0]
            elif k == N - 1:
                dpdt = pe[k - 1] ** 2 / Q[k - 1] - kT
            else:
                dpdt = (pe[k - 1] ** 2 / Q[k - 1] - kT) - pe[k + 1] / Q[k + 1] * pe[k]
            pe_next[k] = dpdt * dt + pe[k]
        x_array[i], p_array[i] = x_next, p_next
        pe, x, p = pe_next, x_next, p_next
    return x_array, p_array


def stochastic(dt, step, gamma):
    kT = 1
    x, p, x_array, p_array = 1, 0, np.zeros(step), np.zeros(step)
    C1 = np.exp(-gamma * dt / 2)
    C2 = np.sqrt((1 - C1**2) / (gamma * dt))
    for i in range(step):
        R1 = normal(0, kT * np.sqrt(gamma * dt))
        R2 = normal(0, kT * np.sqrt(gamma * dt))
        p_next = C1 * p + C2 * R1
        x_next = x + dt * p_next - (dt**2) / 2 * x
```

```
        p_next = p_next - dt/2 * (x_next + x)
        p_next = C1 * p_next + C2 * R2
        x_array[i], p_array[i] = x_next, p_next
        x, p = x_next, p_next
    return x_array, p_array
```