《非平衡统计物理专题》第一次作业

（作业提交截止时间： 3 月 22 日 21:00）

说明：

1）文件名格式范例：“第一次作业_19720192203306_程##.docx”

1. 有粒子处在简谐势阱中,其哈密顿表示如下：

$$H = \frac{p^2}{2} + \frac{1}{2}x^2$$

现放置该粒子于温度为 T 的热浴中作布朗运动。

1）试写出刻画该粒子运动的郎之万方程；

2）解析计算该粒子能量平均值随温度 T 的变化规律；

3）采用分子动力学模拟（随机 RK2 算法）验算 2）的结果。

解：

1）该粒子运动的郎之万方程应为：

$$\begin{cases} \dfrac{dx}{dt} = v \\ m\dfrac{dv}{dt} = -\gamma v - x + \xi(t) \end{cases} \tag{1}$$

其中郎之万力 $\xi(t)$ 为白噪声，满足以下条件：

$$\begin{cases} <\xi(t)>=0 \\ <\xi(t_1)\xi(t_2)>=2\gamma k_B T\delta(t_1 - t_2) \end{cases}$$

也即郎之万力均值为 0，且前后不相关。

其中，γ 为粘滞力系数，k_B 为玻尔兹曼常数，T 为温度。

2）该系统动能可以表示为：

$$E_k(t)=\frac{1}{2}mv(t)^2$$

该系统势能可以表示为：

$$E_p(t)=\frac{1}{2}x(t)^2$$

该系统的总能量可以表示为：

$$\begin{aligned}E(t)&=E_p(t)+E_k(t)\\&=\frac{1}{2}mv(t)^2+\frac{1}{2}x(t)^2\end{aligned}$$

因此该能量平均则可以表示为：

$$\begin{aligned}<E(t)>&=<E_p(t)>+<E_k(t)>\\&=<\frac{1}{2}mv(t)^2>+<\frac{1}{2}x(t)^2>\\&=\frac{1}{2}m<v(t)^2>+\frac{1}{2}<x(t)^2>\end{aligned} \tag{2}$$

也即求解郎之万方程，再求取粒子位置和速度的自相关函数的时间平均即可（时间平均等于系综平均）。

对（1）式作傅里叶变换有：

$$\begin{cases}-i\omega x(\omega)=v(\omega)\\ m\cdot[-i\omega v(\omega)]=-\gamma v(\omega)-x(\omega)+\xi(\omega)\end{cases}$$

可以分别解出

$$x(\omega)=\frac{1}{m}\cdot\frac{\xi(\omega)}{\frac{1}{m}-\omega^2-\frac{\gamma}{m}i\omega} \tag{3}$$

$$v(\omega)=\frac{\xi(\omega)}{\gamma-\frac{1}{i\omega}-i\omega m} \tag{4}$$

由“谱频率为自相关函数的傅里叶变换”出发。现在令：谱频率$S_x(\omega)$ 为粒子位置的自相关函数$C_x(t)$的傅里叶变换；谱频率$S_v(\omega)$ 为粒子速度的自相关函数$C_v(t)$的傅里叶变换；谱频率$S_\xi(\omega)$ 为郎之万力的自相关函数$C_\xi(t)$的傅里叶变换。

由$<\xi(t_1)\xi(t_2)>=2\gamma k_B T\delta(t_1-t_2)$ 有$S_\xi(\omega)=\int_{-\infty}^{\infty}<\xi(t_1)\xi(t_2)>e^{i\omega t}dt=2\gamma k_B T$ 。

利用傅里叶逆变换，由（3）式有

$$\begin{aligned}
C_x(t)&=\frac{1}{2\pi}\int_{-\infty}^{\infty}x(\omega)^2 e^{-i\omega t}d\omega=\frac{1}{2\pi}\int_{-\infty}^{\infty}[\frac{1}{m^2}\cdot\frac{\xi(\omega)^2}{(\frac{1}{m}-\omega^2-\frac{\gamma}{m}i\omega)^2}]e^{-i\omega t}d\omega \\
&=\frac{1}{2\pi}\int_{-\infty}^{\infty}[\frac{1}{m^2}\cdot\frac{2\gamma k_B T}{(\frac{1}{m}-\omega^2-\frac{\gamma}{m}i\omega)^2}]e^{-i\omega t}d\omega \\
&=\frac{2\gamma k_B T}{2m^2\pi}\int_{-\infty}^{\infty}[\frac{1}{(\frac{1}{m}-\omega^2-\frac{\gamma}{m}i\omega)^2}]e^{-i\omega t}d\omega
\end{aligned} \quad (5)$$

利用傅里叶逆变换，由（4）式有

$$\begin{aligned}
C_v(t)&=\frac{1}{2\pi}\int_{-\infty}^{\infty}v(\omega)^2 e^{-i\omega t}d\omega=\frac{1}{2\pi}\int_{-\infty}^{\infty}\frac{\xi(\omega)^2}{(\gamma-\frac{1}{i\omega}-i\omega m)^2}e^{-i\omega t}d\omega \\
&=\frac{1}{2\pi}\int_{-\infty}^{\infty}\frac{2\gamma k_B T}{(\gamma-\frac{1}{i\omega}-i\omega m)^2}e^{-i\omega t}d\omega \\
&=\frac{2\gamma k_B T}{2m^2\pi}\int_{-\infty}^{\infty}\frac{1}{(\frac{\gamma}{m}-\frac{1}{i\omega m}-i\omega)^2}e^{-i\omega t}d\omega
\end{aligned} \quad (6)$$

（这个地方没积出来，还在思考中~）

3）

由“时间平均等于系综平均”出发，对（1）式利用欧拉法和 SRK2 算法结合进行模拟：微分方程使用龙格库塔算法，随机微分方程部分使用随机龙格库塔算法。

在松弛一段时间后，取模拟的时间平均作为（2）式的估计值。欧拉法和 SRK2 算法与 PPT 给出的流程基本一致，这里不再赘述，其中，由涨落耗散定理可有 $D=\gamma k_B T$ 。

假定 $m=\gamma=1$，模拟的时间为 100s，松弛时间为 20s。模拟的粒子能量平均值随温度 T 的变化规律如图 1 所示。

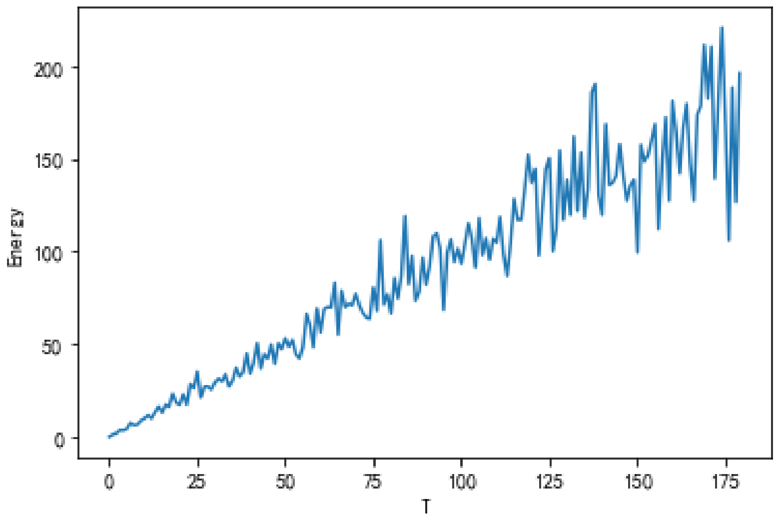

图 1 粒子能量平均值随温度 T 的变化规律

代码（https://github.com/Acpnohc/conputational_method_in_theory_physics/tree/main/hw2 ）在该网址下载运行。

代码问题与思考

方程（1）我认为实际上是一个 随机微分方程组。

1. 根据第二节课老师您说分两部分处理：第一部分用龙格库塔算法，第二部分用随机龙格库塔算法。但是实际上第一个方程似乎不适合用龙格库塔：因为二阶要向前递推一步，除非用插值，不然怎么递推？

2. 微分方程组的龙格库塔有另外的形式，不知是我对流程图的理解有误，我感觉得用方程组形式的，我还没有尝试，后续会和您继续讨论。

参考文献

---

[1] Force calibration and noise in single-moleculemanipulation experiments
[2] Yaghoubi, Mohammad, et al. "Energetics of a driven Brownian harmonic oscillator." Journal of Statistical Mechanics: Theory and Experiment 2017.11 (2017): 113206.

代码附录

```
# -*- coding: utf-8 -*-
"""
Created on Mon Mar 22 19:03:14 2021

@author: JChonpca_Huang
"""

import numpy as np
import matplotlib.pyplot as plt


def f_sode(x,v):

    return    -x -v

def SRK2(fv,h,t,x0,v0,T):

    XX = np.zeros(int(t/h)+1)
    XX[0] = x0

    VV = np.zeros(int(t/h)+1)
    VV[0] = v0

    noise = np.random.normal(loc=0,scale=1,size=int(t/h))*np.sqrt(2*T*h)

    for i in range(1,int(t/h)+1):

        vk1 = fv(XX[i-1], VV[i-1])
        vk2 = fv(XX[i-1], VV[i-1] + h*vk1 + noise[i-1])

        VV[i] = VV[i-1] + (1/2)*h*(vk1+vk2) + noise[i-1]
```

```
            XX[i] = XX[i-1] + VV[i-1]*h

        return [XX,VV]

def E_stat(P):

        x = P[0][2000::]

        v = P[1][2000::]

        E = 0

        for i in range(len(x)):

            E = E + (1/2)*(x[i]**2 + v[i]**2)

        return E/len(x)

X = []
Y = []
P = []

for T in range(0,180):


     p = SRK2(f_sode,0.01,100,0,0,T)
     P.append(p)
     X.append(T)
     Y.append(E_stat(p))


plt.plot(X,Y)

plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False

plt.xlabel("T")
plt.ylabel("Energy")

plt.show()
```