作业 1：

2025 年 3 月 12 日

请提交源代码和红字描述的结果，每个作业中的（1）（2）……是提供思路和流程，并非需要提交的结果（当然，愿意提交中间步骤的结果也鼓励）。

1

已知冷暗物质（cold dark matter，简称 CDM）密度场的功率谱为

$$P_{\rm CDM}(k) = A_s k^{n_s} T^2(k) \tag{1}$$

其中 A_s 是一个归一化系数，通过观测数据确定。$n_s = 0.96$，$T(k)$ 是 transfer function，从给定的表格中插值得到（提示：插值的时候 $k \to \ln k$ 或者 $\log_{10} k$，$T(k) \to \ln T(k)$ 或者 $\log_{10} T(k)$，这样曲线更加平滑，插值误差更小）。

平滑尺度 R 的密度起伏的 variance 为：

$$\sigma^2(R) = \int_0^\infty \frac{4\pi k^2}{(2\pi)^3} P_{\rm CDM}(k) W^2(k, R) dk, \tag{2}$$

其中窗函数

$$W(k, R) = \frac{3[\sin(kR) - (kR)\cos(kR)]}{(kR)^3} \tag{3}$$

为实空间中半径为 R 的 top-hat 函数在傅里叶空间里的形式。记 $\sigma(R = 8h^{-1}{\rm Mpc}) = \sigma_8$，观测表明，$\sigma_8 = 0.82$。（提示：算积分的时候，用变量代换 $t = \ln k$，$dk = k dt$，这样更容易收敛。）

先设 $A_s = 1$，取 $R = 8h^{-1}$ Mpc，$h = 0.6774$，从 Eq. (2) 计算出 $\sigma^2(R = 8h^{-1}{\rm Mpc})$，然后用观测的值 $(0.82)^2$ 除以这个计算出来的结果，即得到归一化系数 A_s。确定 A_s 之后，**将 $P_{\rm CDM}(k)$ 作为 k 的函数画出来，k 的取值范围为 $10^{-5} - 10^3$ Mpc^{-1}**。

提示：实际积分 Eq. (2) 时，不用真的从 0 积分到无穷大，从 $k = 10^{-5}$ 积分到 10^5 即可，或者自己试一试看取什么范围结果就已经收敛。

2

温暗物质（warm dark matter，简称 WDM）的功率谱为：

$$P_{\rm WDM}(k) = P_{\rm CDM}(k) T^2_{\rm WDM}(k), \tag{4}$$

其中

$$T_{\rm WDM}(k) = (1 + (\alpha k)^{2\mu})^{-5/\mu}, \tag{5}$$

$\mu = 1.12$,

$$\alpha = 0.049 \left(\frac{m_{\rm WDM}}{\rm keV}\right)^{-1.11} \left(\frac{\Omega_{\rm WDM}}{0.25}\right)^{0.15} \left(\frac{h}{0.7}\right)^{1.22} h^{-1}\ [\rm Mpc]. \tag{6}$$

取 $\Omega_{\rm WDM} = \Omega_m = 0.32$，$m_{\rm WDM} = 10$ keV，将 Eq. (2) 中的 $P_{\rm CDM}(k)$ 替换成 $P_{\rm WDM}(k)$，用 (1) 中同样的步骤，先确定 A_s，**然后再画出** $P_{\rm WDM}(k)$。

相仿的，**把** $m_{\rm WDM} = 1$ **keV 和 0.1 keV 的** $P_{\rm WDM}(k)$ **也画出来。**

图 1 中给出了一个 Python 使用插值函数的例子，图 2 中给出了一个 Python 使用积分函数的例子，作为参考。

```python
# 开始把需要用到的package或者其中的function先导入
import numpy as np
from scipy.interpolate import interp1d


#读入transfer function的表格，用于插值。注意忽略第一行，因为那一行是用于说明的文字
data=np.loadtxt('./transfer-function.txt',skiprows=1)


#data的第一列是k，第二列是T(k)，k是*对数均匀*分割的，同时T(k)本身的数量级涵盖范围较广。为了
#使得插值更加准确，都取对数，以使数据更加平滑
data_lgk=np.log10(data[:,0])
data_lgTk=np.log10(data[:,1])

#先定义插值函数，插值方式选择三次样条插值"cubic"
lgTk_interp=interp1d(data_lgk,data_lgTk,kind='cubic')

#然后从插值函数定义一个Tk，输入k之后返回插值得出的T(k)。后面的[()]符号是因为interp1d函数默认返回一个矩阵，
#即使只有一个元素也是矩阵类型，有时做函数使用的时候不方便。[()]可以把只有一个元素的矩阵转换为标量。
def Tk(k):
    return 10**lgTk_interp(np.log10(k))[()]


#然后Tk就可以作为函数使用，给出任意k对应的Tk。

#e.g. k=1e-3
k=1e-3
a=Tk(k)
print('k=',k,'Tk=',a)

#e.g. k=10.0
k=10.0
a=Tk(k)
print('k=',k,'Tk=',a)
```

图 1: Python 插值使用举例（如运行程序时遇到问题，可以删掉中文注释）。

```python
import numpy as np
from scipy.integrate import quad


#被积函数只有自变量x，无参数
def fx1(x):
    return np.sin(x)

a=0.0
b=np.pi/2

# quad返还积分结果I和估计的误差err，我们只需要积分结果即可
I,err=quad(fx1,a,b,epsrel=1e-3)

print('int_0_^pi/2 sin(x)dx=',I)


#被积函数除了自变量x之后，还有一个参数omega。
def fx2(x,oemga):
    return np.sin(omega*x)

a=0.0
b=np.pi/2

omega=2.3

I,err=quad(fx2,a,b,args=(omega),epsrel=1e-3)

print('int_0_^pi/2 sin(2.3x)dx=',I)
```

图 2: Python 积分函数 quad 使用举例（如运行程序时遇到问题，可以删掉中文注释）。