

Cosmology

§1 Tools

Lecturer: 黄志琦

<http://zhiqihuang.top/cosm>

符号推演工具sympy



Sympy

在本课程中，我们将借助python的符号演算库sympy来进行运算。

```
#.....source code: .....  
  
import sympy as sym  
  
# .....
```

Warning: 一般不提倡为了省事而 `from sympy import *`，这样可能会产生大量的重名冲突问题；你当然也知道用 `numpy` 时，尽量要采用 `import numpy as np` 而不是 `from numpy import *`，都是同一个道理。

Sympy Example: 求极限

```
#.....source code: lim.py .....  
  
import sympy as sym  
x = sym.symbols('x')  
print(sym.limit((sym.log(1+x)-x)/x**2, x, 0))  
print(sym.limit((1+x)**(1/x), x, 0))  
  
# .....
```

Sympy Example: 求导

```
#.....source code: deriv.py .....  
  
import sympy as sym  
x, y = sym.symbols('x, y')  
print(sym.diff(sym.sin(x), x))  
f = x * sym.exp(y)  
print(sym.diff(f, x))  
print(sym.diff(f, y))  
  
# .....
```

Sympy Example: 不定积分运算

```
#.....source code: indefint.py .....  
  
import sympy as sym  
x, n = sym.symbols('x, n')  
print(sym.integrate(sym.log(x),x))  
print(sym.integrate(x**n, x))  
  
# .....
```

Sympy Example: 定积分运算

```
#.....source code: defint.py .....  
  
import sympy as sym  
x, a = sym.symbols('x, a')  
print(sym.integrate(x**2, (x, -1, 1)))  
print(sym.integrate(sym.exp(-a*x**2), (x, -sym.oo, sym.oo)))  
  
# .....
```

Sympy Example: 级数展开

```
#.....source code: expan.py .....  
  
import sympy as sym  
x, a = sym.symbols('x, a')  
f = sym.sqrt(a**2+x**2)  
print(sym.series(f, x, 0, 4))  
print(sym.series(sym.sinh(x)*sym.sin(x),x, 0, 12))  
  
# .....
```


Sympy Example: 化简

```
#.....source code: simp.py .....  
  
import sympy as sym  
x = sym.symbols('x')  
expr = sym.cos(3*x)-4*sym.cos(x)**3  
print(sym.simplify(expr))  
expr = (x**3+x**2-x-1)/(x**2+2*x+1)  
print(sym.simplify(expr))  
f, g = sym.symbols('f, g', cls=sym.Function)  
expr = f(g(x))  
print(sym.simplify(sym.diff(expr, x)/sym.diff(g(x), x)))  
  
# .....
```

科学计算工 具 **scipy** 和 **numpy**

插值和解微分方程

numpy, scipy 和 pyplot

我们将频繁地使用 `numpy` 和 `scipy` 进行科学运算，并用 `pyplot` 将结果可视化。

`scipy` 里面又包含很多子库，因此我们经常的使用方法是：

```
#.....source code: .....
```

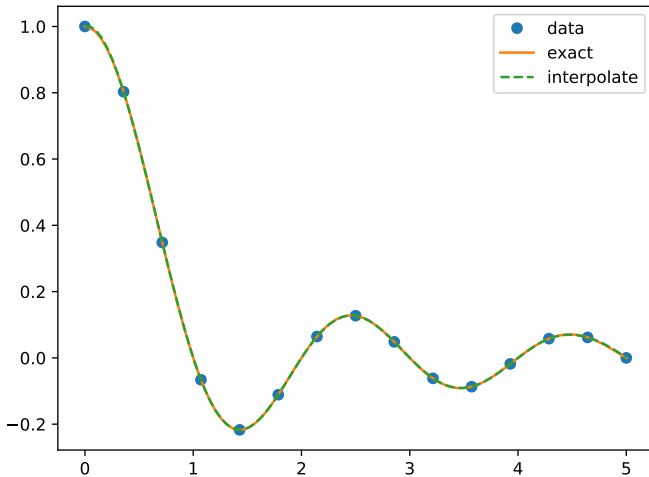
```
import numpy as np
import scipy.xxx as ...
import matplotlib.pyplot as plt
```

```
# .....
```

插值的例子

```
#.....source code: interp.py .....  
  
import numpy as np  
import scipy.interpolate as itp  
import matplotlib.pyplot as plt  
x = np.linspace(0., 5., 15)  
y = np.sinc(x)  
f3 = itp.interp1d(x, y, kind='cubic')  
xdense = np.linspace(0.,5., 100)  
yapprox = f3(xdense)  
yexact = np.sinc(xdense)  
plt.plot(x, y, 'o', xdense, yexact, '--', xdense, yapprox, '--')  
plt.legend(['data', 'exact', 'interpolate'], loc='best')  
plt.show()  
  
# .....
```

运行结果



有时候你希望把图存为pdf格式

```
#.....source code: saveinterp.py .....

import numpy as np
import scipy.interpolate as itp
from matplotlib import use
use('pdf')
import matplotlib.pyplot as plt
x = np.linspace(0., 5., 15)
y = np.sinc(x)
f3 = itp.interp1d(x, y, kind='cubic')
xdense = np.linspace(0.,5., 100)
yapprox = f3(xdense)
yexact = np.sinc(xdense)
plt.plot(x, y, 'o', xdense, yexact, '-', xdense, yapprox, '--')
plt.legend(['data', 'exact', 'interpolate'], loc='best')
plt.savefig('interp.pdf', format='pdf')

# .....
```

解ODE的例子I

我们来解一个简单的微分方程：

$$\frac{d^2x}{dt^2} + x = 0, \quad x(0) = 1, x'(0) = 0$$

这个问题的解显然是 $x = \cos t$ 。

不过，我们要假装不知道这个严格解，采用数值解法。为此，我们要通过引入中间变量的方法把所有的二阶以上的导数转化为一阶导数：

令矢量 $\begin{pmatrix} y_0 \\ y_1 \end{pmatrix} = \begin{pmatrix} x \\ \frac{dx}{dt} \end{pmatrix}$ ，则它满足一阶微分方程：

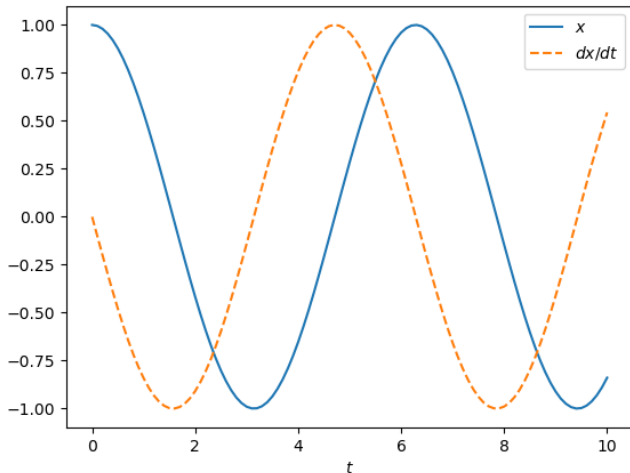
$$\frac{d}{dt} \begin{pmatrix} y_0 \\ y_1 \end{pmatrix} = \begin{pmatrix} y_1 \\ -y_0 \end{pmatrix}$$

这已经是标准数值ODE算法可以接受的形式了。

解ODE的例子I

```
#.....source code: solveode1.py .....  
  
import numpy as np  
import matplotlib.pyplot as plt  
import scipy.integrate as itg  
def diff(y, t):  
    return np.array([y[1], -y[0]])  
  
t = np.linspace(0., 10., 101)  
y = itg.odeint(diff, [1., 0.] , t)  
plt.xlabel(r'$t$')  
plt.plot(t, y[:, 0], '-', t, y[:, 1], '--')  
plt.legend([r'$x$', r'$dx/dt$'], loc='best')  
plt.show()  
  
# .....
```


运行结果



解ODE的例子II

下面我们来考虑耦合的非线性受迫阻尼振动的例子

$$\frac{d^2x}{dt^2} + \frac{dx}{dt} + xy^2 = \sin t,$$

$$\frac{d^2y}{dt^2} + \frac{dy}{dt} + x^2y = \cos t,$$

$$x(0) = 1, x'(0) = 0, y(0) = 1, y'(0) = 0$$

解ODE的例子II

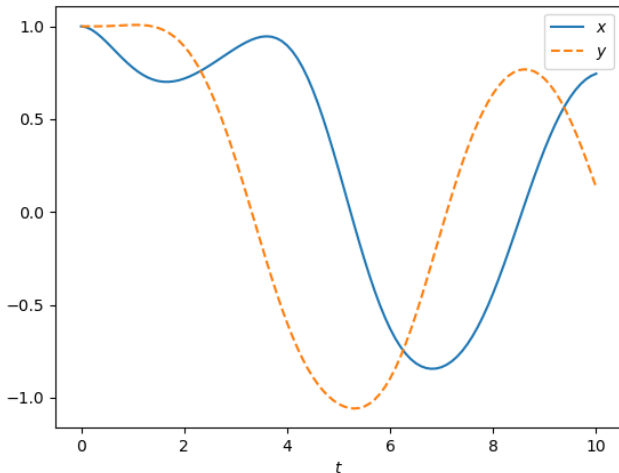
我们的矢量就是 $\begin{pmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \end{pmatrix} = \begin{pmatrix} x \\ \frac{dx}{dt} \\ y \\ \frac{dy}{dt} \end{pmatrix}$, 它满足一阶的微分方程:

$$\frac{d}{dt} \begin{pmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \end{pmatrix} = \begin{pmatrix} z_1 \\ \sin t - z_0 z_2^2 - z_1 \\ z_3 \\ \cos t - z_0^2 z_2 - z_3 \end{pmatrix}$$

解ODE的例子II

```
#.....source code: solveode2.py .....  
  
import numpy as np  
import matplotlib.pyplot as plt  
import scipy.integrate as itg  
def diff(z, t):  
    return np.array([z[1], np.sin(t)-z[0]*z[2]**2-z[1],  
z[3], np.cos(t)-z[0]**2*z[2]-z[3]])  
  
t = np.linspace(0., 10., 101)  
z = itg.odeint(diff, [1., 0., 1., 0.] , t)  
plt.xlabel(r'$t$')  
plt.plot(t, z[:, 0], '- ', t, z[:, 2], '--')  
plt.legend([r'$x$', r'$y$'], loc='best')  
plt.show()  
  
# .....
```

运行结果



好了代码就讲到这里

python是“互联网语言”，你并不需要专门进行学习，只要会用搜索引擎寻找答案就行。



同志快醒醒 你还有一串
代码提示错误 起来改改

如果你运行我放在网站上的代码不成功，很可能是安装时的环境配置问题。

- 检查默认是否使用python3而不是python2（python2 已经在2020年被官宣淘汰了）
- 检查sympy,matplotlib, scipy, numpy等库是否安装
- 检查pyplot是否支持latex

自然单位制和量纲

$$c = \hbar = k_B = 1$$

自然单位制和自然单位制量纲

自然单位制 $c = \hbar = k_B = 1$ 下:

长度量纲 = 时间量纲 = 质量量纲⁻¹, 能量量纲 = 温度量纲 = 质量量纲

因此自然单位制下所有量纲都可以写成质量的若干次方。



力的量纲

= 质量量纲 × 长度量纲 / 时间量纲²

= 质量量纲²

思考题



下面物理量的自然单位制量纲是质量的多少次方？

速度 v

角速度 ω

角动量 L

熵 S

能量密度 ρ

压强 p

牛顿引力常数 G

普通单位制等式和自然单位制等式之间的转换

- 普通单位制 \Rightarrow 自然单位制:
把所有 c , \hbar 和 k_B 都换成 1; 得到的等式自然单位制量纲应该是自动配平的。
- 自然单位制 \Rightarrow 普通单位制:
 - ① 如果等式中含有温度, 先把所有温度配上 k_B 使之成为能量量纲;
 - ② 取一项作为基准项, 计算它的普通单位制MLT量纲 (质量 M 若干次*长度 L 若干次*时间 T 若干次)
 - ③ 其余每一项都配上形如 $c^\alpha \hbar^\beta$ 的因子使得其MLT量纲和基准项相同。

Planck量

通过把 c, \hbar, G, k_B 适当组合，我们能得到所有五种量纲的量（另外两个国际单位摩尔和弧度是无量纲的）。这些量就称为Planck量。

$$\text{普朗克时间 } t_P = \sqrt{\frac{G\hbar}{c^5}} = 5.391 \times 10^{-44} \text{ s}$$

$$\text{普朗克长度 } l_P = \sqrt{\frac{G\hbar}{c^3}} = 1.616 \times 10^{-35} \text{ m}$$

$$\text{普朗克质量 } m_P = \sqrt{\frac{\hbar c}{G}} = 2.176 \times 10^{-8} \text{ kg}$$

$$\text{普朗克能量 } E_P = \sqrt{\frac{\hbar c^5}{G}} = 1.956 \times 10^9 \text{ J}$$

$$\text{普朗克温度 } T_P = \sqrt{\frac{\hbar c^5}{k_B^2 G}} = 1.417 \times 10^{32} \text{ K}$$

Planck单位制

在自然单位制基础上再令 $G = 1$ 就得到 Planck 单位制。
在Planck单位制里所有量都是无量纲的。

Warning: 从普通单位制 到 自然单位制 再到 Planck单位制，公式越来越简洁。但要注意付出的代价：缺少了校对量纲的工具，推导出错的概率逐级增加；当你把物理常数都省略不写，也可能会丢失一些物理图像。

普通单位制等式和Planck单位制等式之间的转换

- 普通单位制 \Rightarrow Planck单位制:
把所有 c , \hbar , k_B , G 都换成 1
- Planck单位制 \Rightarrow 普通单位制:
把每一个物理量都换成它和对应的Planck量之比

自然单位制 \Rightarrow 普通单位制也变得简单了：扔掉 G 变成Planck单位制，然后再按照上面的方法变到普通单位制。

思考题

天文学的距离单位秒差距 pc 大约是 3.26 光年。宇宙学里经常用百万秒差距 $\text{Mpc} = 10^6 \text{ pc}$ 做距离单位。请问： Mpc^{-1} 等于多少 eV 的能量？



根本不知道你在说什么.jpg

思考题

宇宙学里把每一种成份(不妨记为 x)的平均能量密度 (设为 ρ_x) 用无量纲参数:

$$\Omega_x \equiv \frac{8\pi G \rho_x}{3H_0^2}$$

来表示。这里的 G 是牛顿引力常数, H_0 是哈勃常数。

从量子统计理论可以推导出温度为 T 的光子气体的能量密度为

$$\rho_\gamma = \frac{\pi^2}{15} T^4.$$

取哈勃常数 $H_0 = 70\text{km/s/Mpc}$, 计算宇宙微波背景辐射 (温度为 2.73K 的光子气体) 的能量无量纲参数 Ω_γ 。

Planck量的计算以及思考题的解答请参考



<http://zhiqihuang.top/cosm/codes/units.py>

统计置信度

已知你本课通过的置信度为 2.2222σ ，请问是否要退课？

思考题



已知“你可以通过本课”这个结论的置信度(confidence level)为 2.2222σ ，请问你挂科概率是多大？

思考题

你自己测量你的 (身高, 体重) = (180, 180), 测量误差关联矩阵(covariance matrix)为:

$$\begin{pmatrix} 100 & 10 \\ 10 & 100 \end{pmatrix}$$

医院测量到你的 (身高, 体重) = (140, 240), 测量误差关联矩阵(covariance matrix)为:

$$\begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$$

请问两次测量之间有几个 σ 的不一致性(tension)?