科学计算基础 (1)

September 13, 2022

一 内容

1 Python 运行环境 (以 Jupyter 为例)

1.1 创建 cell，运行 cell，写 Markdown 类 cell

2 简单运算

2.1 实数运算

运算	加法	减法	乘法	除法	幂方
符号	+	−	*	/	**

2.2 常用函数 (sin, cos, exp, log)

函数名称	正弦	余弦	正切	以 e 为底的指数函数	以 e 为底的对数函数
数学符号	$\sin(x)$	$\cos(x)$	$\tan(x)$	e^x	$\ln(x)$
Numpy 函数符号	np.sin(x)	np.cos(x)	np.tan(x)	np.exp(x)	log(x)

注意: 数学符号与 Numpy 函数符号的区别

3 矩阵输入

3.1 向量

numpy.array(), numpy.arange(), numpy.linspace(), numpy.zeros(), numpy.ones()

3.2 矩阵

numpy.array(), numpy.zeros(), numpy.ones()

说明: 控制对以上函数输入的维度来获得向量或矩阵

4 矩阵运算

4.1 《线性代数》中提及的矩阵运算

运算	加法	减法	乘法	幂方	转置
符号	+	−	@ 或 dot()	**	.T

注意:Python 矩阵除法运算需要通过 numpy.linalg.inv() 方法求解矩阵的逆来实现.

4.2 点运算 (两个同型矩阵对应元素的运算)

运算	点乘	点除
符号	*	/ 或 numpy.devide

4.3 其他矩阵运算

运算	行列式	秩	特征值
Numpy 函数	np.linalg.det	np.linalg.matrix_rank	np.linalg.eig

5 矩阵函数

5.1 正弦，余弦，指数，对数 (与实数函数符号相同)

5.2 其他函数

numpy.ones(), numpy.zeros(), numpy.abs(), numpy.max(), numpy.min()

6 变量

7 py 文件: 命名规则，保存路径

8 多项式相关

- 多项式表示 (数值计算)

np.poly1d 函数

```python
import numpy as np
a = np.array([2,1,1])
np.poly1d(a)
```

np.roots 函数

```python
p = np.array([3, -2, -4])
r = np.roots(p)
```

np.convolve 函数

```python
u = np.array([1, 0, 1])
v = np.array([2, 7])
w = np.convolve(u, v)
```

np.polyval 函数

```python
coef = np.array([1, 0, -1])
np.polyval(coef, np.arange(-1, 1, 0.1))
```

- 多项式表示 (符号计算)

sympy

```python
from sympy import *
x = Symbol('x')
```

solve

```python
solve(Eq(x**2 - 1))
```

- 符号计算转为数值运算

lambdify 函数

```python
a = np.pi / 3
x = symbols('x')
expr = sin(x)
f = lambdify(x, expr, 'numpy')
f(a)
```

9 画图

9.1 matplotlib.pyplot.plot 函数

绘制函数 $\frac{1}{1+x^2}$ 在区间 $[-5,5]$ 上的图形

绘制多项式 $2.1x^4-1.2x+2$ 在区间 $[-1,2]$ 上的图形

二 练习

1. 计算如下表达式的值

   (1) $1+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}$

   (2) $\sin(\frac{\pi}{15})\cos(1.234)$

   (3) $e^2+\ln(\frac{6}{5})-1.01^{-10}$

2. 已知 $A=\begin{bmatrix}1&1&1\\1&1&-1\\1&-1&1\end{bmatrix}, B=\begin{bmatrix}1&2&3\\-1&-2&4\\0&5&1\end{bmatrix}$，求 $3AB-2A$ 和 A^TB.

3. 已知 $A=\begin{bmatrix}1&1&0\\0&1&1\\0&0&1\end{bmatrix}$，求 A^{22}.

4. 已知 $A=\begin{bmatrix}1&2\\2&5\end{bmatrix}$，计算

   (1) $\sin(A).*\cos(A)$ (2) $\sin(A)*\cos(A)$ (3) 行列式 $|A|$

   (4) A 的秩 (5) A 的全部特征值

5. 已知 $A=[1\ \ 2\ \ 3\ \ -2\ \ 4\ \ 1]$，计算 A 的绝对值，最大值，最小值.

6. 用以下两种数学上等价的表达式进行计算，对比结果，解释误差产生的原因.

$$\ln(10^{15}+1)-\ln(10^{15}),\quad \ln\left(\frac{10^{15}+1}{10^{15}}\right)=\ln(1+10^{-15})$$

7. 分别使用 np.roots 和 solve 函数，计算 $x^2-56x+1=0$ 的两个根.

8. 计算多项式乘法

(1) $(x^2+1)(-3x+1)$ (2) $(x+1)(x^3+x+9)$

9. 计算多项式 x^4+8x^3-10 除以多项式 $2x^2-x+3$.

10. 计算多项式 $x^{20}-1$ 在点 $x=0,1,\cdots 10$ 处的函数值.

11. 绘制以下函数在区间 $[1,2]$ 上的图形.

(1) $e^x-\cos(20\pi x)$ (2) $\tan(\ln(10x))$

三 作业

1. 二进制产生的误差

实验目的: 理解计算机内部的二进制运算法则及其产生的误差

实验内容: 用 Python 计算

$$\sum_{i=1}^{1000} 0.1-100$$

观察发生的现象并试着解释原因.

提示: 结果居然有误差! 因为从十进制数角度分析, 这一计算应该是准确的. 该实验反映了计算机内部的二进制本质.

2. 数值稳定性

实验目的: 对数值稳定性有一定的直观感受.

实验内容: 考虑一个高次的代数多项式.

$$p(x)=(x-1)(x-2)\cdots(x-20)=\prod_{k=1}^{20}(x-k)$$

显然，该多项式的全部根为 $1,2,\cdots,20$ 共计 20 个，且每个根都是单重的.

现考虑该多项式的一个扰动

$$p^{\epsilon}(x)=p(x)+\epsilon x^{19}$$

其中 ϵ 是一个非常小的数. 这相当于对原多项式中 x^{19} 的系数做一个小扰动.

试通过以下实验分析比较多项式 $p(x)$ 和 $p^{\epsilon}(x)$ 的零点，对小量 ϵ 的敏感性.

(1) 分别取 $\epsilon=10^{-8},10^{-9},10^{-10}$，利用 np.roots 函数计算 $p(x)$ 和 $p^{\epsilon}(x)$ 的零点;

(2) 将这些零点会在同一张图上 (注意: 图中 x-轴为实轴,y-轴为虚轴).

3. 数值计算与符号计算

实验目的: 感受数值计算与符号计算在计算速度上的差别

实验内容: 利用 solve 函数, 采用符号计算的方式完成上一题中的实验内容