

一、修改代码，编写测试用例

最终运行结果如下图所示（A，B的矩阵结果已经包含在main代码块当中）：

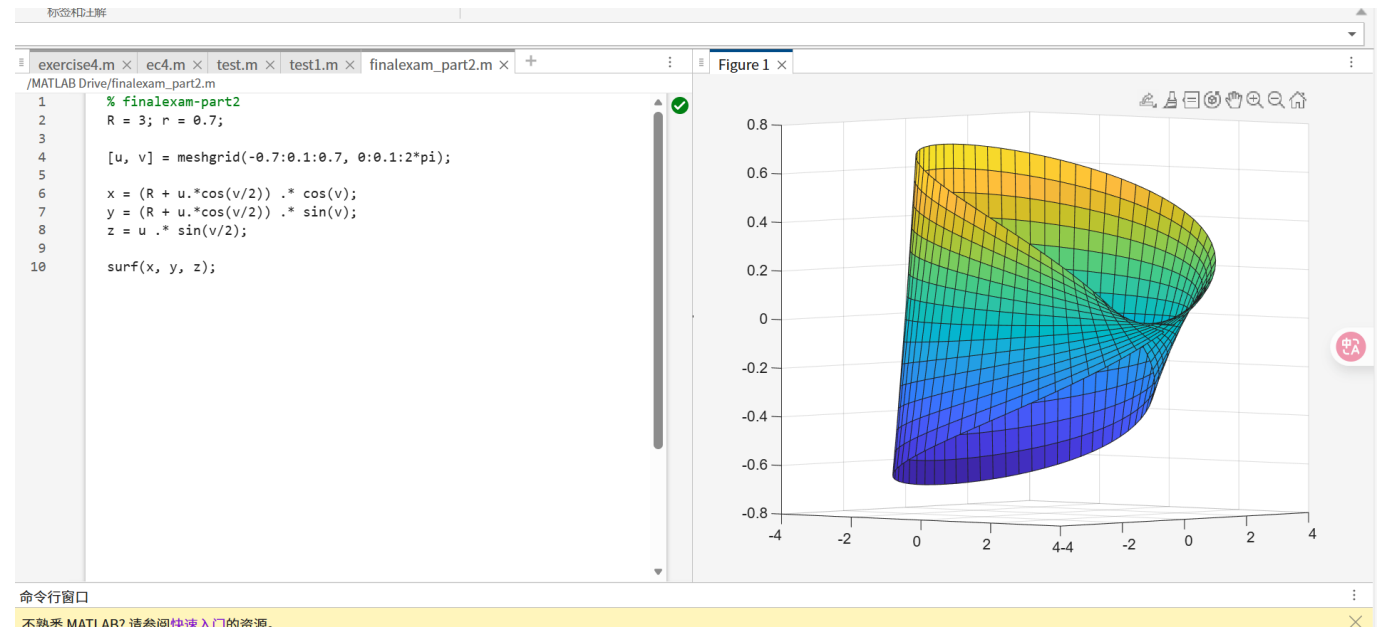
```
求A
{'max': np.int64(6), 'min': np.int64(1), 'mean': np.float64(3.5), 'std': np.float64(1.707825127659933), 'sr': np.float64(2.04939015319192), 'sum': np.int64(21)}
求B
{'max': np.int64(1), 'min': np.int64(-1), 'mean': np.float64(0.0), 'std': np.float64(0.7071067811865476), 'sr': np.float64(0.0), 'sum': np.int64(0)}
测试1: 正常3x3矩阵
结果: {'max': np.int64(9), 'min': np.int64(1), 'mean': np.float64(5.0), 'std': np.float64(2.581988897471611), 'sr': np.float64(1.9364916731037085), 'sum': np.int64(45)}
测试2: 单元素矩阵
结果: {'max': np.int64(5), 'min': np.int64(5), 'mean': np.float64(5.0), 'std': np.float64(0.0), 'sr': inf, 'sum': np.int64(5)}
测试3: 空矩阵
结果: {'max': nan, 'min': nan, 'mean': nan, 'std': nan, 'sr': nan, 'sum': 0}
测试4: 包含负数的矩阵
结果: {'max': np.int64(1), 'min': np.int64(-2), 'mean': np.float64(-0.5), 'std': np.float64(1.118033988749895), 'sr': np.float64(-0.4472135954999579), 'sum': np.int64(-2)}
测试5: 常数矩阵
结果: {'max': np.int64(3), 'min': np.int64(3), 'mean': np.float64(3.0), 'std': np.float64(0.0), 'sr': inf, 'sum': np.int64(12)}
```

修改问题如下：

- 1. 当输入矩阵为空时，需要专门处理
- 2. 原代码中有除法内容，需要保证除数不为0

二、Matlab

运行结果：



代码见本文件夹下的文件finalexam_part2.m

三、mathematic

我利用WOLFRAM CLOUD在线网站完成了这个，但是网站上下载下来的文件似乎看起来不太美观（详见本文件夹下mathematic.nb文件）

1

代码：

```
NSum[Cos[Pi/n]/n^3,{n,1,Infinity}]
```

```
In[1]:= Sum[Cos[Pi / n] / n^3, {n, 1, Infinity}]
```

$$\text{Out[1]} = \sum_{n=1}^{\infty} \frac{\cos\left[\frac{\pi}{n}\right]}{n^3}$$

无法得到具体的结果，用NSum：

```
In[3]:= NSum[Cos[Pi / n] / n^3, {n, 1, Infinity}]
```

```
Out[3]= -0.948855
```

2

代码：

```
NIntegrate[Sin[x]/(x*(E^x+1)),{x,0,Infinity}]
```

```
In[2]:= Integrate[Sin[x] / (x * (E^x + 1)), {x, 0,
Infinity}]
```

$$\text{Out[2]} = \int_0^{\infty} \frac{\sin[x]}{(1 + e^x) x} dx$$

同样

无法得到具体的结果，用NIntegrate：

```
In[4]:= NIntegrate[Sin[x] / (x * (E^x + 1)), {x, 0,
Infinity}]
```

```
Out[4]= 0.506671
```



四、4.1markdown

Linear Least Squares

Linear least squares (LLS) is the least squares approximation of linear functions to data. It is a set of formulations for solving statistical problems involved in linear regression, including variants for ordinary (unweighted), weighted, and generalized (correlated) residuals. Numerical methods for linear least squares include inverting the matrix of the normal equations and orthogonal decomposition methods.

Basic Formulation

Consider the linear equation $\mathbf{A}\mathbf{x} = \mathbf{b}$

where $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{b} \in \mathbb{R}^m$ are given and $x \in \mathbb{R}^n$ is variable to be computed. When $m > n$, it is generally the case that Eq. (1) has no solution. For example, there is no value of x that satisfies $\begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix} x = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

because the first two rows require that $x = (1, 1)$, but then the third row is not satisfied. Thus, for $m > n$, the goal of solving Eq. (1) exactly is typically replaced by finding the value of x that minimizes some error. There are many ways that the error can be defined, but one of the most common is to define it as $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$. This produces a minimization problem, called a least squares problem $\min_{x \in \mathbb{R}^n} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$. The solution to the least squares problem is computed by solving the normal equation $\mathbf{A}^T \mathbf{A} \mathbf{x} = \mathbf{A}^T \mathbf{b}$

where \mathbf{A}^T denotes the transpose of the matrix \mathbf{A} .

四、4.2预览效果

The screenshot shows a Jupyter Notebook with two cells. The first cell is a code cell containing the following text:

```

30 # Linear Least Squares
31
32 statistical problems involved in linear regression, including
33 variants for ordinary (unweighted), weighted, and generalized
34 (correlated) residuals. Numerical methods for linear least squares
35 include inverting the matrix of the normal equations and orthogonal
36 decomposition methods.
37
38 ## Basic Formulation
39 Consider the linear equation
40
41  $\mathbf{A}\mathbf{x} = \mathbf{b}$ 
42
43 where  $\mathbf{A} \in \mathbb{R}^{m \times n}$  and  $\mathbf{b} \in \mathbb{R}^m$  are given and  $x \in \mathbb{R}^n$  is
44 variable to be computed. When  $m > n$ , it is generally the case that
45 Eq. (1) has no solution. For example, there is no value of  $x$  that
46 satisfies
47
48  $\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} x = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ 
49
50 because the first two rows require that  $x = (1, 1)$ , but then the
51 third row is not satisfied. Thus, for  $m > n$ , the goal of solving
52 Eq. (1) exactly is typically replaced by finding the value of  $x$ 
53 that minimizes some error. There are many ways that the error can be
54 defined, but one of the most common is to define it as  $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$ . This produces a minimization problem,
55 called a least squares problem
56
57  $\min_{x \in \mathbb{R}^n} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$ 
58
59 The solution to the least squares problem is computed by solving the
60 normal equation
61
62  $\mathbf{A}^T \mathbf{A} \mathbf{x} = \mathbf{A}^T \mathbf{b}$ 
63
64 where  $\mathbf{A}^T$  denotes the transpose of the matrix  $\mathbf{A}$ .

```

The second cell is a preview of the rendered HTML, which includes the title "Linear Least Squares", a description of the problem, the basic formulation, and the normal equation. The preview shows the rendered LaTeX equations and the text of the notebook cells.

但在我的电脑当中没有安装latex，也没有配置相关环境，可能渲染的pdf有点不尽如人意。