

21307130453 蒋瑜贤 期末 Part 2

- [1. `find.py`](#)
 - [1. 为`find_name_value`添加功能描述](#)
 - [2. 设计测试样例](#)
 - [修改代码](#)
 - [返回第一个pair](#)
 - [返回所有pair](#)
 - [结果](#)
- [2. Matlab 作图](#)
- [3. `Mathematica` 求解](#)
 - [1.](#)
 - [2.](#)
- [4. Markdown写作](#)

1. find.py

1. 为 find_name_value 添加功能描述

函数 `find_name_value` 用于将数据目录的名称拆分为一个 (name, value) 元组。目录名称的格式为 `<name><value>`，其中 `<value>` 可以是正数或负数。如果值是负数，则应在其后跟一个 'n'。例如，'phi0.1' 应返回 ('phi', 0.1)，'xN14.2' 应返回 ('xN', 14.2)，'kappa0.5n' 应返回 ('kappa', -0.5)。

2. 设计测试样例

- 正常输入：此时输入的字符串符合上述格式，如 'phi0.1', 'xN14.2', 'kappa0.5n'。

输入	预期输出	实际输出	测试是否通过
'phi0.1'	('phi', 0.1)	('phi', 0.1)	是
'xN14.2'	('xN', 14.2)	('xN', 14.2)	是
'kappa0.5n'	('kappa', -0.5)	('kappa', -0.5)	是
'beta0.2n'	('beta', -0.2)	('beta', -0.2)	是

输入	预期输出	实际输出	测试是否通过
"phi0.1n_xN14.2_kappa0.5n"	('phi', -0.1)	('phi', 0.1)	否

错误一：由于函数只判断第三个匹配到的字符是否为 `n`，若 `n` 后面还有字符，这个 `n` 就无法判断成功。

- 异常输入：此时输入的字符串不符合上述格式，如 `'gemma'`，`'phi0.1x'`

输入	预期输出	实际输出	测试是否通过
'gemma'	('gemma', None)	('gemma', None)	是
'alpha-0.3n'	('alpha-', '-0.3')	ValueError	否
'alpha-0.3_phi0.1x'	('alpha-', '0.3')	('alpha', -0.3)	否
'phi0.1x'	('phi', 0.1)	('phi', 0.1)	是

错误二：由于函数没有处理负号的情况，所以当输入字符串中的数字为负数时，应该把负数作为名称的一部分，而不是作为数值。

错误三：由于函数没有处理字符串中包含多个 `pair` 的情况，这里题目中没有说明如何处理，可以返回所有的 `pair`，也可以返回第一个 `pair`。

- 边界输入：

输入	预期输出	实际输出	测试是否通过
"	("", None)	("", None)	是
'23'	("", 23.0)	("", 23.0)	是
'phi'	('phi', None)	('phi', None)	是
'phi0.0n'	('phi', -0.0)	('phi', -0.0)	是

修改代码

由于题目中没有写明如何处理多个 `pair` 的情况，这里提供两个版本，一个返回第一个 `pair`，一个返回所有的 `pair`。

返回第一个 `pair`

```
def find_name_value(folder_name):  
    '''Split the name of a data directory into a (name, value)  
tuple.
```

The format of ``folder_name``:

<name><value>

If the value is negative, it should be followed by a 'n'.

Examples:

::

phi0.1	# should return 'phi', 0.1
xN14.2	# should return 'xN', 14.2
kappa0.5n	# should return 'kappa', -0.5

Args:

folder_name (str): the name of a :term:`data directory`.

Returns:

tuple: a tuple contains:

- * name (str): variable name.
- * value (float): value of the variable.

...

```
pattern = '([-+]?\d*\.\d+|[-+]?\d+)'
```

```
rst = re.split(pattern, folder_name)
```

```
if len(rst) < 2:
```

```
    return folder_name, None
```

```
name = rst[0]
```

```
valuestr = rst[1]
```

```
# handle negative value
```

```
if valuestr[0] == '-':
```

```
    value = valuestr[1:]
```

```
    name = name + '-'
```

```
if len(rst) == 3
```

```
sign_str = ''
```

```
if len(rst) > 2:
```

```
    sign_str = rst[2]
```

```
if sign_str[0] == 'n':
```

```
    value = '-' + valuestr
```

```

else:
    value = valustr

return name, float(value)

```

返回所有pair

```

import re

def find_name_value(folder_name):
    '''Split the name of a data directory into a list of (name, value)
    tuples.

    The format of ``folder_name``:

        <name><value>

    If the value is negative, it should be followed by a 'n'.

    Examples:
        ::

            phi0.1xN14.2kappa0.5n # should return [('phi', 0.1), ('xN',
            14.2), ('kappa', -0.5)]

    Args:
        folder_name (str): the name of a :term:`data directory`.

    Returns:
        list: a list of tuples, each containing:

        * name (str): variable name.
        * value (float): value of the variable.
    '''
    pattern = r'([a-zA-Z]+)(-?\d*\.\d+|-?\d+)(n?)'
    matches = re.findall(pattern, folder_name)
    result = []
    for match in matches:
        name, value_str, sign = match
        # handle negative value
        if value_str[0] == '-':
            value_str = value_str[1:]

```

```
        name = name + '-'
    value = float(value_str)
    if sign == 'n':
        value = -value
    result.append((name, value))
return result
```

结果

若使用返回第一个pair的版本，结果为

```
('phi', 0.1)
('a', 1.0)
```

若使用返回所有pair的版本，结果为

```
[('phi', 0.1), ('xN', 14.2), ('kappa', -0.5)]
[('a', 1.0), ('b', -14.0), ('n', 0.0), ('c', 0.2)]
```

2. Matlab 作图

Matlab 代码

```
% 参数
R = 3;
r = 1;

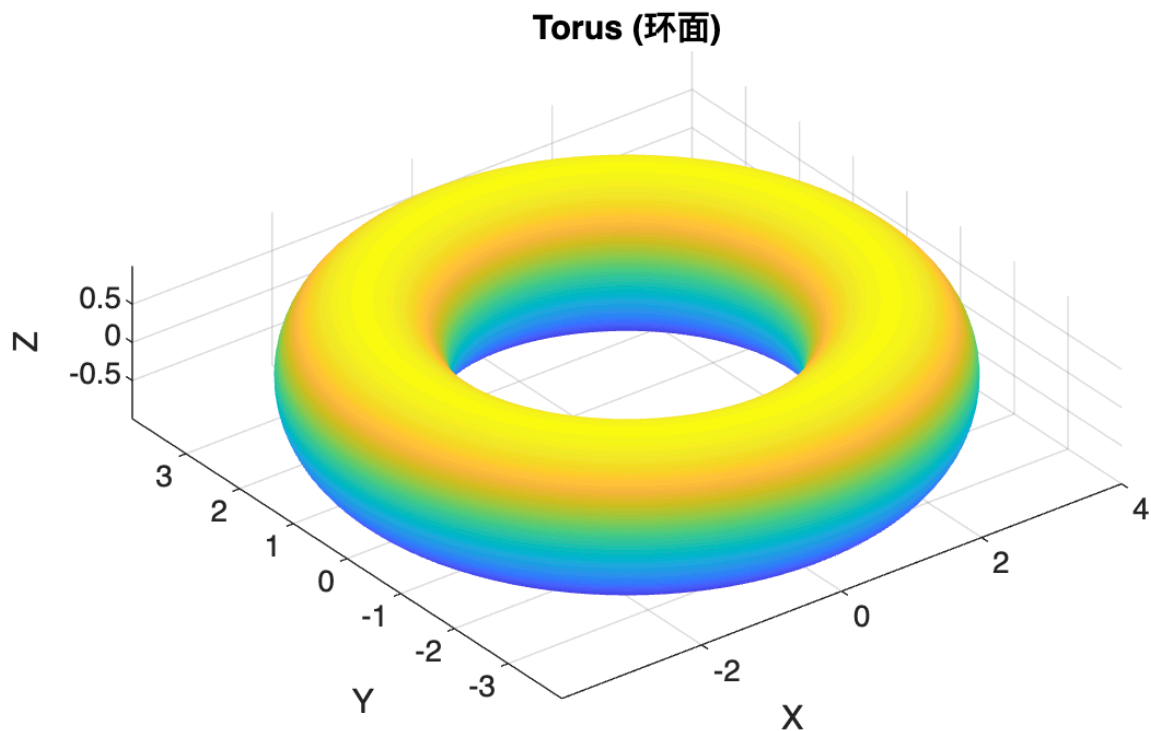
% 创建角度网格
theta = linspace(0, 2*pi, 100);
phi = linspace(0, 2*pi, 100);
[Theta, Phi] = meshgrid(theta, phi);

% 计算圆环的坐标
X = (R + r * cos(Theta)) .* cos(Phi);
Y = (R + r * cos(Theta)) .* sin(Phi);
Z = r * sin(Theta);
```

```
% 绘制圆环
surf(X, Y, Z)

% 设置图像属性
axis equal
xlabel('X')
ylabel('Y')
zlabel('Z')
title('Torus (环面)')
shading interp
```

运行结果：



3. Mathematica 求解

1.

代码为：

```
Sum[1/(n^3 + n^2), {n, 1, Infinity}]
```

```
In[3]:= Sum[1 / (n^3 + n^2), {n, 1, Infinity}]
```

```
Out[3]= -1 +  $\frac{\pi^2}{6}$ 
```

2.

代码为:

```
Integrate[Sqrt[x] Log[x]/(x + 1)^2, {x, 0, Infinity}]
```

```
In[4]:= Integrate[Sqrt[x] × Log[x] / (x + 1)^2, {x, 0, Infinity}]
```

```
Out[4]=  $\pi$ 
```

4. Markdown写作

Q: Find the solution of the following equation with respect to θ :

$$A \cos \theta + B \sin \theta + C = 0$$

A:

let $x_1 = \cos \theta$ and $x_2 = \sin \theta$, then the solution is given by the intersection of the circle and the line:

$$x_1^2 + x_2^2 = 1$$

$$Ax_1 + Bx_2 + C = 0$$

We reformulate the equations in a parametric form:

$$|x|^2 = 1$$

$$x(t) = a + tb$$

where $\mathbf{x} = (x_1, x_2)$, $\mathbf{a} = (0, -C/B)$, $\mathbf{b} = (-C/A, C/B)$, and t is a parameter. The intersection points satisfy the following equation:

$$|\mathbf{a} + t\mathbf{b}|^2 = 1$$

which can be solved for t to find the intersection points:

$$t_{1,2} = \frac{-\mathbf{a} \cdot \mathbf{b} \pm \sqrt{(\mathbf{a} \cdot \mathbf{b})^2 - |\mathbf{b}|^2(|\mathbf{a}|^2 - 1)}}{|\mathbf{b}|^2}$$