

第一题

函数名称: `find_name_value`

功能描述: 该函数用于解析数据目录的名称，并将其拆分为变量名和数值的元组。它接受一个字符串形式的目录名称，该名称遵循特定的格式，即变量名后紧跟一个数值，数值可以是正数或负数。如果数值为负数，它后面应当跟随一个字母'n'。

参数说明:

- `folder_name` (str) : 需要解析的数据目录名称。

返回值:

- 返回一个元组，包含两个元素：
 - `name` (str) : 变量的名称。
 - `value` (float) : 变量的数值。

工作流程:

1. 使用正则表达式 `([-+]?\\d*\\.\\d+|[-+]?\\d+)` 匹配目录名称中的数值部分。
2. 使用 `re.split` 函数根据匹配到的数值将目录名称分割成多个部分。
3. 如果分割后的结果少于两个部分，说明目录名称不符合预期格式，函数返回原始目录名称和 `None`。
4. 如果分割后的结果包含两个或更多部分，函数将第一部分视为变量名，第二部分视为数值字符串。
5. 如果分割后的结果包含第三部分，且第三部分为 'n'，则将数值字符串转换为负数。
6. 最后，函数返回包含变量名和数值的元组。

以下是为 `find_name_value` 函数设计的10个测试用例，包括正常输入、异常输入和边界输入，并给出预期的测试结果：

测试用例

1. 正常输入 - 正数

- 输入: `phi0.1`
- 预期输出: `('phi', 0.1)`

2. 正常输入 - 负数

- 输入: `kappa0.5n`
- 预期输出: `('kappa', -0.5)`

3. 正常输入 - 整数

- 输入: `xN14`
- 预期输出: `('xN', 14.0)`

4. 正常输入 - 负整数

- 输入: `xN14n`
- 预期输出: `('xN', -14.0)`

5. 正常输入 - 浮点数

- 输入: `xN14.7`
- 预期输出: `('xN', 14.7)`

6. 正常输入 - 负浮点数

- 输入: `xN14.7n`
- 预期输出: `('xN', -14.7)`

7. 异常输入 - 空字符串

- 输入: `''`
- 预期输出: `('', None)`

8. 异常输入 - 无数值

- 输入: `phi`
- 预期输出: `('phi', None)`

9. 边界输入 - 非常小的浮点数

- 输入: `xN0.0001`
- 预期输出: `('xN', 0.0001)`

10. 边界输入 - 非常大的浮点数

- 输入: `xN1000000.0`
- 预期输出: `('xN', 1000000.0)`

所有测试用例均通过, 表明`find_name_value`函数能够正确处理各种输入情况, 包括正常输入、异常输入和边界输入。

为了处理复合文件夹名称, 需要改进函数以支持多个变量名和值的解析。可以修改正则表达式来匹配多个变量和值, 并且调整代码逻辑以处理多个匹配结果。以下是改进后的函数:

```
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>
```

```
The value can be a positive or negative number and if negative, it should be
followed by an 'n'.
```

```
The function can handle multiple variables separated by underscores (_).
```

```
Examples:
```

```
::
```

```
        phi0.1_xN14.2_kappa0.5n          # should return [('phi', 0.1), ('xN', 14.2),
('kappa', -0.5)]
```

```

        a1_b14n_n0_c0.2                # should return [('a', 1.0), ('b', -14.0),
('n', 0.0), ('c', 0.2)]

Args:
    folder_name (str): The name of a data directory.

Returns:
    list: A list of tuples containing variable names and their corresponding
values.
...
pattern = r'([a-zA-Z]+)([-+]?\\d*\\.\\d+|[-+]?\\d+)(n)?'
matches = re.findall(pattern, folder_name)

result = []
for match in matches:
    name = match[0]
    value_str = match[1]
    sign_str = match[2] if len(match) > 2 else ''

    if sign_str == 'n':
        value = -float(value_str)
    else:
        value = float(value_str)

    result.append((name, value))

return result

```

测试用例

```
folder_name1 = "phi0.1_xN14.2_kappa0.5n" folder_name2 = "a1_b14n_n0_c0.2"
```

```
print(find_name_value(folder_name1)) # [('phi', 0.1), ('xN', 14.2), ('kappa', -0.5)]
```

```
print(find_name_value(folder_name2)) # [('a', 1.0), ('b', -14.0), ('n', 0.0), ('c', 0.2)]
```

第二题

```
% 定义参数 R = 3; % 环面中心到管中心的距离 r = 1; % 环面管的半径
```

```
% 定义角度范围 theta = linspace(0, 2pi, 100); % 角度θ从0到2π phi = linspace(0, 2pi, 100); % 角度φ从0到2π
```

```
% 创建网格 [THETA, PHI] = meshgrid(theta, phi);
```

```
% 计算环面的X, Y, Z坐标 X = (R + rcos(THETA)) . cos(PHI); Y = (R + rcos(THETA)) . sin(PHI); Z = r * sin(THETA);
```

```
% 使用surf函数渲染环面 figure; surf(X, Y, Z);
```

```
% 添加坐标轴标签 xlabel('X'); ylabel('Y'); zlabel('Z');
```

```
% 设置视图角度 view(3);
```

```
% 添加标题 title('Toroidal Surface with R=3 and r=1');

% 显示网格 grid on;

% 显示颜色条 colorbar;


! 图片
```

第三题

1.

Sum[1/(n^3 + n^2),{n, 1, Infinity}] 答案: -1 + pi^2/6

2.

 * ln x)/(x + 1)^2, {x, 0, Infinity}] 答案: ln x * pi/2

第四题

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:
 $|x|=1$ $x(t)=a+tb$ where $x = (x_1,x_2)$, $a = (0,-C/B)$, $b = (-C/A,C/B)$, and t is a parameter. The intersection points satisfy the following equation: $|a+tb|^2=1$ which can be solved for t to find the intersection points: $t_{1,2}=\frac{-a\cdot b\pm\sqrt{(a\cdot b)^2-(|b|^2(|a|^2-1))}}{|b|^2}$