

例 3.13 求函数的梯度和 Hessian 矩阵

修改参数：函数所含变量

f: 给定的函数

"""

```
import numpy as np
from sympy import *
```

```
def Hessian():
```

```
    # 设置变量
```

```
    x1 = symbols("x1")
```

```
    x2 = symbols("x2")
```

```
    x3 = symbols("x3")
```

```
    # 求偏导
```

```
    difyL_x1 = diff(f, x1)
```

```
    difyL_x2 = diff(f, x2)
```

```
    difyL_x3 = diff(f, x3)
```

```
    # 得梯度
```

```
    P = np.array([[difyL_x1],
                  [difyL_x2],
                  [difyL_x3]
                  ])
    print("该函数的梯度为: " '\n', P)
```

```
    # 计算 Hessian 矩阵
```

```
    difyL_x1_x1 = diff(difyL_x1, x1)
```

```
    difyL_x1_x2 = diff(difyL_x1, x2)
```

```
    difyL_x1_x3 = diff(difyL_x1, x3)
```

```
    difyL_x2_x2 = diff(difyL_x2, x2)
```

```
    difyL_x2_x3 = diff(difyL_x2, x3)
```

```
    difyL_x3_x3 = diff(difyL_x3, x3)
```

```
    H = np.array([[difyL_x1_x1, difyL_x1_x2, difyL_x1_x3],
                  [difyL_x1_x2, difyL_x2_x2, difyL_x2_x3],
                  [difyL_x1_x3, difyL_x2_x3, difyL_x3_x3]
                  ])
    print("该函数的 Hessian 矩阵为: " '\n', H)
```

```
if __name__ == '__main__':
```

```
    x1 = symbols("x1")
```

```
    x2 = symbols("x2")
```

```
x3 = symbols("x3")  
f = x1 ** 2 + x2 ** 2 + x3 ** 2 - 2 * x1 * x2 - 2 * x2 * x3 + 3 * x3  
Hessian()
```