

例 3.12 求函数的梯度

修改参数：函数所含变量

f: 给定的函数

"""

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

```
def gradient():
    # 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)
```

```
if __name__ == '__main__':
    x1 = symbols("x1")
    x2 = symbols("x2")
    x3 = symbols("x3")
    f = x1 ** 2 + x1 * x2 ** 2 + 3 * x3 ** 2 - 4 * x1 * x2 * x3
    gradient()
    f = 4 * x1 * x2 ** 3 + pow(sympy.E, 2 * x1 * x2)
    gradient()
```