问题1

1. 最速下降法（Steepest Descent）

- 梯度：

$$\nabla f(x) = \begin{bmatrix} 2x_1 + x_2 \\ x_1 + 4x_2 \end{bmatrix}$$

- 更新公式：

$$x^{(k+1)} = x^{(k)} - \alpha_k \nabla f(x^{(k)})$$

- 最终结果：

$$x^* \approx \begin{bmatrix} 0.0661 \\ -0.0078 \end{bmatrix}$$

---

2. 牛顿法（Newton's Method）

- Hessian：

$$H = \nabla^2 f(x) = \begin{bmatrix} 2 & 1 \\ 1 & 4 \end{bmatrix}$$

- 初始梯度：

$$g^{(0)} = \begin{bmatrix} 2(2) + (-2) \\ 2 + 4(-2) \end{bmatrix} = \begin{bmatrix} 2 \\ -6 \end{bmatrix}$$

- 解线性方程 \$Hd = -g\$，得：

$$d = -H^{-1} g = \begin{bmatrix} -2 \\ 2 \end{bmatrix}$$

- 一步迭代得最优解：

$$x^* = \begin{bmatrix} 2 \\ -2 \end{bmatrix} + \begin{bmatrix} -2 \\ 2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

---

3. 共轭梯度法（Conjugate Gradient Method）

- 目标函数为二次型，可写成：

$$f(x) = \frac{1}{2} x^T Q x \quad \text{其中 } Q = \begin{bmatrix} 2 & 1 \\ 1 & 4 \end{bmatrix}$$

- 共轭梯度法对二次函数在 \$n\$ 步内收敛（\$n=2\$），最优解为：

$$x^* = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

---

4. 拟牛顿法（DFP）

- 初始设 $H_0 = I$
- 每次迭代更新：

$$H_{k+1} = H_k + \frac{s_k s_k^T}{s_k^T y_k} - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k}$$

  其中 $s_k = x^{(k+1)} - x^{(k)}$, $y_k = \nabla f(x^{(k+1)}) - \nabla f(x^{(k)})$

- 由于目标函数为严格凸的二次函数，DFP 方法也能较快收敛至最优解：

$$x^* = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

问题二

Python 实现：最速下降法

```
import numpy as np

def f(x):
    return x[0]**2 + 2 * x[1]**2 + x[0]*x[1]

def grad_f(x):
    return np.array([2*x[0] + x[1], x[0] + 4*x[1]])

def exact_line_search(f, x, d):
    alpha_vals = np.linspace(0, 1, 100)
    f_vals = [f(x + alpha*d) for alpha in alpha_vals]
    return alpha_vals[np.argmin(f_vals)]

def steepest_descent(x0, epsilon=0.2, max_iter=100):
    x = x0.copy()
    for i in range(max_iter):
        g = grad_f(x)
        if np.linalg.norm(g) < epsilon:
            break
        d = -g
        alpha = exact_line_search(f, x, d)
        x = x + alpha * d
    return x

x0 = np.array([2.0, -2.0])
x_star = steepest_descent(x0)
print("最优解为：", x_star)
```

问题 3

1. 模式搜索法

初始点：$x^{(0)} = [1, 1]^T$，$f(x^{(0)}) = 1^2 + 1^2 = 2$

第一次迭代：

沿 $d_1 = [1, 0]$，$x = [1 + \alpha, 1]$，$f = (1 + \alpha)^2 + 1$，最小化得 $\alpha = -1$，$x = [0, 1]$，$f = 1$

沿 $d_2 = [0, 1]$，$x = [0, 1 + \beta]$，$f = \beta^2$，最小化得 $\beta = -1$，$x^{(1)} = [0, 0]$

$\nabla f(x^{(1)}) = [0, 0]$，$||\nabla f(x^{(1)})|| = 0$

结果：$x^* = [0, 0]^T$，$f(x^*) = 0$（一步收敛）

2. Rosenbrock 方法

初始点：$x^{(0)} = [1, 1]^T$，方向 $d_1 = [1, 0]$，$d_2 = [0, 1]$

第一次迭代：

沿 d_1，$x = [1 + \alpha, 1]$，$f = (1 + \alpha)^2 + 1$，最小化得 $\alpha = -1$，$x = [0, 1]$

沿 d_2，$x = [0, 1 + \beta]$，$f = \beta^2$，最小化得 $\beta = -1$，$x^{(1)} = [0, 0]$

$\nabla f(x^{(1)}) = [0, 0]$，$||\nabla f(x^{(1)})|| = 0$

结果：$x^* = [0, 0]^T$，$f(x^*) = 0$（一步收敛）

3. Powell 方法

初始点：$x^{(0)} = [1, 1]^T$

第一次迭代：

沿 $d_1 = [1, 0]$，$x = [1 + \alpha, 1]$，$f = (1 + \alpha)^2 + 1$，最小化得 $\alpha = -1$，$x = [0, 1]$

沿 $d_2 = [0, 1]$，$x = [0, 1 + \beta]$，$f = \beta^2$，最小化得 $\beta = -1$，$x^{(1)} = [0, 0]$

$\nabla f(x^{(1)}) = [0, 0]$，$||\nabla f(x^{(1)})|| = 0$

结果：$x^* = [0, 0]^T$，$f(x^*) = 0$（一步收敛）