优化方法第二次作业报告

3230104932 吴官旺

2025 年 11 月 7 日

1 第一题：用 CG 算法优化希尔伯特系统

1.1 算法

CG 算法实现如下：

```
def conjugate_gradient_method(A, b, x0=None, tol=1e-5, max_iter=None, save_full_data=False, filename="conjugate_gradient_results.txt"):
    data_dir = Path("data")
    data_dir.mkdir(exist_ok=True)
    filepath = data_dir / filename
    n = len(b)
    if x0 is None:
        x0 = np.zeros(n)
    else:
        x = x0.copy()
    r = A @ x - b
    p = -r.copy()
    rsold = r @ r
    if max_iter is None:
        max_iter = n+1
    with open(filepath, 'w') as file:
        if save_full_data:
            file.write("iteration,residual_norm,x\n")
        else:
            file.write("iteration,residual_norm\n")
        for i in range(max_iter):
            Ap = A @ p
            alpha = rsold / (p @ Ap)
            x += alpha * p
            r += alpha * Ap
            rsnew = r @ r
            if save_full_data:
                file.write(f"{i},{np.sqrt(rsnew)},{x.tolist()}\n")
            else:
                file.write(f"{i},{np.sqrt(rsnew)}\n")
            if np.sqrt(rsnew) < tol:
                return x, i+1
            beta = rsnew / rsold
```

```
            p = -r + beta * p
            rsold = rsnew
    return x, max_iter
```

1.2 实验结果

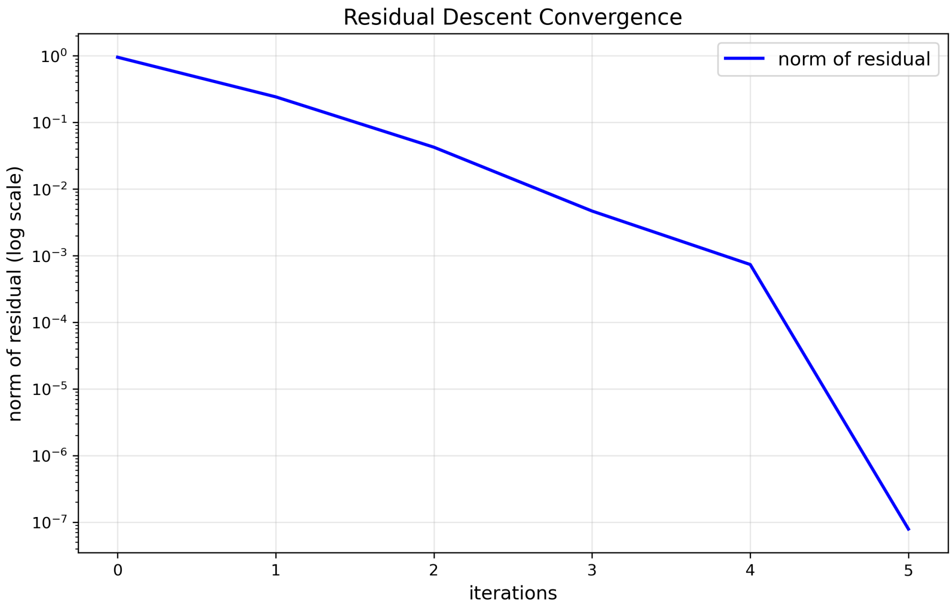


图 1: 当 n=5 时的残差收敛图

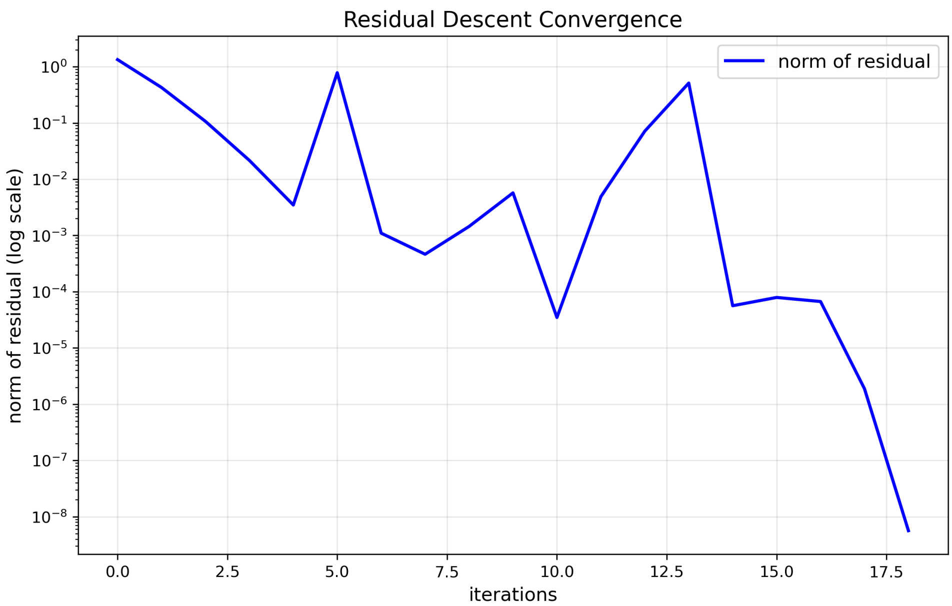


图 2: 当 n=8 时的残差收敛图

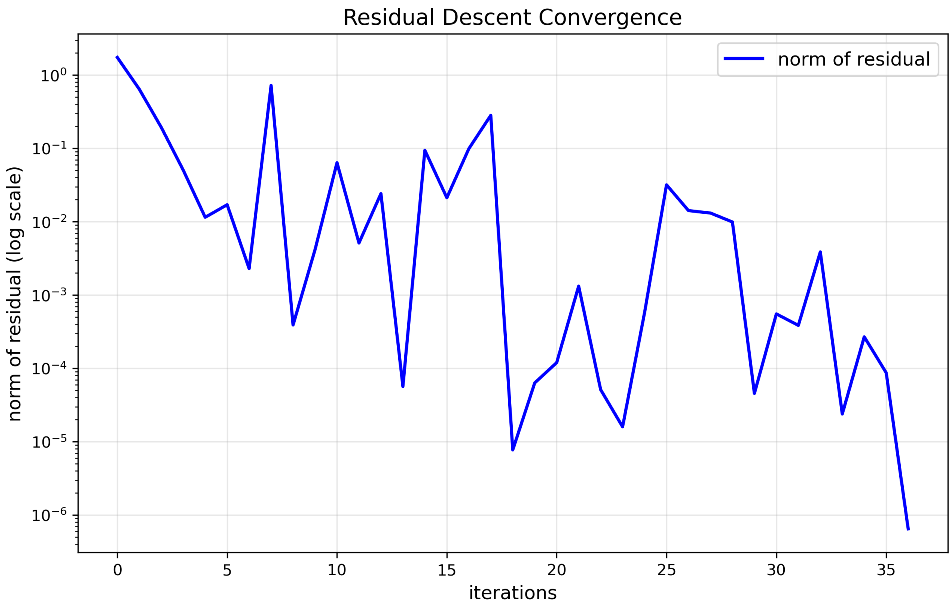


图 3: 当 n=12 时的残差收敛图

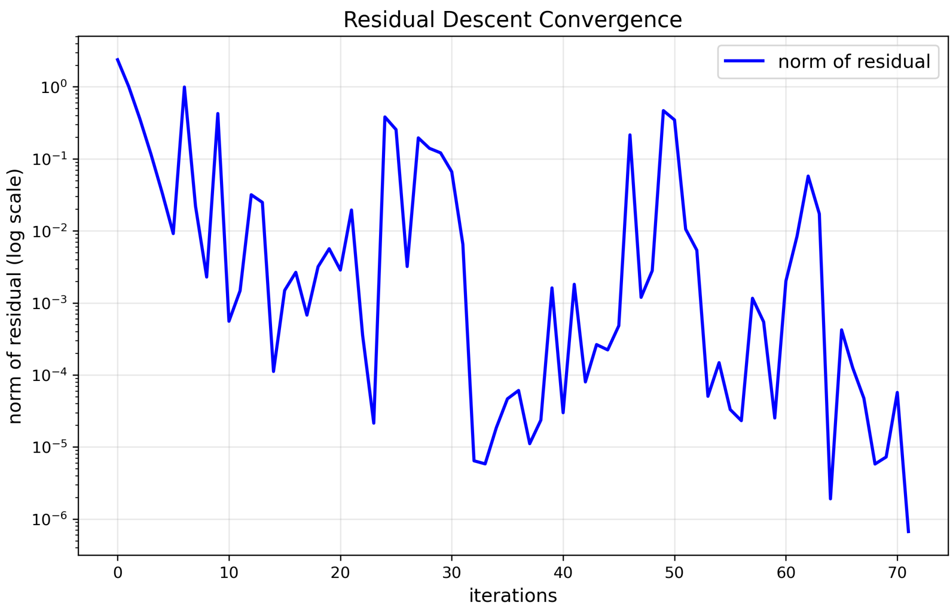


图 4: 当 n=20 时的残差收敛图

1.3 实验分析

可以看出随着 n 的增大，残差的锯齿效应越明显，说明 CG 方法随着 n 增大收敛会变慢，但仍然在稳定收敛，说明 CG 方法稳定性好。

2 第二题：预处理共轭梯度法

证明. 令 $M = LL^T$，其中 L 为下三角矩阵，令 $\hat{x} = L^T x$，则

$$\hat{A}\hat{x} = \hat{b}, \quad \hat{A} = L^{-1}AL^{-T}, \quad \hat{b} = L^{-1}b$$

对 $\hat{A}\hat{x} = \hat{b}$ 应用 CG 算法得:

$$\hat{x}_0 = L^T x_0, \quad \hat{r}_0 = \hat{b} - \hat{A}\hat{x}_0, \quad \hat{p}_0 = \hat{r}_0$$

对于 $k = 0, 1, 2, \ldots$，执行:

(1) $\hat{\alpha}_k = \dfrac{\hat{r}_k^T \hat{r}_k}{\hat{p}_k^T \hat{A} \hat{p}_k}$

(2) $\hat{x}_{k+1} = \hat{x}_k + \hat{\alpha}_k \hat{p}_k$

(3) $\hat{r}_{k+1} = \hat{r}_k - \hat{\alpha}_k \hat{A} \hat{p}_k$

(4) $\hat{\beta}_k = \dfrac{\hat{r}_{k+1}^T \hat{r}_{k+1}}{\hat{r}_k^T \hat{r}_k}$

(5) $\hat{p}_{k+1} = \hat{r}_{k+1} + \hat{\beta}_k \hat{p}_k$

转换回原始变量:

$$\hat{x} = L^T x \Rightarrow x = L^{-T}\hat{x}$$

$$\hat{r} = L^{-1} r \Rightarrow r = L\hat{r}$$

$$p_k = L^{-T}\hat{p}_k \Rightarrow \hat{p}_k = L^T p_k$$

$$\begin{aligned}
\hat{\alpha}_k &= \frac{\hat{r}_k^T \hat{r}_k}{\hat{p}_k^T \hat{A} \hat{p}_k} \\
&= \frac{(L^{-1} r_k)^T (L^{-1} r_k)}{(L^T p_k)^T (L^{-1} A L^{-T})(L^T p_k)} \\
&= \frac{r_k^T L^{-T} L^{-1} r_k}{p_k^T L L^{-1} A L^{-T} L^T p_k} \\
&= \frac{r_k^T M^{-1} r_k}{p_k^T A p_k}
\end{aligned}$$

$$\begin{aligned}
\hat{\beta}_k &= \frac{\hat{r}_{k+1}^T \hat{r}_{k+1}}{\hat{r}_k^T \hat{r}_k} \\
&= \frac{(L^{-1} r_{k+1})^T (L^{-1} r_{k+1})}{(L^{-1} r_k)^T (L^{-1} r_k)} \\
&= \frac{r_{k+1}^T M^{-1} r_{k+1}}{r_k^T M^{-1} r_k}
\end{aligned}$$

$$\hat{p}_{k+1} = \hat{r}_{k+1} + \hat{\beta}_k \hat{p}_k$$

左乘 L^{-T}:

$$L^{-T}\hat{p}_{k+1} = L^{-T}\hat{r}_{k+1} + \hat{\beta}_k L^{-T}\hat{p}_k$$

$$p_{k+1} = M^{-1} r_{k+1} + \hat{\beta}_k p_k$$

$\square$

3 第三题证明：Broyden 类更新矩阵的奇异性

问题描述

考虑 Broyden 类拟牛顿更新公式：

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k} + \phi_k (s_k^T B_k s_k) v_k v_k^T$$

其中

$$v_k = \left(\frac{y_k}{y_k^T s_k} - \frac{B_k s_k}{s_k^T B_k s_k} \right)$$

且

$$\mu_k = \frac{(s_k^T B_k s_k)(y_k^T H_k y_k)}{(s_k^T y_k)^2}, \quad H_k = B_k^{-1}$$

当 $\phi = \phi_k^c = \frac{1}{1-\mu_k}$ 时，证明 B_{k+1} 是奇异矩阵。

为简化记号，定义：

$$a = s_k^T B_k s_k, \quad b = s_k^T y_k, \quad d = y_k^T H_k y_k$$

则

$$\mu_k = \frac{ad}{b^2}, \quad \phi_k^c = \frac{1}{1-\mu_k} = \frac{b^2}{b^2 - ad}$$

定义向量：

$$z = H_k y_k - \frac{d}{b} s_k$$

将 Broyden 类更新公式应用于 $H_k y_k$：

$$\begin{aligned} B_{k+1} H_k y_k &= \left[B_k - \frac{B_k s_k s_k^T B_k}{a} + \frac{y_k y_k^T}{b} + \phi_k a v_k v_k^T \right] H_k y_k \\ &= B_k H_k y_k - \frac{B_k s_k s_k^T B_k}{a} H_k y_k + \frac{y_k y_k^T}{b} H_k y_k + \phi_k a v_k v_k^T H_k y_k \end{aligned}$$

逐项计算：

(1) $B_k H_k y_k = y_k$

(2) $\frac{B_k s_k s_k^T B_k}{a} H_k y_k = \frac{B_k s_k}{a} (s_k^T B_k H_k y_k) = \frac{B_k s_k}{a} (s_k^T y_k) = \frac{b}{a} B_k s_k$

(3) $\frac{y_k y_k^T}{b} H_k y_k = \frac{y_k}{b} (y_k^T H_k y_k) = \frac{d}{b} y_k$

(4) 计算 $v_k^T H_k y_k$：

$$\begin{aligned} v_k^T H_k y_k &= \left(\frac{y_k}{b} - \frac{B_k s_k}{a} \right)^T H_k y_k \\ &= \frac{y_k^T H_k y_k}{b} - \frac{s_k^T B_k H_k y_k}{a} \\ &= \frac{d}{b} - \frac{s_k^T y_k}{a} = \frac{d}{b} - \frac{b}{a} \end{aligned}$$

因此：

$$\phi_k a v_k v_k^T H_k y_k = \phi_k a v_k \left(\frac{d}{b} - \frac{b}{a} \right)$$

代入 $v_k = \frac{y_k}{b} - \frac{B_k s_k}{a}$：

$$\begin{aligned} B_{k+1} H_k y_k &= y_k - \frac{b}{a} B_k s_k + \frac{d}{b} y_k + \phi_k a \left(\frac{y_k}{b} - \frac{B_k s_k}{a} \right) \left(\frac{d}{b} - \frac{b}{a} \right) \\ &= y_k \left(1 + \frac{d}{b} + \phi_k \frac{a}{b} \left(\frac{d}{b} - \frac{b}{a} \right) \right) + B_k s_k \left(-\frac{b}{a} - \phi_k \left(\frac{d}{b} - \frac{b}{a} \right) \right) \end{aligned}$$

简化系数:

y_k 的系数:

$$1+\frac{d}{b}+\phi_k\left(\frac{ad}{b^2}-1\right)=1+\frac{d}{b}-\phi_k+\phi_k\frac{ad}{b^2}$$

$B_k s_k$ 的系数:

$$-\frac{b}{a}-\phi_k\left(\frac{d}{b}-\frac{b}{a}\right)=-\frac{b}{a}(1-\phi_k)-\phi_k\frac{d}{b}$$

因此:

$$B_{k+1}H_k y_k=y_k\left(1+\frac{d}{b}-\phi_k+\phi_k\frac{ad}{b^2}\right)+B_k s_k\left(-\frac{b}{a}(1-\phi_k)-\phi_k\frac{d}{b}\right) \tag{1}$$

由拟牛顿条件，$B_{k+1}s_k=y_k$（与 ϕ_k 无关），因此:

$$\begin{aligned}B_{k+1}z&=B_{k+1}\left(H_k y_k-\frac{d}{b}s_k\right)\\&=B_{k+1}H_k y_k-\frac{d}{b}B_{k+1}s_k\\&=B_{k+1}H_k y_k-\frac{d}{b}y_k\end{aligned}$$

代入式 (1):

$$\begin{aligned}B_{k+1}z&=y_k\left(1+\frac{d}{b}-\phi_k+\phi_k\frac{ad}{b^2}-\frac{d}{b}\right)+B_k s_k\left(-\frac{b}{a}(1-\phi_k)-\phi_k\frac{d}{b}\right)\\&=y_k\left(1-\phi_k+\phi_k\frac{ad}{b^2}\right)+B_k s_k\left(-\frac{b}{a}(1-\phi_k)-\phi_k\frac{d}{b}\right)\end{aligned}$$

令 $\phi_k=\phi_k^c=\frac{b^2}{b^2-ad}$，计算各项系数:

(1) y_k 的系数:

$$\begin{aligned}1-\phi_k+\phi_k\frac{ad}{b^2}&=1-\phi_k\left(1-\frac{ad}{b^2}\right)\\&=1-\frac{b^2}{b^2-ad}\left(1-\frac{ad}{b^2}\right)\\&=1-\frac{b^2}{b^2-ad}\cdot\frac{b^2-ad}{b^2}=0\end{aligned}$$

(2) $B_k s_k$ 的系数:

$$\begin{aligned}&-\frac{b}{a}(1-\phi_k)-\phi_k\frac{d}{b}\\&=-\frac{b}{a}\left(1-\frac{b^2}{b^2-ad}\right)-\frac{b^2}{b^2-ad}\cdot\frac{d}{b}\\&=-\frac{b}{a}\cdot\frac{-ad}{b^2-ad}-\frac{bd}{b^2-ad}\\&=\frac{bd}{b^2-ad}-\frac{bd}{b^2-ad}=0\end{aligned}$$

因此，当 $\phi_k=\phi_k^c$ 时，$B_{k+1}z=0$。

由于 z 是非零向量（一般情况下 $H_k y_k$ 与 s_k 线性无关），故 B_{k+1} 是奇异矩阵。

当 $\phi=\phi_k^c=\frac{1}{1-\mu_k}$ 时, Broyden 类更新矩阵 B_{k+1} 是奇异矩阵, 这由非零向量 $z=H_k y_k-\frac{y_k^T H_k y_k}{y_k^T s_k}s_k$ 满足 $B_{k+1}z=0$ 所保证。

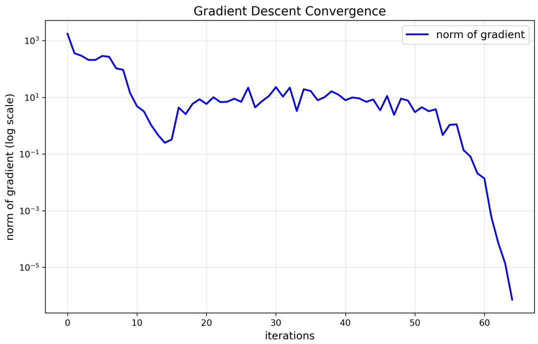
4 第四题：用 BFGS 算法优化 Rosenbrock 函数与 Powell 奇异函数

BFGS 算法如下：

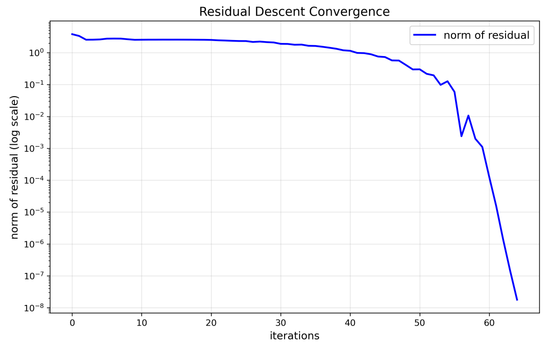
```python
def BFGS_method(f, grad_f, x0, tol=1e-6, max_iter=1000, save_full_data=False,
         filename="BFGS_results.txt"):
    data_dir = Path("data")
    data_dir.mkdir(exist_ok=True)
    filepath = data_dir / filename

    n = len(x0)
    x = x0.copy()
    H = np.eye(n)

    with open(filepath, 'w') as file:
        if save_full_data:
            file.write("iteration,gradient_norm,x\n")
        else:
            file.write("iteration,gradient_norm\n")

        for i in range(max_iter):
            grad = grad_f(x)
            norm_grad = np.linalg.norm(grad)

            if save_full_data:
                file.write(f"{i},{norm_grad},{x.tolist()}\n")
            else:
                file.write(f"{i},{norm_grad}\n")

            if norm_grad < tol:
                return x, i

            direction = -H @ grad

            def f_alpha(alpha):
                return f(x + alpha * direction)

            def f_deriv_alpha(alpha):
                return np.dot(grad_f(x + alpha * direction), direction)

            alpha_opt = wolfe_search(f_alpha, f_deriv_alpha)
            s = alpha_opt * direction
            x_new = x + s
            y = grad_f(x_new) - grad

            rho = 1.0 / (y @ s)
            I = np.eye(n)
            H = (I - rho * np.outer(s, y)) @ H @ (I - rho * np.outer(y, s)) + rho *
                 np.outer(s, s)

```

```
            x = x_new

    print("最大迭代次数达到，未收敛。")
    return x, max_iter
```

4.1 实验结果

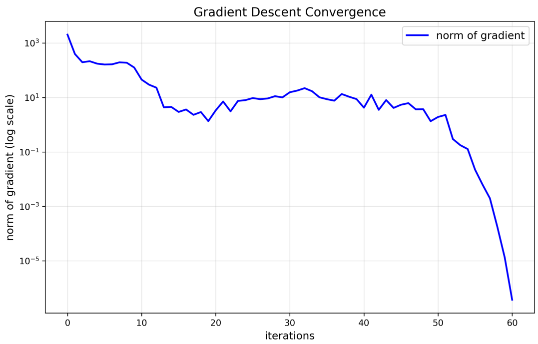


(a) n=6 时的梯度下降图

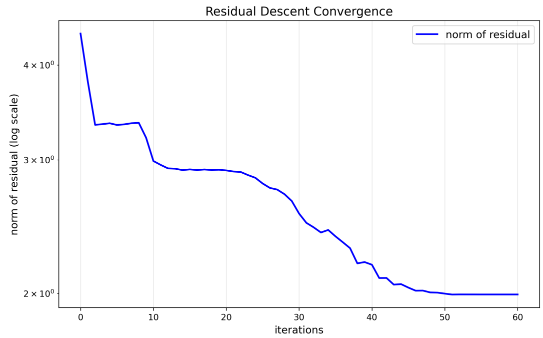


(b) n=6 时的残差下降图

图 5: n=6 时 Rosenbrock 函数实验

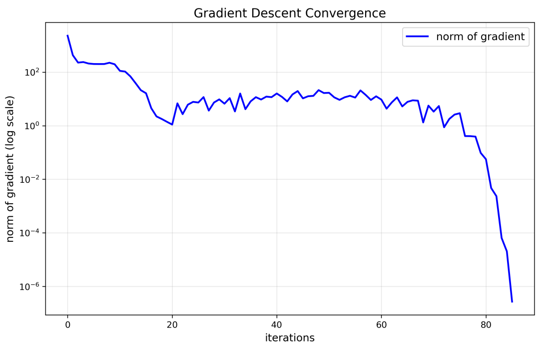


(a) n=8 时的梯度下降图

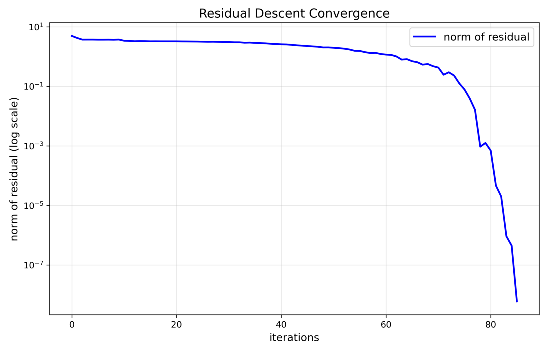


(b) n=8 时的残差下降图

图 6: n=8 时 Rosenbrock 函数实验

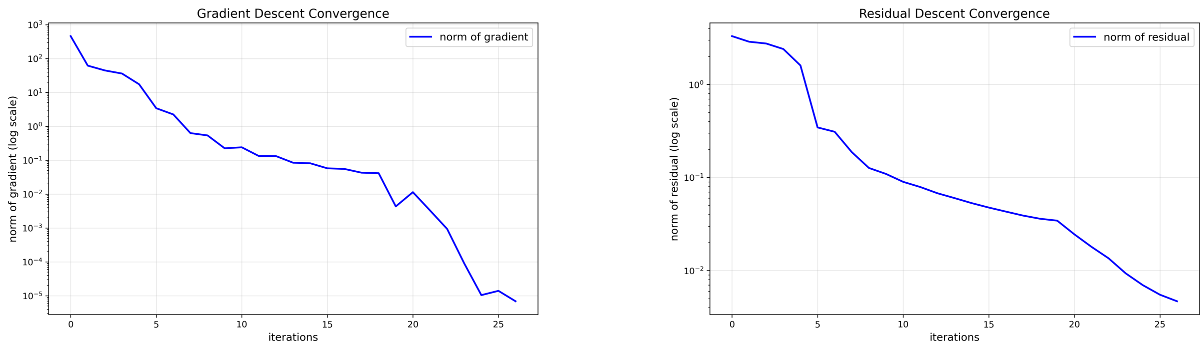


(a) n=10 时的梯度下降图



(b) n=10 时的残差下降图

图 7: n=10 时 Rosenbrock 函数实验



(a) Powell 函数梯度下降图　(b) Powell 函数残差下降图

图 8: Powell 奇异函数实验

4.2 实验分析

由图可以看出，在使用 BFGS 方法时，总是先经历一段平缓期，后在接近最优解时迅速下降，这也是拟牛顿方法的特点。

其次可以看出 BFGS 方法收敛快，并且稳定。