1 第一题

令 A 为 Hibert 矩阵，$A_{ij} = \frac{1}{i+j-1}$，b 为元素全为 1 的列向量。用共轭梯度法求解线性方程组 $Ax = b$，初始点选为 $x_0 = 0$，迭代终止条件为 $\|r_k\|_2 < 1e-6$，其中 $r_k = Ax_k - b$ 为第 k 次迭代的残差。

对于不同的维度 $n = 5, 8, 12, 20$，记录达到终止条件时的迭代次数，结果如表 1所示：

表 1: 不同维度下的迭代收敛情况

维度 n	迭代次数	最终残差范数
5	6	7.173835×10^{-8}
8	19	1.048855×10^{-8}
12	39	6.591321×10^{-7}
20	70	7.808762×10^{-7}

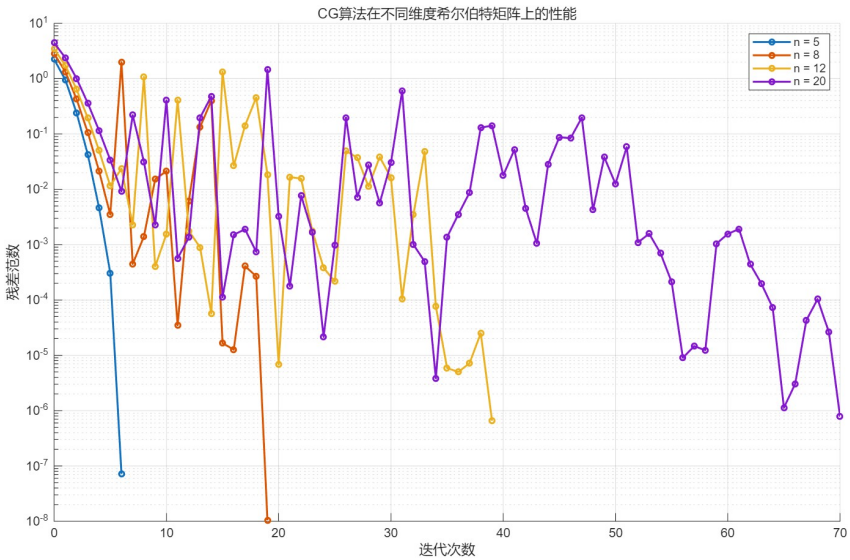

图 1: CG 法在不同维度下的收敛曲线

理论上，共轭梯度法在至多 n 次迭代内应能达到最优解，但由于浮点数的数值误差存在以及 Hibert 矩阵的条件数很大，放大了精度损失，导致收敛速度变慢，实际迭代次数超过了 n。

2 第二题

原先求解的是：$\min f(x) = \frac{1}{2}x^T Ax - b^T x$，Preconditioned CG 用了 $\hat{x} = Cx$，将其代入后要最小化的函数变成了：

$$\hat{\phi}(\hat{x}) = \frac{1}{2}\hat{x}^T(C^{-T}AC^{-1})\hat{x} - (C^{-T}b)^T\hat{x}$$

等价于求解 $(C^{-T}AC^{-1})\hat{x} = C^{-T}b$，即把原来标准 CG 法中的 A 换成 $C^{-T}AC^{-1}$，b 换成 $C^{-T}b$。

在 Preconditioned CG 中，我们令 $p_0 = -y_0$，$My_0 = r_0$，于是得到下面两个方程：

$$\hat{r}_0 = C^{-T}AC^{-1}\hat{x}_0 - C^{-T}b = C^{-T}(Ax_0 - b) = C^{-T}r_0$$
$$\hat{p}_0 = -\hat{r}_0 = -C^{-T}r_0$$

$$\hat{\alpha}_0 = \frac{(C^{-T}r_0)^T(C^{-T}r_0)}{(C^{-1}\hat{p}_0)^T A(C^{-1}\hat{p}_0)} = \frac{r_0^T \cdot C^{-1}C^{-T}r_0}{(C^{-1}\hat{p}_0)^T A(C^{-1}\hat{p}_0)} = \frac{r_0^T y_0}{p_0^T A p_0}$$

由上面两式，可以推出 M：

$$\begin{cases} C^{-1}C^{-T}r_0 = y_0 = M^{-1}r_0 \\ C^{-1}\hat{p}_0 = -C^{-1}C^{-T}r_0 = p_0 = -M^{-1}r_0 \end{cases} \quad \Rightarrow \quad M = C^T C$$

我们计算剩下的几步：

$$\hat{\beta}_{k+1} = \frac{\hat{r}_{k+1}^T \hat{r}_{k+1}}{\hat{r}_k^T \hat{r}_k} = \frac{r_{k+1}^T (C^{-1}C^{-T}) r_{k+1}}{r_k^T (C^{-1}C^{-T}) r_k} = \frac{r_{k+1}^T y_{k+1}}{r_k^T y_k}$$

$$\begin{aligned} p_{k+1} = C^{-1}\hat{p}_{k+1} &= C^{-1}\left(-C^{-T}r_{k+1} + \hat{\beta}_{k+1} C p_k\right) \\ &= -C^{-1}C^{-T}r_{k+1} + \hat{\beta}_{k+1} p_k \\ &= -y_{k+1} + \hat{\beta}_{k+1} p_k \end{aligned}$$

Preconditioned CG 算法如下：

Algorithm 1 Preconditioned Conjugate Gradient (PCG)

Given x_0, preconditioner M;

Set $r_0 \leftarrow Ax_0 - b$, $p_0 \leftarrow -y_0$, $k \leftarrow 0$;

while $r_k \neq 0$ **do**

$\alpha_k \leftarrow \dfrac{r_k^T y_k}{p_k^T A p_k}$;

$x_{k+1} \leftarrow x_k + \alpha_k p_k$;

$r_{k+1} \leftarrow r_k + \alpha_k A p_k$;

Solve $M y_{k+1} = r_{k+1}$;

$\beta_{k+1} \leftarrow \dfrac{r_{k+1}^T y_{k+1}}{r_k^T y_k}$;

$p_{k+1} \leftarrow -y_{k+1} + \beta_{k+1} p_k$;

$k \leftarrow k + 1$;

end while

3 第三题

$$B_{k+1}^{\mathrm{BFGS}} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k},$$

则

$$B_{k+1} = B_{k+1}^{\mathrm{BFGS}} + \phi_k (s_k^T B_k s_k) v_k v_k^T.$$

$$\det(B_{k+1}) = \det(B_{k+1}^{\mathrm{BFGS}}) \left[1 + \phi_k (s_k^T B_k s_k) \cdot v_k^T (B_{k+1}^{\mathrm{BFGS}})^{-1} v_k\right].$$

要证明 B_{k+1} 奇异，只需证明括号内项为零，即：

$$1 + \phi_k (s_k^T B_k s_k) \cdot v_k^T (B_{k+1}^{\mathrm{BFGS}})^{-1} v_k = 0.$$

令 $H_{k+1}^{\mathrm{BFGS}} = (B_{k+1}^{\mathrm{BFGS}})^{-1}$，则等价于验证：

$$v_k^T H_{k+1}^{\mathrm{BFGS}} v_k = -\frac{1}{\phi_k(s_k^T B_k s_k)}.$$

首先观察向量 v_k 与 s_k 的内积：

$$s_k^T v_k = s_k^T \left(\frac{y_k}{y_k^T s_k} - \frac{B_k s_k}{s_k^T B_k s_k} \right) = \frac{s_k^T y_k}{y_k^T s_k} - \frac{s_k^T B_k s_k}{s_k^T B_k s_k} = 1 - 1 = 0.$$

因此，$s_k^T v_k = 0$。

对于

$$H_{k+1}^{\mathrm{BFGS}} = (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T, \quad \text{其中} \rho_k = \frac{1}{y_k^T s_k},\ H_k = B_k^{-1}.$$

由于 $s_k^T v_k = 0$，我们有：

$$(I - \rho_k y_k s_k^T) v_k = v_k - \rho_k y_k (s_k^T v_k) = v_k,$$

且

$$\rho_k s_k s_k^T v_k = \rho_k s_k \cdot 0 = 0.$$

因此，

$$H_{k+1}^{\mathrm{BFGS}} v_k = (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) v_k + \rho_k s_k s_k^T v_k = (I - \rho_k s_k y_k^T) H_k v_k.$$

代入 $v_k = \rho_k y_k - \sigma_k B_k s_k$，其中 $\sigma_k = \dfrac{1}{s_k^T B_k s_k}$：

$$H_k v_k = H_k(\rho_k y_k - \sigma_k B_k s_k) = \rho_k H_k y_k - \sigma_k H_k B_k s_k = \rho_k H_k y_k - \sigma_k s_k.$$

于是：

$$H_{k+1}^{\mathrm{BFGS}} v_k = (I - \rho_k s_k y_k^T)(\rho_k H_k y_k - \sigma_k s_k) = \rho_k H_k y_k - \sigma_k s_k - \rho_k s_k y_k^T (\rho_k H_k y_k - \sigma_k s_k).$$

展开最后一项：

$$y_k^T (\rho_k H_k y_k - \sigma_k s_k) = \rho_k y_k^T H_k y_k - \sigma_k y_k^T s_k,$$

所以：

$$H_{k+1}^{\mathrm{BFGS}} v_k = \rho_k H_k y_k - \sigma_k s_k - \rho_k^2 (y_k^T H_k y_k) s_k + \rho_k \sigma_k (y_k^T s_k) s_k.$$

注意到 $\rho_k \sigma_k (y_k^T s_k) = \sigma_k$，因此：

$$-\sigma_k s_k + \rho_k \sigma_k (y_k^T s_k) s_k = -\sigma_k s_k + \sigma_k s_k = 0,$$

故：

$$H_{k+1}^{\mathrm{BFGS}} v_k = \rho_k H_k y_k - \rho_k^2 (y_k^T H_k y_k) s_k.$$

$$\begin{aligned} v_k^T H_{k+1}^{\mathrm{BFGS}} v_k &= v_k^T \left(\rho_k H_k y_k - \rho_k^2 (y_k^T H_k y_k) s_k \right) \\ &= \rho_k v_k^T H_k y_k - \rho_k^2 (y_k^T H_k y_k)(v_k^T s_k) \\ &= \rho_k v_k^T H_k y_k \quad （因为 v_k^T s_k = 0）. \end{aligned}$$

现在计算 $v_k^T H_k y_k$：

$$\begin{aligned} v_k^T H_k y_k &= (\rho_k y_k - \sigma_k B_k s_k)^T H_k y_k \\ &= \rho_k y_k^T H_k y_k - \sigma_k s_k^T B_k H_k y_k \\ &= \rho_k y_k^T H_k y_k - \sigma_k s_k^T y_k \quad （因为 B_k H_k = I） \\ &= \rho_k y_k^T H_k y_k - \sigma_k (y_k^T s_k). \end{aligned}$$

代入得:

$$\begin{aligned} v_k^T H_{k+1}^{\mathrm{BFGS}} v_k &= \rho_k \left(\rho_k y_k^T H_k y_k - \sigma_k y_k^T s_k \right) \\ &= \rho_k^2 y_k^T H_k y_k - \rho_k \sigma_k (y_k^T s_k). \end{aligned}$$

代入 $\rho_k = \dfrac{1}{y_k^T s_k}$, $\sigma_k = \dfrac{1}{s_k^T B_k s_k}$:

$$\begin{aligned} v_k^T H_{k+1}^{\mathrm{BFGS}} v_k &= \frac{y_k^T H_k y_k}{(y_k^T s_k)^2} - \frac{1}{s_k^T B_k s_k} \cdot \frac{y_k^T s_k}{y_k^T s_k} \\ &= \frac{y_k^T H_k y_k}{(y_k^T s_k)^2} - \frac{1}{s_k^T B_k s_k}. \end{aligned}$$

由 $\mu_k = \dfrac{(s_k^T B_k s_k)(y_k^T H_k y_k)}{(s_k^T y_k)^2}$, 可得:

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

因此:

$$v_k^T H_{k+1}^{\mathrm{BFGS}} v_k = \frac{\mu_k}{s_k^T B_k s_k} - \frac{1}{s_k^T B_k s_k} = \frac{\mu_k - 1}{s_k^T B_k s_k}.$$

$$\begin{aligned} 1 + \phi_k (s_k^T B_k s_k) \cdot v_k^T H_{k+1}^{\mathrm{BFGS}} v_k &= 1 + \phi_k (s_k^T B_k s_k) \cdot \frac{\mu_k - 1}{s_k^T B_k s_k} \\ &= 1 + \phi_k (\mu_k - 1). \end{aligned}$$

取 $\phi_k = \phi_k^c = \dfrac{1}{1 - \mu_k}$, 则:

$$1 + \frac{1}{1 - \mu_k}(\mu_k - 1) = 1 + \frac{-(1 - \mu_k)}{1 - \mu_k} = 1 - 1 = 0.$$

因此:

$$\det(B_{k+1}) = \det(B_{k+1}^{\mathrm{BFGS}}) \cdot 0 = 0.$$

即 B_{k+1} 奇异, 证毕。

4 第四题

用 BFGS 算法求解 Rosenbrock 函数:

$$f(x) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2 \right],$$

其中 $x_0 = [-1.2, 1, \cdots, -1.2, 1]^T$, $x^* = [1, 1, \cdots, 1, 1]^T$, $f(x^*) = 0$.

表 2: BFGS 算法解 Rosenbrock 函数在不同维度下的收敛表现

维度 n	迭代次数	收敛状态	最终函数值	最终梯度范数	前 4 个解分量
6	102	收敛	0.0080	0.000×10^{0}	[1.0000, 1.0000, 1.0000, 1.0000]
8	141	收敛	0.0037	0.000×10^{0}	[1.0000, 1.0000, 1.0000, 1.0000]
10	500	未收敛	0.0020	2.110×10^{-5}	[1.0000, 1.0000, 1.0000, 1.0000]

从表 2中可以看出，BFGS 算法在较低维度下能够较快收敛到 Rosenbrock 函数的最优解，而对于 10 维的情况，结果程序运行的结果显示未收敛，但解得的点和函数值已经非常接近最优解。

用 BFGS 算法求解 Powell 函数

$$f(x) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4,$$

其中 $\varepsilon = 10^{-5}$, $x_0 = [3, -1, 0, 1]^T$, $x^* = [0, 0, 0, 0]$, $f(x^*) = 0$.

表 3: BFGS 算法求解 Powell 函数的运行结果

指标	**值**
迭代次数	203
收敛状态	收敛
最终函数值	4.948×10^{-9}
最终梯度范数	0.000×10^{0}
最终解	[0.0041, -0.0004, 0.0040, 0.0040]

由表 3可见，BFGS 算法经过 203 次迭代成功收敛到 Powell 函数的最优解，最终函数值和梯度范数均接近于零，表明算法表现良好。

5 附录：代码实现

```
function [x,k,residuals] = CG(A,b,x0,tol,max_iter)

% 输出:
%   x          - 解
%   k          - 迭代次数
%   residuals  - 残差
    x = x0;
    r = A * x0 -b;
    p = -r;
    residual = norm(r);
    residuals = zeros(max_iter+1, 1);
    residuals(1) = residual;

    if residual < tol
        k = 0;
        return;
    end

    for k = 1:max_iter
        Ap = A * p;
        alpha =  - (r' * p) / (p' * Ap);
        x = x + alpha * p;
        r_old = r;
        r = r + alpha * Ap;
        residual = norm(r);
        residuals(k+1) = residual;
        if residual < tol
```

```
            break;
        end
        beta = (r' * r) / (r_old' * r_old);
        p = -r + beta * p;
    end

    if k == max_iter && residual >= tol
        warning('CG:maxIter', '共轭梯度算法在达到最大迭代次数 %d 后仍未收敛。', max_iter);
    end

    residuals = residuals(1:k+1);
end
```

Listing 1: CG 法

```
function [x,f_val,k,grad_hist] = BFGS(f,grad,x0,tol,max_iter)
% 输入:
%   f            - 目标函数
%   grad         - 梯度
%   x0           - 初始点
%   tol          - 停止迭代的条件
%   max_iter     - 最大迭代次数
%
% 输出:
%   x            - 计算出的最优解
%   f_val        - 最优解处的函数值
%   k            - 实际执行的迭代次数
%   grad_hist    - 梯度范数历史
    n = length(x0);
    x = x0;
    H = eye(n);

    g = grad(x);
    grad_norm = norm(g);
    grad_hist = zeros(max_iter+1,1);
    grad_hist(1) = grad_norm;

    alpha_init = 1.0; % 初始步长
    c1 = 1e-4;   % Armijo Condition
    rho = 0.5;   % 步长缩减因子

    for k = 1:max_iter

        if grad_norm < tol
            break;
        end

        % p_k = -H_k * g_k
        p = -H * g;

        % 回溯线搜索
        f_x = f(x);
```

```
        alpha = alpha_init;

        % Armijo Condition
        while f(x + alpha * p) > f_x + c1 * alpha * (g' * p)
            alpha = rho * alpha;
        end

        ap = alpha * p;
        x_old = x;
        x = x + ap;

        g_old = g;
        g = grad(x);
        y = g - g_old;
        s = x - x_old;

        if y' * s > 1e-12 % 避免除以零或负数
            rho_k = 1 / (y' * s);
            I = eye(n);
            term1 = I - rho_k * s * y';
            term2 = I - rho_k * y * s';
            H = ((y' *s) / (y' * y)) * I;
            H = term1 * H * term2 + rho_k * s * s';
        end

        grad_norm = norm(g);
        grad_hist(k+1) = grad_norm;
    end

    f_val = f(x);

    if k == max_iter && grad_norm >= tol
        warning('BFGS:maxIter', 'BFGS在达到最大迭代次数 %d 后仍未收敛。', max_iter);
    end
end
```

Listing 2: BFGS 法