数值分析作业

cxd@tongji.edu.cn

September 29, 2024

- **重要事项:** 作业布置于周四中午.
- **注意事项:** 请确定你交作业的方式, 线下或者线上, 同一次作业不能兼用两种方式(否则两边都按照没做全评分), 最好也不要在学习过程中改变提交方式.
- 纸质作业提交要求: 请下周四在课前提交到D202讲台。所有题目及程序写(不能打印!)在作业本上, 只提交画图的程序, 不必画图.
- 线上作业提交要求: 请下周四中午13:00前提交。请提交word或者pdf版本, 字体只能是楷书、黑体、宋体和仿宋，不能有任何手写痕迹(ipad等方式完成的, 请做OCR), 不得提交扫描、拍照、截图，否则退回。要求有图形的题目必须插入matlab生成的图形.

5 第三章

1. 试用SOR迭代计算线性方程组

$$\begin{cases} -55x_1 - 5x_2 + 12x_3 = 41 \\ 21x_1 + 36x_2 - 8x_3 = 52 \\ 24x_1 + 7x_2 + 74x_3 = 12 \end{cases} \tag{1}$$

取$\mathbf{x}^{(0)} = (0,0,0)^{\mathrm{T}}$, 松弛因子分别选取为$\omega = 0.1t$, $1 \le t \le 19$, 要求达到精度$\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| \le 10^{-4}$. 试通过数值计算得出不同的松弛因子所需要的迭代次数和收敛最快的松弛因子, 并指出哪些松弛因子使得迭代发散. 你可以把迭代结果画成柱状图, 或者下面的纯文本图形吗?

```
                 The iteration times of "SOR"
            0         10        20        30        40        50
OMEGA       ---------#---------#---------#---------#---------#   #Iter
 0.05      |************************************************      48
 0.10      |****************************************              40
 0.15      |*************************************                 37
 0.20      |***********************************                   35
 0.25      |*************************************                 37
```

2. 给出下面方程组用Jacobi迭代收敛的充要条件:

$$\left\{\begin{array}{rrrl} \alpha x_1 & -4x_2 & +4x_3 & =5 \\ 4x_1 & +\alpha x_2 & +3x_3 & =7 \\ -4x_1 & +3x_2 & +\alpha x_3 & =3 \end{array}\right. \tag{2}$$

3. 第三版现代数值计算, 习题三2

4. (选做,4,5两题任选其一) 计算矩阵A的逆矩阵的$(1,1)$元素, 其中矩阵A的对角元为非完全平方数的平方从小到大排列, 即下面这些数的平方:

$$2, 3, 5, 6, 7, 8, 10, 11, 12, 13, 14, 15, 17, 18, 19, \cdots.$$

而所有$|i-j|$ 是3 的(非负)幂次的(i,j) 元素为1, 否则为0, 矩阵的阶数为200000. 请给出你计算出该逆矩阵$(1,1)$元素到小数点后10位数的计算时间。
(你可以使用命令得到计算机运行时间:
(`t0=cputime; your_command ... ;  cputime-t0`
(我们将在下次课上点名要求同学讲解此题的完成过程, 请大家准备.

5. (选做,4,5两题任选其一) 第三版现代数值计算, 数值实验三2(给出当$n=10^6$时$\|x\|_2$的值, 保留10个有效数字.)

3 第二章

1. (手算)用乔列斯基分解计算下述线性方程组：

$$\begin{pmatrix} 4 & -1 & & & \\ -1 & 4 & -1 & & \\ & -1 & 4 & -1 & \\ & & -1 & 4 & -1 \\ & & & -1 & 4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 10 \\ 20 \\ 20 \\ 20 \\ 10 \end{pmatrix}.$$

2. (编程)已知Hilbert矩阵$H \in R^{n\times n}$定义为$H_{ij} = \frac{1}{i+j-1}$, matlab中的命令为`hilb`. 尝试用下面四种不同方法求解方程组$Hx = d$, 其中$d_i = n+1-i$, 矩阵阶数n可以取值为$5, 10, 20, 40$. 哪一种方法效果最好? 你可以把这些方法同matlab 的左除相比较.

   - 高斯消去法;
   - 列主元高斯消去法;
   - 全主元高斯消去法;
   - 平衡加权高斯消去法. 以第一次选主元为例, 此方法应用于方程组$Ax = b$时, 下面的最大值由第i个值取得, 则第i行为主元所在行:
   $$\max\left\{\frac{|a_{11}|}{\max_i |a_{1i}|}, \frac{|a_{21}|}{\max_i |a_{2i}|}, \cdots, \frac{|a_{n1}|}{\max_i |a_{ni}|}\right\}.$$

3. 设计一个方法求解方程组$Ax = b$, 其中$B, I \in R^{n\times n}$, $A \in R^{n^2\times n^2}$, I是单位矩阵。你可以令$n = 80$或者100:

$$A = \begin{bmatrix} B & -I & & \\ -I & B & \ddots & \\ & \ddots & \ddots & -I \\ & & -I & B \end{bmatrix} \quad B = \begin{bmatrix} 4 & -1 & & \\ -1 & 4 & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 4 \end{bmatrix}, \qquad (3)$$

   而$b = (0, 1, 2, 3, \cdots, n^2-1)^T$. 写明你设计这个算法的理由.

4. 第三版现代数值计算, 习题二2,3,4, 数值实验二1

4 第二章答案

2.1 ANSWER:设原方程组为$Ax = b$, 记A的Cholesky分解为$A = LDL^T$, 则

$$L = \begin{pmatrix} 1 & & & & \\ -\frac{1}{4} & 1 & & & \\ & -\frac{4}{15} & 1 & & \\ & & -\frac{15}{56} & 1 & \\ & & & -\frac{56}{209} & 1 \end{pmatrix}, \quad D = \begin{pmatrix} 4 & & & & \\ & \frac{15}{4} & & & \\ & & \frac{56}{15} & & \\ & & & \frac{209}{56} & \\ & & & & \frac{780}{209} \end{pmatrix}.$$

求解$Ly = b$, 得$y = (10, 45/2, 26, 755/28, 3600/209)^T$.
求解$Dz = y$, 得$z = (5/2, 6, 195/28, 1510/209, 60/13)^T$.
求解$L^T x = z$, 得$x = \frac{1}{13}(60, 110, 120, 110, 60)^T$.

2.2 ANSWER:编写程序如下, 可以在同一个函数中使用四种不同的高斯消去法:

```
function [x,err] = gauss(A,b,opt)
% Usage: [x,err] = gauss(A,b,opt)
% Solving linear equation Ax=b with
%   opt = 1: Gauss Elimination
%         2: Gauss Elimination with column pivot
%         3: Gauss Elimination with full pivot
%         4: Gauss Elimination with scaled-column pivot
% err =  norm( A*x-b )
    if nargin<3, opt = 1; end
    AA = A;
    bb = b;
    n = length(b);
    x = zeros(n,1);
    p = 1:n;
    for k = 1:n-1
% begin pivot
        switch opt
            case 2
                [~,t] = max( abs( A(k:n,k) ) );
                t = k - 1 + t;
```

```
            A([t,k],:) = A([k,t],:);
            b([t,k])   = b([k,t]);
        case 3
            [m,r] = max( abs( A(k:n,k:n) ) );
            [~,c] = max( m );
            r = k - 1 + r(c);
            c = k - 1 + c;
            A([r,k],:) = A([k,r],:);
            b([r,k])   = b([k,r]);
            A(:,[c,k]) = A(:,[k,c]);
            p([c,k])   = p([k,c]);
        case 4
            v = A(k:n,k) ./ max( abs(A(k:n,k:n)),[],2 );
            [~,t] = max( abs( v ) );
            t = k - 1 + t;
            A([t,k],:) = A([k,t],:);
            b([t,k])   = b([k,t]);
    end
% begin elimination
    for j = k+1:n
        r      = A(j,k) / A(k,k);
        i      = k+1:n;               % delete i and replace i by :
                                      %    in next row to read whole A
        A(j,i) = A(j,i) - r * A(k,i); % no need to compute (1:k)
        b(j)   = b(j)   - r * b(k);
    end
end
% back-track
for k = n:-1:1
    x(k) = ( b(k) - A(k,k+1:n)*x(k+1:n) ) / A(k,k);
end
x(p) = x;
err = norm( AA*x-bb );
```

可以使用下面的函数调用, 并生成数值报告. 这里我们有三种不同的方程组作为测试的例子. 你可以把该文件存为gauss_demo.m并运行它

```
data_opt = ceil(rand*3);
```

```
C = strvcat('Hilbert matrix with regular RHS',...
            'Hilbert matrix with random RHS',...
            'Random matrix with random RHS');
fprintf('%s.\n',C(data_opt,:));
S = strvcat('Gauss Elimin','GE with column-pivot',...
            'GE with full-pivot','GE with scaled-pivot');
fprintf('%20s%8d%8d%8d%8d\n','n = ',[5 10 20 40] );
fprintf('%60s\n',repmat('=',1,60));
for opt = 1:4
    fprintf('%20s',S(opt,:));
    for n = [5 10 20 40]
        switch data_opt
            case 1
                A = hilb(n);
                b = (n:-1:1)';
            case 2
                A = hilb(n);
                b = rand(n,1)*100;
            case 3
                A = rand(n)*100;
                b = randn(n,1);
        end
        [x,err] = gauss(A,b,opt);
        fprintf('%9.2g',err);
    end
    fprintf('\n');
end
```

数值报告可以直接从命令行得到:

```
>> gauss_demo
```

2.3 Hint: 你可能使用的方法, 一般都应该是块方法。比如, 可以使用块LU分解, 块Cholesky 分解, 当然也可以使用Gauss 消去法. 下面的方法是块Cholesky分解.

```
function b = ex25(n)
   if nargin<1, n=3; end
```

```
    b = (1:n^2)';
    blk = @(i,n)((i-1)*n+1:i*n);
    D2 = diag(ones(n-1,1),1);
    B = diag(ones(n,1)*4) - D2 - D2';
    Z = zeros(n);
    G{1} = chol( B );
% decomposition
    for k = 1:n-1
        E{k}   = - inv( G{k}' );
        G{k+1} = chol( B - E{k}'*E{k} );
    end
% back substitution 1
    b(blk(1,n)) =  G{1}' \ b(blk(1,n));
    for k = 2:n
        b(blk(k,n)) = b(blk(k,n)) - E{k-1}'*b(blk(k-1,n));
        b(blk(k,n)) = G{k}' \ b(blk(k,n));
    end
% back substitution 2
    b(blk(n,n)) =  G{n} \ b(blk(n,n));
    for k = n-1:-1:1
        b(blk(k,n)) = b(blk(k,n)) - E{k}*b(blk(k+1,n));
        b(blk(k,n)) = G{k} \ b(blk(k,n));
    end
```

1 第一章

1. 用最少的计算量和存储量计算给定矩阵的逆矩阵的(1,1)元素, 该给定矩阵为n阶矩阵, 其对角元全部为3, 上下次对角元(即行列下标差为1 的元素)都是-1, 未涉及元素全部为0. 要求输入n, 输出所求逆矩阵(1,1)元素的小数点后16位. n可能的输入值为1, 10, 100，1000, 10000. 列表给出你的计算量和存储量关于阶数n的关系。

2. 有效数$a = 3.1416$, $b = 2.72$分别是数a^*和b^*的近似值, 请估算a^*b^*, $a^* - b^*$以及a^*/b^*的近似值, 并把它们写成有效数.

3. 第三版现代数值计算, 习题一5,6, 数值实验一6,9

2 第一章答案

1.1 **ANSWER:**记该n阶矩阵为A_n. 因为$A_n^{-1} = \frac{1}{|A_n|}A_n^*$, 其中$A_n^*$ 为A_n 的伴随矩阵.
因此, $(A_n^{-1})_{1,1} = \frac{(A_n^*)_{1,1}}{|A_n|} = \frac{|A_{n-1}|}{|A_n|}$. 由行列式$|A_n|$ 最后一列展开, 得

$$|A_n| = 3|A_{n-1}| - |A_{n-2}|.$$

两边除以$|A_{n-1}|$得

$$\left((A_n^{-1})_{1,1}\right)^{-1} = \frac{|A_n|}{|A_{n-1}|} = 3 - \frac{|A_{n-2}|}{|A_{n-1}|} = 3 - (A_{n-1}^{-1})_{1,1}.$$

此即

$$(A_n^{-1})_{1,1} = \frac{1}{3 - (A_{n-1}^{-1})_{1,1}}.$$

因此, 有程序如下:

```
function v = ex1_1(n)
   a(1) = 1/3;
   for k = 2:n
       a(k) = 1 / (3-a(k-1));
   end
   v = a(end);
```

$n = 1, 10, 100, 1000, 10000$时的值分别为0.3333333333333333, 0.381966009824403, 0.381966011250105, 0.381966011250105, 0.381966011250105. 列表略。

1.2 **ANSWER:**由于a, b是有效数, 因此$3.14155 \le a^* < 3.14165$, $2.715 \le b^* < 2.725$.
因此, $8.52930825 \le a^*b^* < 8.56099625$. a^*b^*不论写成8或8.5都可能没有有效位。只能写成9.
同理, $0.41655 \le a^* - b^* < 0.42665$. $a^* - b^* \approx 0.4$.
同理, $1.1528623853211 \le a^*/b^* < 1.15714548802947$. $a^*/b^* \approx 1.2$.

1.3 略。