数值第一次上机作业

121110008 陈晓宇

1. 作业题目：

练习 6.1. 设 x⋆= (1, 1, 1, . . . , 1)⊤为线性方程组(6.0.3) 的真解，右端向量由这个真解给出。

1. 编制高斯消元法和 LDL⊤算法，求解线性方程组。

2. 利用纵轴上的对数坐标，绘制数值误差同参数 n 的关系图。

3. 绘制所用的 CPU 时间同参数 n 的关系图。

4. 绘制矩阵条件数与参数 n 的关系。请问：摄动理论给出的舍入误差估计 (1.5.17) 是否完美地刻画了相对误差的大小？.

练习6.2矩阵非零元素的分布对数值计算的影响，这个问题最优答案的确定是个NP 问题。考虑行列重排后相等的两个矩阵B1= B2=比较三角分解后的非零元素分布。在Matlab 中观测矩阵结构图的命令是spy()。利用矩阵的元素分布特点修改Crout 算法，努力尝试省掉那

些无用的零运算时间，观察是否有CPU 时间的节省。

练习6.3. 求Tn 的逆矩阵。

练习6.4. 考虑线性方程组DnTnx = b，其中Dn = diag(2..i)。

取真解为x. = (1; 2; 3; : : : ; n).，令n 从500 变换到2000.

1. 观测追赶法的数值误差与矩阵阶数n 的关系。

2. 考虑等价的问题Tnx =  b，追赶法的计算结果能否得到改善？

练习6.5. 随机构造大量的n 阶可逆矩阵A，进行列主元高斯消元分解。观测主元增长因子 (A) 是否处于n^(2/3) 或n^(1/2) 的量级。

1. 实现过程（代码见附录，附录代码不含在command window里运行的代码）

1（1）利用算法进行计算

1（2）数值误差：将真实值与估计值进行相减，取模的最大值，因为差距较小，以此为纵坐标画的图不能体现出差距，因此去一下对数，才能发现具体关系

1（3）利用matlab中自带的tic，toc函数，可以测得CPU时间

1（4）利用matlab中的cond函数可以直接取得所要求矩阵条件数

2.考虑B1和B2的矩阵分布特点，在构造B1和B2的时候，可以看出B2只是B1的一个旋转，所以构造出B1之后就可以使用rot函数将B1进行旋转即可得到B2，用spy函数观察两个矩阵的构图时发现，虽然两者重拍后都是相等的，但是在进行Crout算法的时候，会发现因为0元素分布的特殊性，可以根据0元素的位置修改Crout算法，可以将Crout算法中有的求和公式去掉，直接采取赋值即可，并且原来的Crout算法是三阶循环，但是修改之后可以将循环的阶数降低一阶，可以变成二阶循环。但是对于B2，在用spy进行观察时，无法进行改进。

3.求逆矩阵，按照已给的算法进行即可。

4.使用追赶法解线性方程组，直接利用书上的算法即可，在matlab中使用的时候，本来是可以使用matlab的优势直接进行行或者列运算，但是在算法中发现，算法中进行的运算是无法按照行或者列进行的运算，只能按照矩阵的布局进行一个个的单独运算

5.随机构造大量的矩阵，可以使用rand函数，但rand函数生成的是在（0,1）的数，所以可以使用100\*rand来扩大矩阵元素的数值，主元因子可以通过max函数求出或者利用二阶范数norm求出，最后可以将（A），n^(1/2)和n

^(2/3)画在同一个图中，进行比较。

1. 结果和分析

1（2）. 

这是利用高斯顺序消元法解得的线性方程组的解与原来真实解之间的最大误差值与n之间的关系。这里只取了[1,50]内的整数，会发现运算越来越慢。



这是算法的误差分析，会发现到后来取log之后误差越来越趋于0，也就是原来的误差越来越趋于稳定，由此图也可以看出算法的误差是递增的

1（3）. 这是运行Gauss顺序消元法所需的CPU时间与n之间的关系，会发现随着n取值的变大，time增长速度越来越快，最后直接趋于无穷了。



上图是算法的运行时间与n之间的关系，与上图相比，二者在整体上相差不大

1（4）. 

上图是矩阵的条件数与n之间的一个关系。

下图是舍入误差与n之间的关系图：



2.

这是B1的lu分解之后所产生的矩阵L的spy分解之后的图形



下图是B1的lu分解之后的U



接下来展示B2lu分解之后的L：



下图是U：



对比B1的分解会发现，B1的0元素分布的更加稀疏，这就产生了可以根据0元素所在的位置修改Crout算法的可能，而B2的0元素的分布就只是传统的上三角和下三角，就无法进行Crout算法的改进。

4.下图是未进行任何改变的追赶法的数值误差与n之间的关系

 这里的n是取[500,1000]的，在实际运行中会发现，当n超过1000多点的时候，因为所取的矩阵的特殊性，matlab会无法的出一个数，直接报出NAN了，所以从图中可以看出在1000多的地方，图像突然向上有个转折点，说明那以后，追赶法已经不适用了。

而下图是改变矩阵之后所得到的误差值与n之间的关系图



由此图看出，改变之后的计算结果在误差上是比原来的小了很多，而原来的在1000附近已经是NAN了，这是因为所求的矩阵有些病态，要是换成单位矩阵之类的，还是可以运行的，而这个依旧能进行运算。

5. 

上图将所取得的随机矩阵的主元增长因子和power(n,1/2)和power（n，2/3）画在同一个图中，可以比较发现，主元增长因子确实处于两个量级。

//四．附录（代码）

练习6.1

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileWriter;

import java.io.IOException;

import java.util.Scanner;

//这是Gauss消元法解方程组

public class Gauss {

public static void main(String[] args) throws IOException {

long a = System.currentTimeMillis();

Gauss gauss = new Gauss();

double[] M = new double[100];

M[0] = 0;

double[] T = new double[100];

T[0] = 0;

for (int m = 20; m < 51; m++) {

M[m-20] = gauss.Matirx(m);

T[m-20] = (System.currentTimeMillis()-a)/1000f;

System.out.println(M[m]);

System.out.println(m);

}

System.out.println("结了");

File file = new File("wucha.txt");

File file2 = new File("m.txt");

File file3 = new File("shijian.txt");

try {

if (file.exists()) file.delete();

if (file2.exists()) file2.delete();

if(file3.exists()) file3.delete();

BufferedWriter bufferedWriter =new BufferedWriter(new FileWriter(file));

BufferedWriter bufferedWriter2 = new BufferedWriter(new FileWriter(file2));

BufferedWriter bufferedWriter3 = new BufferedWriter(new FileWriter(file3));

StringBuffer outBuffer = new StringBuffer();

StringBuffer ouStringBuffer = new StringBuffer();

StringBuffer ouStringBuffer2 = new StringBuffer();

for (int i = 20; i < 51; i++) {

outBuffer.append(Math.log(M[i-20])+"\r\t");

ouStringBuffer.append(i+"\r\t");

ouStringBuffer2.append(T[i-20]+"\r\t");

}

bufferedWriter.write(outBuffer.toString());

bufferedWriter2.write(ouStringBuffer.toString());

bufferedWriter3.write(ouStringBuffer2.toString());

bufferedWriter.flush();

bufferedWriter.close();

bufferedWriter2.flush();

bufferedWriter2.close();

bufferedWriter3.flush();

bufferedWriter3.close();

} catch (Exception e) {

e.printStackTrace();// TODO: handle exception

}

//System.out.println((System.currentTimeMillis()-a)/1000f);

}

public double Matirx(int m) {

// System.out.println("请输入方程的阶数:");

// Scanner input = new Scanner(System.in);

// int m = input.nextInt();

int n= m\*m;

// input.close();

int ord = 10000;

double[][] a = new double[ord][ord];

double[] c = new double[ord];

double[] x = new double[ord];

for (int i = 0; i < ord; i++) {

for (int j = 0; j < ord; j++) {

a[i][j] = 0;

}

}

for (int i = 0; i < n; i++) {

a[i][i] = 4;

if (i < n-m) {

a[i][m+i] = -1;

a[m+i][i] = -1;

}

if (i < n-1 && (i+1)%m != 0) {

a[i][i+1] = -1;

a[i+1][i] = -1;

}

}

for (int i = 0; i < n; i++) {

for (int j = 0; j < n; j++) {

// System.out.println(a[i][j]);

}

}

for (int i = 0; i < n; i++) {

for (int j = 0; j < n; j++) {

a[i][n] += a[i][j];

}

}

for (int k = 0; k < n - 1; k++) {

for (int i = k+1; i < n; i++) {

c[i] = a[i][k]/a[k][k];

for (int j = k; j <= n; j++) {

a[i][j] = a[i][j] - c[i]\*a[k][j];

}

}

}

x[n-1] = a[n-1][n]/a[n-1][n-1];

for (int i = n-2; i >= 0; i--) {

x[i] = a[i][n];

for (int j = n-1; j > i; j--) {

x[i] -= a[i][j]\*x[j];

}

x[i] /= a[i][i];

}

// for (int i = 0; i < n; i++) {

// System.out.println(Math.abs(x[i]-1));

// }

double max = Math.abs(x[0]-1);

for (int i = 0; i < n; i++) {

if(max <= Math.abs(x[i]-1)){

max = Math.abs(x[i]-1);

}

}

return max;

}

}

6.1的LDL

function [wucha] = E6112(m)

n = m\*m;

A = 4\*ones(1,m);

B = (-1)\*ones(1,m-1);

T = diag(A)+diag(B,1)+diag(B,-1);

I = (-1)\*eye(m);

a = blkdiag(kron(eye(m),T))+kron(diag(ones(1,m-1),1),I)+kron(diag(ones(1,m-1),-1),I);

x = ones(n,1);

b = a\*x;

y = zeros(n,1);

g = zeros(n,1);

%t0 = cputime;

if a(1,1) == 0

disp('Method Failed');

end

g(1) = a(2,1);

a(2,1) = g(1)/a(1,1);

a(2,2) = a(2,2)-g(1)\*a(2,1);

if a(2,2) == 0

disp('Method Failed');

end

for i = 3:n

g(1) = a(i,1);

for j = 2:n-1

summ = 0;

for k = 2:j-1

summ = summ + g(k)\*a(j,k);

end

g(j) = a(i,j) - summ;

end

a(i,1) = g(1)/a(1,1);

for j = 2:i-1

a(i,j) = g(j)/a(j,j);

end

summ = 0;

for k = 1:i-1

summ = summ + g(k)\*a(j,k);

end

a(i,i) = a(i,i) - summ;

if a(i,i) == 0

disp('Method Failed');

end

end

for i = 2:n

summ = 0;

for k = 1:i-1

summ = summ+a(i,k)\*b(k);

end

y(i) = b(i) - summ;

end

for i = 2:n

b(i) = y(i);

end

x(n) = b(n)/a(n,n);

for i = n-1:-1:1

summ = 0;

for k = i+1:n

summ = summ+a(k,i)\*x(k);

end

x(i) = b(i)/a(i,i) - summ;

end

wucha = log(max((abs(x-ones(n,1)))))

%t1 = cputime;

end

练习6.2

n = 10;

a = 4;

B1 = eye(n-1);

b = a\*ones(n-1,1);

B1 = [B1,b];

b = [b;1];

c = b';

B1 = [B1;c];

spy(B1);

% tic

% [L,U,P] = lu(B1)

% toc

% spy(B1)

l = zeros(n);

u = eye(n);

y = zeros(n,1);

x = zeros(n,1);

%ÏÂÃæÊÇÎ´ÐÞ¸ÄµÄCroutËã·¨

tic

for i = 1:n

l(i,1) = B1(i,1);

end

for j = 1:n

u(1,j) = B1(1,j)/l(1,1);

end

for k = 2:n

for i = k:n

sum = 0;

% syms r;

% l(i,k) = B1(i,k) - symsum(l(i,r)\*u(r,k),[1,k-1]);

% ±È½ÏÄªÃûÆäÃî£¬²»ÖªµÀ´íÔÚÄÄ¶ù£¬±¨´íÒ²Ê®·ÖÆæÌØ

for r = 1:k-1

sum = sum + l(i,r)\*u(r,k);

end

l(i,k) = B1(i,k)-sum;

end

for j = k+1:n

sum = 0;

%u(k,j) = (B1(k,j) - symsum(l(k,r)\*u(r,j),[1,k-1]))/l(k,k);

for r = 1:k-1

sum = sum+l(k,r)\*u(r,j);

end

u(k,j) = (B1(k,j)-sum)/l(k,k);

end

end

toc

%ÏÂÃæÊ¹ÓÃ¸Ä½øºóµÄCroutËã·¨

figure(1);

spy(l);

figure(2);

spy(u);

B1;

tic

l(1,1) = B1(1,1);

u(1,4) = B1(1,4)/l(1,1);

for k = 2:n-1

for i = k:n

l(i,k) = B1(i,k);

end

for j = k+1:n

u(k,j) = B1(k,j)/l(k,k);

end

end

toc

练习62的B2

n = 10;

a = 4;

B2 = diag(ones(1,n));

b = a\*ones(n-1,1);

B2(2:n,1) = a;

B2(1,2:n) = a;

B2

spy(B2);

%[L,U,P] = lu(B2)

% spy(B2);

l = zeros(n);

u = eye(n);

for i = 1:n

l(i,1) = B2(i,1);

end

for j = 1:n

u(1,j) = B2(1,j)/l(1,1);

end

for k = 2:n

for i = k:n

sum = 0;

% syms r;

% l(i,k) = B1(i,k) - symsum(l(i,r)\*u(r,k),[1,k-1]);

% for r = 1:k-1

sum = sum + l(i,r)\*u(r,k);

end

l(i,k) = B2(i,k)-sum;

end

for j = k+1:n

sum = 0;

%u(k,j) = (B1(k,j) - symsum(l(k,r)\*u(r,j),[1,k-1]))/l(k,k);

for r = 1:k-1

sum = sum+l(k,r)\*u(r,j);

end

u(k,j) = (B2(k,j)-sum)/l(k,k);

end

end

% figure(1);

% spy(l);

% figure(2);

% spy(u);

练习6.3

% n = 4;

% a = 2\*ones(1,n);

% b = (-1)\*ones(1,n-1);

% c = (-1)\*ones(1,n-1);

% A = diag(a)+diag(b,1)+diag(c,-1);

%inv(T)

function [F] = Gauss\_Jorden(A)

n = rank(A);

[x,y] = size(A);

if x ~= y

disp('²»´æÔÚÄæ¾ØÕó');

else

d = det(A);

if d == 0

disp('ÎÞÄæ¾ØÕó');

else

B=eye(x);

A = [A,B];

y = 2\*y;

for k = 1:x

max = abs(A(k,k));

r = k;

for L = k+1:x;

if max < abs(A(L,k))

max = abs(A(L,k));

r = L;

end

end

t = A(k,:);

A(k,:) = A(r,:);

A(r,:) = t;

s = A(k,k);

for j = 1:y

A(k,j) = A(k,j)/s;

end

for i = 1:x

if i ~= k

for j = k+1:y

A(i,j) = A(i,j)-A(i,k)\*A(k,j);

end

end

end

end

F = A(:,x+1:y);

disp('¸Ã¾ØÕóµÄÄæ F =')

end

end

练习6.4

function[wucha] = E614(n)

d = zeros(n,1);

x1 = zeros(n,1);

a = 2\*ones(1,n);

b = (-1)\*ones(1,n-1);

c = (-1)\*ones(1,n-1);

T = diag(a)+diag(b,1)+diag(c,-1); %¾ØÕóTµÄÖµ

for i = 1:n

d(i) = 2^(-i);

end

D = diag(d); %¾ØÕóDµÄÖµ

for i = 1:n

x1(i) = i;

end

% size(T)

% size(D)

A = D\*T;

d = diag(A);

a = diag(A,-1);

c = diag(A,1);

b = A\*x1;

if d(1) == 0

disp('Method Failed');

end

p = zeros(n,1);

q = zeros(n,1);

y = zeros(n,1);

p(1) = d(1);

q(1) = c(1)/d(1);

for k = 2:n-1

p(k) = d(k)- a(k-1)\*q(k-1);

if p(k) == 0

disp('Method Failed');

end

q(k) = c(k)/p(k);

end

p(n) = d(n)-a(n-1)\*q(n-1);

if p(n) == 0

disp('Mehtod Failed');

end

y(1) = b(1)/p(1);

for k = 2:n

y(k) = (b(k)-a(k-1)\*y(k-1))/p(k);

end

x(n) = y(n);

for k = n-1:-1:1

x(k) = y(k)- q(k)\*x(k+1);

end

wucha = log(max(max(abs((x'-x1)))))

end

练习6.5

function [ha] = E615(n)

A = 100\*rand(n);

a0 = max(max(A));

if a0 == 0

disp('Method Failed');

return;

end

a = zeros(1,n-1);

for k = 1:n-1

r = k;

for i = k+1:n

if abs(A(i,k))>abs(A(r,k))

r = i;

end

end

if A(r,k) == 0

disp('A is singular');

return

end

if r ~= k

t = A(k,k:n);A(k,k:n) = A(r,k:n);A(r,k:n) = t;

end

A(k+1:n,k) = A(k+1:n,k)/A(k,k);

A(k+1:n,k+1:n) = A(k+1:n,k+1:n)-A(k+1:n,k)\*A(k,k+1:n);

A(k+1:n,k) = 0;

a(k) = max(max(A));

end

ha = max(a);

ha = ha/a0;

end