ЛАБОРАТОРНАЯ РАБОТА 3. РЕШЕНИЕ СИСТЕМ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ ПРЯМЫМИ МЕТОДАМИ. ТЕОРИЯ ВОЗМУЩЕНИЙ

Евтеева Марина, БПМ-143

Вариант 7

2017

Оглавление

Задание 3.1.7 Дана система уравнений $Ax = b$ порядка n. Исследовать зависимость погрешности решения x от погрешностей правой части системы b
Вариант 7
$n = 6$
$A_{i,j} = 256/(5 + 0.1 * N * i * j * 0.256)^5$
Решение:

1. Решаю систему $Ax = b$ методом Гаусса при помощи Матлаба.
$$x = \begin{matrix} 1715,38141400797 \\ -12130,1070267084 \\ 40418,1343479742 \\ -68845,5094824987 \\ 57944,4604058678 \\ -19088,6844483299 \end{matrix}$$

2. С помощью функции Матлаба $cond(A)$ нахожу число обусловленности. Оно равно $cond(A) = 3.998997788886389e + 06$ - матрица плохо обусловлена.

3. Вычисляю вектор d относильных погрешностей решений, считая решение из первого пункта точным. $d_i = \frac{\|x-x^i\|_\infty}{\|x\|_\infty}$, x^i - решения систем $Ax^i = b^i$, где $b^i_k = b_k\ k \neq i\ and\ b^i_k = b_k + \delta\ k = i$.
d = $0,0451860894392119 \quad 0,543365676729877 \quad 2,39410749083762 \quad 4,91214835133110 \quad 4,7019718$

4. Получаю, что 4 компонента вектора d самая большая, значит 4 компонента вектора b больше всего влияет на погрешность решения

5. Оцениваю теоретическую погрешность по неравенству $\delta(x^m) \leq cond(A) * \delta(b^m)$, расчеты в матлабе показали, что неравенство верное, в основном за счет числа обусловленности.

Код MATLAB

```
format long
n = 6;
N = 7;
b = zeros(n,1)+N;
aij = @(i,j)(256/(5+0.1*N*i*j*0.256)^5);
A = zeros(n);
for i = 1:n
    for j = 1:n
        A(i,j) = aij(i,j);
    end
end
C = rref([A b]);
x = C(:,end)
%x = inv(A)*b
condnum = cond(A, inf);
d = zeros(n, 1);
delta = N/100;
```

```
for i = 1:n
    d(i) = norm(x - xi(A, b, i, delta),inf)/norm(x,inf);
end
dm = find(d >= max(d));
deltaxm = norm(x - xi(A, b, dm, delta), inf)/norm(x,inf);
bi = b;
bi(dm) = bi(dm) + delta;
deltabm = norm(b - bi, inf)/norm(b, inf);
if deltaxm <= condnum*deltabm
    disp('True');
else
    disp('False');
end



function [x] = xi(A, b, i, delta)
    b(i) = b(i) + delta;
    C = rref([A b]);
    x = C(:,end);
end
```

Задание 3.3.2 Дана матрица A. Найти число обусловленности матрицы, используя вычислительный эксперимент.
Вариант 7

3 1 0 0 0
1 2 1 0 0
0 1 1 1 0
0 0 1 0 1
0 0 0 1 1

Решение:

1. Выбираю последовательность линейно независимых векторов b^i и решаю соответствующие системы уравнений $Ax^i = b^i$ при помощи метода Гаусса Матлаба.
   b =
   1 2 3 4 5
   1 1 2 3 4
   1 2 1 2 3
   1 3 2 1 2
   1 4 3 2 1
   Где соответствующие b^i расположены по столбцам. Получившиеся решения:
   x =
   $$0,285714285714286\ 0,714285714285714\ 0,714285714285714\ 1\ 1,57142857142857$$
   $$0,142857142857143\ -0,142857142857143\ 0,857142857142857\ 1\ 0,285714285714286$$
   $$0,428571428571429\ 0,571428571428571\ -0,428571428571429\ 0\ 1,85714285714286$$
   $$0,428571428571429\ 1,57142857142857\ 0,571428571428571\ 1\ 0,857142857142857$$
   $$0,571428571428571\ 2,42857142857143\ 2,42857142857143\ 1\ 0,142857142857143$$

2. Для каждого x^i вычисляю $\frac{\|x^i\|}{\|b^i\|}$: $\begin{matrix} 0,571428571428571 \\ 0,607142857142857 \\ 0,809523809523809 \\ 0,250000000000000 \\ 0,371428571428571 \end{matrix}$

3. Нахожу $\|A^{-1}\| \approx \max \frac{\|x^i\|}{\|b^i\|} = 0.809523809523809$

4. Вычисляю число обусловленности по формуле $cond(A) \approx \|A\| * \|A^{-1}\| = 3.238095238095238$

Код программы на MATLAB

```
A =  [3 1 0 0 0;
 1 2 1 0 0;
 0 1 1 1 0;
 0 0 1 0 1;
 0 0 0 1 1];
b = [1 2 3 4 5;
1 1 2 3 4;
1 2 1 2 3;
1 3 2 1 2;
1 4 3 2 1];
x = [];
norms = [];
normval = inf;
for i = 1:size(b, 2)
    C = rref([A b(:, i)]);
    xi = C(:,end);
    x = [x, xi];
    norms = [norms; norm(xi, normval)/norm(b(:, i), normval)];
end
NormAinv = max(norms);
NormA = norm(A, normval);
disp(NormA*NormAinv);
disp(cond(A, normval));
disp(norm(A, normval)*norm(inv(A), normval));
```

Задание 3.9.2 Решить систему уравнений $Ax = b$ порядка n из задачи 3.5 методом Холецкого. Вычислить число обусловленности матрицы A.
Вариант 7
$A_{i,j} = \frac{i+j}{m+n},\ i \neq j\ and\ n + m^2 + j/m + i/n\ i = j$
$n = 40, m = 8, b_i = 200 + 50 * i$
Решение:
В методе Холецкого мы представляем нашу матрицу $A = LL^T$, где L - нижнетреугольная матрица с положительными элементами на диагонали, при этом сама A должна быть симметричной положительно определенной.
Элементы матрицы L можно вычислить начиная с верхнего левого угла матрицы A по формулам:

$l_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} l_{ik}^2}$
$l_{ij} = \frac{1}{l_{jj}}(a_{ij} - \sum_{k=1}^{j-1} l_{ik}l_{jk}), \ j < i$
Получив такое разложение можно найти решение системы уравнений $Ax = b$ таким способом: нужно решить последовательно две треугольные системы уравнений $Ly = b$ и $L^T x = y$
Первое решеается методом прямой подстановки, то есть цепочкой вычислений, начиная с верхнего угла матрицы L по возрастанию номера строки i:
$y_1 = b_1$
$y_i = b_i - \sum_{j=1}^{i-1} l_{ij}y_j, \ i = 2, \cdots, n$
Второе решается методом обратной подстановки, где идет цепочка вычислений с нижнего угла матрицы L^T (для удобства назовем ее $U = L^T$) при убывании номера строки:
$x_n = y_n/u_{nn}$
$x_i = (y_i - \sum_{j=i+1}^{n} u_{ij}x_j)/u_{ii}, \ i = n-1, \cdots, 1$
Обратная же матрица находится из уравнения $CLL^T = E$, где E - единичная матрица, а C - искомая.
Вычисление происходит в 2 этапа, на первом этапе вычисляется матрица D исходя из уравнения $DL^T = E$ по таким формулам:
$d_{ii} = \frac{1}{L_{ii}^T}$
$d_{ij} = -(\sum_{k=i}^{j} d_{ik}L_{kj}^T)/d_{jj}$
$j = i, \cdots, n$
$i = 1, \cdots, n$
Искомая матрица C вычисляется на втором этапе исходя из уравнения $CL = D$ по формулам:
$c_{ij} = d_{ij} - \sum_{k=j}^{n} c_{ik}Lkj$
$i = 1, 2, \cdots, j$
$w_i = -\sum_{k=j}^{n} c_{ik}L_{kj}$
$c_{ij} = w_i i = j, j+1, \cdots, n)$
$j = n, \cdots, 1$
Реализующий задание код на MATLAB

```
n = 20;
m = 8;
b = zeros(n, 1);
A = zeros(n);
for i = 1:n
    for j = 1:n
        if i == j
            A(i, j) = n + m^2 + j/m + i/n;
        else
            A(i, j) = (i + j)/(m + n);
        end
    end
    b(i) = 200 + 50*i;
end
L = zeros(n);
for i = 1:n
    for j = 1:n
        lik = 0;
```

```
        for k = 1:j-1
            lik = lik + L(i, k)*L(j, k);
        end
        if i == j
            L(i, j) = sqrt(A(i, j) - lik);
        elseif j < i
            L(i, j) = 1/L(j, j)*(A(i, j) - lik);
        end
    end
end
y = zeros(n, 1);
y(1) = b(1);
for i = 2:n
    lij = 0;
    for j = 1:i-1
        lij = lij + L(i, j)*y(j);
    end
    y(i) = b(i) - lij;
end
U = L.';
x = zeros(n, 1);
x(n) = y(n)*U(n, n);
for i = n-1:-1:1
    uij = 0;
    for j = i+1:n
        uij = uij + U(i,j)*x(j);
    end
    x(i) = (y(i) - uij)/U(i, i);
end

D = zeros(n);
for i = 1:n
    D(i, i) = 1/U(i,i);
end
for i = 1:n
    for j = i:n
        if i ~= j
            dik = 0;
            for k = i:j
                dik = dik + D(i, k)*U(k,j);
            end
            D(i, j) = -(dik)/D(j, j);
        end
    end
end
C = zeros(n);
for j = n:-1:1
```

```
    for i = 1:j
        sum = 0;
        for k = j:n
            sum = sum + C(i, k)*L(k, j);
        end
        C(i, j) = (D(i, k) - sum);
    end
    for i = j:n
        sum = 0;
        for k = j:n
            sum = sum + C(i, k)*L(k, j);
        end
        C(i, j) = -sum;
    end
end
normval = inf;
disp(norm(A, normval) * norm(C, normval));
disp(norm(A, normval) * norm(inv(A), normval));
disp(cond(A, normval))
```