Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Высшая школа программной инженерии

Отчёт по лабораторной работе № 2

Метод Гаусса. LU разложение матрицы. Программы DECOMP и SOLVE

Выполнил
студент гр. в3530904/00030 В.С. Баганов

Руководитель
профессор, к.т.н. С.М. Устинов

«_____»_____ 202__ г.

Санкт-Петербург
2023

Содержание

1. Постановка задачи

Написать процедуру формирования матрицы A по заданному вектору B Задавая n=5 , a1 = 4 , a2 = 3 , a3 = 2 , a4 = var = 1.5 ; 1.01 ; 1.001 ; 1.0001 и вычисляя A^1 с помощью DECOMP и SOLVE , найти нормы матриц R = AA^1 - E для всех вариантов a4 .

A= $\begin{pmatrix} 1 & a_1 & a_1 & ... & a_1 \\ 1 & 1 & a_2 & ... & a_2 \\ ... & ... & ... & ... & ... \\ 1 & 1 & 1 & ... & a_{n-1} \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$

B = ($a_1, a_2, ..., a_{n-1})^T$

2. Подпрограммы DECOMP SOLVLE

Для эффективного нахождения обратной матрицы могут быть и спользованы программы DECOMP и SOLVE. Первая из них строит LU разложение, а вторая решает две системы (**). Программы имеют следующие параметры:

DECOMP(NDIM, N, A, COND, IPVT, WORK)

Здесь **NDIM** - объявленная в описании строчная размерность массива, в котором располагается матрица A;
N - порядок системы уравнений;
A -матрица, подвергающаяся разложению (по окончании работы программы на ее месте располагаются матрицы L и U);
COND - оценка числа обусловленности;
IPVT -вектор индексов ведущих элементов (размерность N);
WORK - рабочий одномерный массив (размерность N);
B - вектор правых частей системы, где по окончании работы программы **SOLVE** размещается вектор решения x.

3. Код программы

```
program lab2 Baganov
   use environment
   use DECOMPMOD
   use SOLVEMOD

   implicit none
   character(*), parameter     :: output_file = "output.txt"
   integer, parameter          :: ndim = 5
   integer                     :: Out = 0, j, IPVT(ndim), k, l, m!,i
   real(R_)    :: A(ndim, ndim), cpA(ndim, ndim), revA(ndim, ndim),
    ↪  E(ndim, ndim), cpE(ndim, ndim)
   real(R_) :: COND, WORK(ndim), R(ndim, ndim), B(ndim), norma(ndim)
   real(R_) :: R_ABS(ndim, ndim), NORMA_1, NORMA_INF
   real(R_) :: B_1(4) = [4.0, 3.0, 2.0, 1.5]
   real(R_) :: B_2(4) = [4.0, 3.0, 2.0, 1.01]
   real(R_) :: B_3(4) = [4.0, 3.0, 2.0, 1.001]
   real(R_) :: B_4(4) = [4.0, 3.0, 2.0, 1.0001]



   call makemat (A, B_1, E, ndim)
      cpA = A
      cpE = E
      call DECOMP(ndim, ndim, cpA, COND, IPVT, WORK)
      do j = 1, ndim
         B = cpE(:, j)
         call SOLVE(ndim, ndim, cpA, B, IPVT)
         revA(:, j) = B
      end do
      R = matmul(A,revA) - E
         do m = 1, ndim
            do l = 1, ndim
               R_ABS(l, m) = ABS(R(l, m))
            end do
         end do

         NORMA_1 = MAXVAL(SUM(R_ABS, DIM = 2))
         NORMA_INF = MAXVAL(SUM(R_ABS, DIM = 1))

      open (file=output_file, encoding=E_, newunit=Out,
       ↪  position='append')
         write (Out, "(a)") "Vector B:"
         write (Out, "("//ndim//"f9.6)") (B_1)
         write (Out, "(a)") "Matrix A:"
         write (Out, "("//ndim//"f6.3)") (A)
         write (Out, "(a)") "Matrix E:"
         write (Out, "("//ndim//"f6.3)") (E)
         write (Out, "(a, f19.9)") "COND: ", COND
         write (Out, "(a)") "Matrix R:"
         write (Out, "("//ndim//"f15.7)") (R(k, :), k = 1, ndim)
         write (Out, "(a)") "Matrix R_ABS:"
         write (Out, "("//ndim//"f15.7)") (R_ABS(k, :), k = 1, ndim)
         write (Out, "(a, f19.9)") "Norma_strings:", NORMA_1
         write (Out, "(a, f19.9)") "Norma_columns:", NORMA_INF
         write (Out, *)
      close (Out)
```

```
55
56
57       call makemat (A, B_2, E, ndim)
58          cpA = A
59          cpE = E
60          call DECOMP(ndim, ndim, cpA, COND, IPVT, WORK)
61          do j = 1, ndim
62             B = cpE(:, j)
63             call SOLVE(ndim, ndim, cpA, B, IPVT)
64             revA(:, j) = B
65          end do
66          R = matmul(A,revA) - E
67             do m = 1, ndim
68                do l = 1, ndim
69                   R_ABS(l, m) = ABS(R(l, m))
70                end do
71             end do
72
73             NORMA_1 = MAXVAL(SUM(R_ABS, DIM = 2))
74             NORMA_INF = MAXVAL(SUM(R_ABS, DIM = 1))
75          open (file=output_file, encoding=E_, newunit=Out,
            ↪  position='append')
76             write (Out, "(a)") "Vector B:"
77             write (Out, "("//ndim//"f9.6)") (B_2)
78             write (Out, "(a, f19.9)") "COND: ", COND
79             write (Out, "(a)") "Matrix R:"
80             write (Out, "("//ndim//"f15.7)") (R(k, :), k = 1, ndim)
81   !          write (Out, "(a)") "Matrix R_ABS:"
82   !          write (Out, "("//ndim//"f15.7)") (R_ABS(k, :), k = 1, ndim)
83             write (Out, "(a, f19.9)") "Norma_strings:", NORMA_1
84             write (Out, "(a, f19.9)") "Norma_columns:", NORMA_INF
85             write (Out, *)
86          close (Out)
87
88
89       call makemat (A, B_3, E, ndim)
90          cpA = A
91          cpE = E
92          call DECOMP(ndim, ndim, cpA, COND, IPVT, WORK)
93          do j = 1, ndim
94             B = cpE(:, j)
95             call SOLVE(ndim, ndim, cpA, B, IPVT)
96             revA(:, j) = B
97          end do
98          R = matmul(A,revA) - E
99             do m = 1, ndim
100               do l = 1, ndim
101                  R_ABS(l, m) = ABS(R(l, m))
102               end do
103            end do
104
105            NORMA_1 = MAXVAL(SUM(R_ABS, DIM = 2))
106            NORMA_INF = MAXVAL(SUM(R_ABS, DIM = 1))
107  !
108         open (file=output_file, encoding=E_, newunit=Out,
            ↪  position='append')
109            write (Out, "(a)") "Vector B:"
110            write (Out, "("//ndim//"f9.6)") (B_3)
111            write (Out, "(a, f19.9)") "COND: ", COND
112            write (Out, "(a)") "Matrix R:"
```

```
113            write (Out, "("//ndim//"f15.7)") (R(k, :), k = 1, ndim)
114    !        write (Out, "(a)") "Matrix R_ABS:"
115    !        write (Out, "("//ndim//"f15.7)") (R_ABS(k, :), k = 1, ndim)
116            write (Out, "(a, f19.9)") "Norma_strings:", NORMA_1
117            write (Out, "(a, f19.9)") "Norma_columns:", NORMA_INF
118            write (Out, *)
119         close (Out)
120
121
122
123      call makemat (A, B_4, E, ndim)
124         cpA = A
125         cpE = E
126         call DECOMP(ndim, ndim, cpA, COND, IPVT, WORK)
127         do j = 1, ndim
128            B = cpE(:, j)
129            call SOLVE(ndim, ndim, cpA, B, IPVT)
130            revA(:, j) = B
131         end do
132         R = matmul(A,revA) - E
133            do m = 1, ndim
134               do l = 1, ndim
135                  R_ABS(l, m) = ABS(R(l, m))
136               end do
137            end do
138
139            NORMA_1 = MAXVAL(SUM(R_ABS, DIM = 2))
140            NORMA_INF = MAXVAL(SUM(R_ABS, DIM = 1))
141         open (file=output_file, encoding=E_, newunit=Out,
           ↪  position='append')
142            write (Out, "(a)") "Vector B:"
143            write (Out, "("//ndim//"f9.6)") (B_4)
144            write (Out, "(a, f19.9)") "COND: ", COND
145            write (Out, "(a)") "Matrix R:"
146            write (Out, "("//ndim//"f15.7)") (R(k, :), k = 1, ndim)
147    !        write (Out, "(a)") "Matrix R_ABS:"
148    !        write (Out, "("//ndim//"f15.7)") (R_ABS(k, :), k = 1, ndim)
149            write (Out, "(a, f19.9)") "Norma_strings:", NORMA_1
150            write (Out, "(a, f19.9)") "Norma_columns:", NORMA_INF
151            write (Out, *)
152         close (Out)
153
154
155
156      contains
157
158      pure subroutine makemat(A, C, E, ndim)
159         real(R_), intent(inout)     :: A(:, :), E(:,:)
160         integer, intent(in)        :: ndim
161         real(R_), intent(in)        :: C(:)
162         integer                    :: i, j
163
164         do i = 1, ndim
165            do j = 1, ndim
166               if (i == j) then
167                  E(i, j) = 1
168               else
169                  E(i, j) = 0
170               end if
171               if (j ⩾ i) then
```

```
172                 A(i, j) = 1
173               else
174                 A(i, j) = C(j)
175               end if
176             end do
177           end do
178       end subroutine makemat
179   end program lab2 Baganov
```

Листинг 1: Код программы

4. Результат программы

```
Vector B:
 4.000000 3.000000 2.000000 1.500000
Matrix A:
 1.000 4.000 4.000 4.000 4.000
 1.000 1.000 3.000 3.000 3.000
 1.000 1.000 1.000 2.000 2.000
 1.000 1.000 1.000 1.000 1.500
 1.000 1.000 1.000 1.000 1.000
Matrix E:
 1.000 0.000 0.000 0.000 0.000
 0.000 1.000 0.000 0.000 0.000
 0.000 0.000 1.000 0.000 0.000
 0.000 0.000 0.000 1.000 0.000
 0.000 0.000 0.000 0.000 1.000
COND:        47.222221375
Matrix R:
0.0000000     -0.0000000      0.0000000      0.0000000      0.0000000
0.0000000     -0.0000001      0.0000000      0.0000000      0.0000000
0.0000000     -0.0000001      0.0000000      0.0000000      0.0000000
0.0000000     -0.0000001      0.0000000      0.0000000      0.0000000
0.0000000     -0.0000001      0.0000000      0.0000000      0.0000000
Matrix R_ABS:
0.0000000      0.0000000      0.0000000      0.0000000      0.0000000
0.0000000      0.0000001      0.0000000      0.0000000      0.0000000
0.0000000      0.0000001      0.0000000      0.0000000      0.0000000
0.0000000      0.0000001      0.0000000      0.0000000      0.0000000
0.0000000      0.0000001      0.0000000      0.0000000      0.0000000
Norma_strings:        0.000000075
Norma_columns:        0.000000298

Vector B:
 4.000000 3.000000 2.000000 1.010000
COND:      3374.609130859
Matrix R:
0.0000000     -0.0000000      0.0000000      0.0000000      0.0000000
0.0000000     -0.0000001      0.0000000      0.0000000      0.0000000
0.0000000     -0.0000001      0.0000000      0.0000000      0.0000000
0.0000000     -0.0000001      0.0000000      0.0000000      0.0000000
0.0000000     -0.0000001      0.0000000      0.0000000      0.0000000
Norma_strings:        0.000000075
Norma_columns:        0.000000298

Vector B:
 4.000000 3.000000 2.000000 1.001000
COND:     33972.921875000
Matrix R:
0.0000000      0.0000000      0.0000000     -0.0000610      0.0000610
0.0000000      0.0000000      0.0000000      0.0000000      0.0000000
0.0000000      0.0000000      0.0000000      0.0000000      0.0000000
0.0000000      0.0000000      0.0000000      0.0000000      0.0000000
0.0000000      0.0000000      0.0000000      0.0000000      0.0000000
Norma_strings:        0.000122085
Norma_columns:        0.000061035

Vector B:
 4.000000 3.000000 2.000000 1.000100
COND:    339918.093750000
```

```
Matrix R:
0.0000000      0.0000000      0.0000000      0.0000000      0.0000000
0.0000000      0.0000000      0.0000000      0.0000000      0.0000000
0.0000000      0.0000000      0.0000000      0.0000000      0.0000000
0.0000000      0.0000000      0.0000000      0.0000000      0.0000000
0.0000000      0.0000000      0.0000000      0.0000000      0.0000000
Norma_strings:         0.000000045
Norma_columns:         0.000000149
```

Листинг 2: Результат программы

Таблица 4.1

Сравнение данных

A_4	COND	NormaStrings	NormaColumns
1.5	47.222221375	-0.000000015	0.000000000
1.01	3374.609130859	0.000000075	0.000000298
1.001	33972.921875000	0.000122085	0.000061035
1.0001	339918.093750000	0.000000045	0.000000149

5. Заключение

При увеличении количества знаков после запятой норма матрицы и число обусловленности увеличиваются, т.е. матрица становится хуже обусловленной.