

Министерство образования и науки Российской Федерации
Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий
Кафедра «Информационная безопасность компьютерных систем»

**ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2**

по дисциплине «Вычислительная математика»

Выполнил
студент гр. 23508/4

Е.Г. Проценко

Проверил
профессор

С.М. Устинов

Санкт-Петербург
2016

1 ФОРМУЛИРОВКА ЗАДАНИЯ (ВАРИАНТ 29)

Составьте процедуру вычисления по заданной матрице $A(N, N)$ матрицы $R = A^{-1}A - E$ и ее нормы $\|R\| = \max \sum_j^N |R_{jk}|$.

Построить три матрицы A при $x_k = \frac{1+\cos(k)}{\sin^2(k)}$, $k = 1, \dots, 4$ и $x_5 = (1 + \cos(1))/\sin^2(1 + \varepsilon)$ для трех значений $\varepsilon = 0.001, 0.00001, 0.000001$ и $N = 5$.

$$A = \begin{pmatrix} 1 & \dots & 1 \\ \vdots & \ddots & \vdots \\ x_1^{N-1} & \dots & x_N^{N-1} \end{pmatrix}$$

Исследовать зависимость погрешности вычисления $\|R\|$ от ε .

2 РЕЗУЛЬТАТЫ РАБОТЫ

- 1) Построены три матрицы A при $x_k = \frac{1+\cos(k)}{\sin^2(k)}$, $k = 1, \dots, 4$ и $x_5 = (1 + \cos(1))/\sin^2(1 + \varepsilon)$ для трех значений $\varepsilon = 0.001, 0.00001, 0.000001$ и $N = 5$.

Input Matrix A With e = 0.001

1.00000000000	1.00000000000	1.00000000000	1.00000000000	1.00000000000
2.17534264970	0.70614146372	0.50251445764	0.60472521853	2.17255396670
4.73211564350	0.49863576678	0.25252078014	0.36569258993	4.71999073810
10.29397298300	0.35210739022	0.12689534287	0.22114353136	10.25443460100
22.39291846400	0.24863762791	0.06376674440	0.13373107033	22.27831256700

Input Matrix B With e = 0.00001

1.00000000000	1.00000000000	1.00000000000	1.00000000000	1.00000000000
2.17534264970	0.70614146372	0.50251445764	0.60472521853	2.17531471470
4.73211564350	0.49863576678	0.25252078014	0.36569258993	4.73199410810
10.29397298300	0.35210739022	0.12689534287	0.22114353136	10.29357641300
22.39291846400	0.24863762791	0.06376674440	0.13373107033	22.39176823900

Input Matrix C With e = 0.000001

1.00000000000	1.00000000000	1.00000000000	1.00000000000	1.00000000000
2.17534264970	0.70614146372	0.50251445764	0.60472521853	2.17533985610
4.73211564350	0.49863576678	0.25252078014	0.36569258993	4.73210348970
10.29397298300	0.35210739022	0.12689534287	0.22114353136	10.29393332400
22.39291846400	0.24863762791	0.06376674440	0.13373107033	22.39280343700

- 2) Cond для всех трех матриц.

COND Matrix A:47703.99809900000
COND Matrix B:3019691.20310000000
COND Matrix C:30047556.46200000000

Чем хуже обусловлена матрица, тем больше значение cond.

- 3) Обратные матрицы для всех трех.

A:

43.30786309300	-239.06246678000	466.84223460000	-370.27812097000	-370.27812097000
32.27933370700	-147.31066369000	221.25623503000	-122.60982682000	-122.60982682000
34.70866903200	-138.47981171000	186.64788018000	-97.32209473000	-97.32209473000
-65.69999521900	284.22637798000	-404.92966424000	217.68687328000	217.68687328000
-43.59587061400	240.62656420000	-469.81668557000	372.52316923000	372.52316923000

B:

4328.81519360000	-23892.83724600000	46650.18074000000	-36989.58976800000	-36989.58976800000
32.25960492000	-147.20177103000	221.04362513000	-122.44124594000	-122.44124594000
34.69540883000	138.40662228000	-186.50497985000	-97.20878737800	-97.20878737800
-65.66782582300	284.04881962000	-404.58298648000	217.41198842000	217.41198842000
-4329.10238150000	23894.39682000000	-46653.14635800000	36991.82781300000	36991.82781300000

C:

43287.87342500000	-238926.60523000000	466497.64228000000	-369891.57425000000	-369891.57425000000
32.25943718400	-147.20084521000	221.04181751000	-122.43981265000	-122.43981265000
34.69529911000	-138.40601669000	186.50379744000	-97.20784983100	-97.20784983100
-65.66755582200	284.04732936000	-404.58007678000	217.40968128000	217.40968128000
-43288.16060500000	238928.16476000000	-466500.60782000000	369893.81223000000	369893.81223000000

4) Вычисление матрицы R из всех трех.

От матрицы A:

0.00000000000	-0.00000000006	-0.00000000007	0.00000000017	0.00000000017
0.00000000047	-0.00000000027	-0.00000000013	-0.00000000012	-0.00000000012
0.00000000093	-0.00000000008	-0.00000000005	-0.00000000011	-0.00000000011
-0.00000000186	0.00000000025	0.00000000009	0.00000000017	0.00000000017
0.00000000000	-0.00000000035	0.00000000004	-0.00000000007	-0.00000000007

От матрицы B:

0.00000023842	0.00000000373	0.00000000373	0.00000002235	0.00000002235
-0.00000000233	-0.00000000033	-0.00000000022	-0.00000000021	-0.00000000021
0.00000000559	0.00000000042	0.00000000024	0.00000000042	0.00000000042
-0.00000000466	-0.00000000047	-0.00000000024	-0.00000000035	-0.00000000035
-0.00000047684	-0.00000001863	-0.00000001490	-0.00000001118	-0.00000001118

От матрицы C:

0.00000190735	-0.00000041723	-0.00000023842	-0.00000020862	-0.00000020862
0.00000000279	0.00000000039	0.00000000030	0.00000000039	0.00000000039
0.00000000186	0.00000000023	0.00000000006	0.00000000010	0.00000000010
-0.00000000093	-0.00000000009	-0.00000000016	-0.00000000013	-0.00000000013
-0.00000190735	-0.00000002980	0.00000003725	0.00000000000	0.00000000000

5) Норма для матриц R.

A_R Matrix Norm: 4.1909515858E-09

B_R Matrix Norm: 7.5995922089E-07

C_R Matrix Norm: 6.5863132477E-06

Чем хуже обусловлена исходная матрица, тем больше норма матрицы R.

3 ВЫВОД

4 ПРИЛОЖЕНИЕ

Листинг написанной программы:

```
uses CRT, FMM, MATH;

Const
    N = 5;
    ea = 0.001;
    eb = 0.00001;
    ec = 0.000001;

Var
    f: text;

    i, j: integer;
    xk: float;
    A, B, C: floatmatrix;
    _A, _B, _C: floatmatrix;
    A_I, B_I, C_I: floatmatrix;
    A_R, B_R, C_R: floatmatrix;
    bb: floatvector;

    cond: float;
    ipvt: ivector;
    work: floatvector;

procedure print_matrix(MTRX: floatmatrix);
Var
    i, j: integer;
begin
    for i:=1 to N do begin
        for j:=1 to N - 1 do begin
            write(MTRX[i,j]:16:15, ' ');
            write(f, MTRX[i,j]:16:15, ' ');
        end;
        writeln(MTRX[i,N]:16:15);
        writeln(f, MTRX[i,j]:16:15, ' ');
    end;
end;

procedure create_inverse_matrix(MTRX: floatmatrix; Var MTRX_I: floatmatrix);
Var
    i, j: integer;
    bb: floatvector;
begin
    for j:=1 to N do begin
        for i:=1 to N do bb[i]:=0;
        bb[j] := 1;

        solve(N, MTRX, bb, ipvt);

        for i:=1 to N do begin
            MTRX_I[i, j] := bb[i];
        end;
    end;
end;

procedure create_r_matrix(MTRX: floatmatrix; MTRX_I: floatmatrix;
    Var MTRX_R: floatmatrix);
```

```

Var
    i, j, k: integer;
    sum: float;
begin
    for i:=1 to N do begin
        for j:=1 to N do begin

            sum := 0;

            for k:=1 to N do begin
                sum := sum + MTRX_I[i, k] * MTRX[k, j];
            end;

            MTRX_R[i, j] := sum;
            if i = j then MTRX_R[i, j] := MTRX_R[i, j] - 1;

        end;
    end;
end;

function get_norm(MTRX: floatmatrix) : float;
Var
    max_norm, norm: float;
begin
    max_norm := 0;
    for i:=1 to N do begin
        norm:=0;
        for j:=1 to N do begin
            if MTRX[i, j] > 0 then norm := norm + MTRX[i, j]
            else norm := norm - MTRX[i, j];
        end;
        if max_norm < norm then max_norm := norm;
    end;
    get_norm := max_norm;
end;

begin
    assign(f, 'output_3.txt');
    rewrite(f);

    clrscr;

    {Matrix Builder}

    for j:=1 to N - 1 do begin
        xk := (1+cos(j))/power(sin(j), 2);
        for i:=1 to N do begin
            A[i, j] := power(xk, i - 1);
            B[i, j] := power(xk, i - 1);
            C[i, j] := power(xk, i - 1);
        end;
    end;

    xk := (1 + cos(1))/power(sin(1 + ea), 2);
    for i:=1 to N do begin
        A[i, N] := power(xk, i-1);
    end;

    xk := (1 + cos(1))/power(sin(1 + eb), 2);
    for i:=1 to N do begin
        B[i, N] := power(xk, i-1);
    end;
end;

```

```

xk := (1 + cos(1))/power(sin(1 + ec), 2);
for i:=1 to N do begin
    C[i, N] := power(xk, i-1);
end;

writeln('Input Matrix A With e = 0.001');
print_matrix(A);
readln;

writeln('Input Matrix B With e = 0.00001');
print_matrix(B);
readln;

writeln('Input Matrix C With e = 0.000001');
print_matrix(C);
readln;

{Copying}

for i:=1 to N do begin
    for j:=1 to N do begin
        _A[i,j] := A[i,j];
        _B[i,j] := B[i,j];
        _C[i,j] := C[i,j];
    end;
end;

{Inverse Matrix}

decomp(N, _A, cond, ipvt, work);
writeln('COND Matrix A:', cond:13:12);
create_inverse_matrix(_A, A_I);

decomp(N, _B, cond, ipvt, work);
writeln('COND Matrix B:', cond:13:12);
create_inverse_matrix(_B, B_I);

decomp(N, _C, cond, ipvt, work);
writeln('COND Matrix C:', cond:13:12);
create_inverse_matrix(_C, C_I);

readln;

writeln('Inverse Matrix A');
print_matrix(A_I);
readln;

writeln('Inverse Matrix B');
print_matrix(B_I);
readln;

writeln('Inverse Matrix C');
print_matrix(C_I);
readln;

{R}

create_r_matrix(A, A_I, A_R);
writeln('Matrix R from A');
print_matrix(A_R);
readln;

create_r_matrix(B, B_I, B_R);
writeln('Matrix R from B');

```

```

    print_matrix(B_R);
    readln;

    create_r_matrix(C, C_I, C_R);
    writeln('Matrix R from C');
    print_matrix(C_R);
    readln;

    {R Matrix Norm}

    writeln('A_R Matrix Norm: ', get_norm(A_R));
    readln;

    writeln('B_R Matrix Norm: ', get_norm(B_R));
    readln;

    writeln('C_R Matrix Norm: ', get_norm(C_R));
    readln;

    close(f);
end.

```