1. Министерство образования и науки Российской Федерации
2. Санкт-Петербургский политехнический университет Петра Великого
3. —
4. Институт компьютерных наук и технологий
5. **Кафедра «Информационная безопасность компьютерных систем»**

**ОТЧЕТ**

**по лабораторной работе № 2**

1. по дисциплине «Вычислительная математика»
2. Выполнил
3. студент гр. 23508/4 Е.Г. Проценко
4. Проверил
5. профессор С.М. Устинов
6. Санкт-Петербург
7. 2016

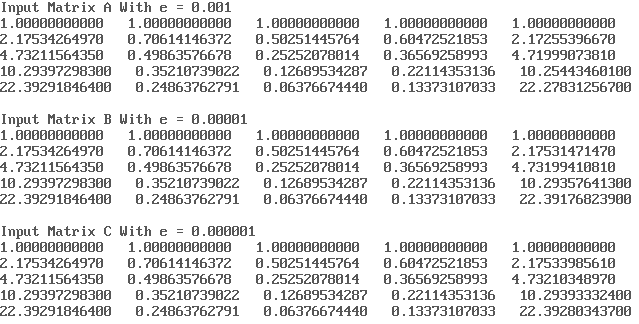
# Формулировка задания (Вариант 29)

Составьте процедуру вычисления по заданной матрице матрицы и ее нормы .

Построить три матрицы A при и для трех значений ε = 0.001, 0.00001, 0.000001 и .

Исследовать зависимость погрешности вычисления от ε.

# Результаты работы

1. Построены три матрицы A при и для трех значений ε = 0.001, 0.00001, 0.000001 и .
2. Cond для всех трех матриц.



Чем хуже обусловлена матрица, тем больше значение cond.

1. 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:

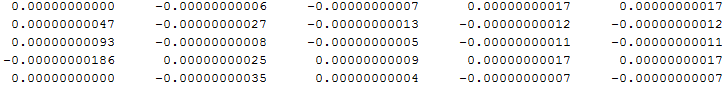
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4328.81519360000 | -23892.83724600000 | 46650.18074000000 | -36989.58976800000 | -36989.58976800000 |
| 32.25960492000 | -147.20177103000 | 221.04362513000 | -122.44124594000 | -122.44124594000 |
| 34.69540883000 | 138.40662228000 | -186.50497985000 | -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:

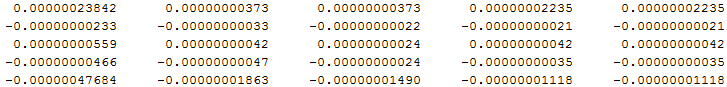
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 43287.87342500000 | -238926.60523000000 | 466497.64228000000 | -369891.57425000000 | -369891.57425000000 |
| 32.25943718400 | -147.20084521000 | 221.04181751000 | -122.43981265000 | -122.43981265000 |
| 34.69529911000 | -138.40601669000 | 186.50379744000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

1. Вычисление матрицы R из всех трех.

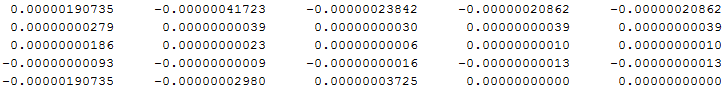
От матрицы A:



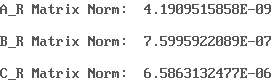
От матрицы B:



От матрицы C:



1. Норма для матриц R.



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

# 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;

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.