ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧЕРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Санкт-Петербургский политехнический университет Петра Великого»

Институт компьютерных наук и технологий
Высшая школа программной инженерии

Отчет по лабораторной работе по дисциплине «Вычислительная математика»

Выполнила студентка гр. 3530904/80001 Прохорова А. И.

Руководитель Устинов С. М.

Санкт-Петербург
2020

1 Задание

1. Составить процедуру вычисления матрицы A вида

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & \cdots & x_N \\ \cdots & \cdots & \cdots & \cdots \\ x_1^{N-1} & x_2^{N-1} & \cdots & x_N^{N-1} \end{pmatrix}$$

2. Используя её, решить несколько раз линейную систему $A \cdot z = b$, где
   $N = 5, 7, 9$
   $b_k = 2^k + cos(k)\ k = 1...N$
   $x_k = k, k = 1...N-1$ и $x_N = 1.1,\ 1.01,\ 1.001,\ 1.0001$.
   Системы решать, используя програмы **DECOMP** и **SOLVE**. Сделать необходимые выводы.

2 Результаты работы программы

```
N = 5
B: 2.5403 ; 3.58385 ; 7.01001 ; 15.3464 ; 32.2837 ;
X: 1 ; 2 ; 3 ; 4 ; 1.1 ;
Original matrix:
1 ; 1 ; 1 ; 1 ; 1 ;
1 ; 2 ; 3 ; 4 ; 1.1 ;
1 ; 4 ; 9 ; 16 ; 1.21 ;
1 ; 8 ; 27 ; 64 ; 1.331 ;
1 ; 16 ; 81 ; 256 ; 1.4641 ;

Cond: 53842.3
Answers: 12.4967 ; 1.74069 ; 0.371622 ; -0.080522 ; -11.9882 ;

X: 1 ; 2 ; 3 ; 4 ; 1.01 ;
Original matrix:
1 ; 1 ; 1 ; 1 ; 1 ;
1 ; 2 ; 3 ; 4 ; 1.01 ;
1 ; 4 ; 9 ; 16 ; 1.0201 ;
1 ; 8 ; 27 ; 64 ; 1.0303 ;
1 ; 16 ; 81 ; 256 ; 1.0406 ;

Cond: 427806
Answers: 3621.52 ; 48.4691 ; -14.8044 ; 2.27776 ; -3644.97 ;

X: 1 ; 2 ; 3 ; 4 ; 1.001 ;
Original matrix:
1 ; 1 ; 1 ; 1 ; 1 ;
1 ; 2 ; 3 ; 4 ; 1.001 ;
1 ; 4 ; 9 ; 16 ; 1.002 ;
1 ; 8 ; 27 ; 64 ; 1.003 ;
1 ; 16 ; 81 ; 256 ; 1.00401 ;

Cond: 4.18039e+06
Answers: 1.33988e+07 ; 18985.5 ; -5992.99 ; 945.52 ; -1.34091e+07 ;

X: 1 ; 2 ; 3 ; 4 ; 1.0001 ;
Original matrix:
1 ; 1 ; 1 ; 1 ; 1 ;
1 ; 2 ; 3 ; 4 ; 1.0001 ;
1 ; 4 ; 9 ; 16 ; 1.0002 ;
1 ; 8 ; 27 ; 64 ; 1.0003 ;
1 ; 16 ; 81 ; 256 ; 1.0004 ;
```

```
Cond: 4.17074e+07
Answers: 5.11709e+11 ; 7.33241e+07 ; -2.33114e+07 ; 3.69792e+06 ; -5.11749e+11 ;
--------------------------------------

N = 7
B: 2.5403 ; 3.58385 ; 7.01001 ; 15.3464 ; 32.2837 ; 64.9602 ; 128.754 ;
X: 1 ; 2 ; 3 ; 4 ; 5 ; 6 ; 1.1 ;
Original matrix:
1 ; 1 ; 1 ; 1 ; 1 ; 1 ; 1 ;
1 ; 2 ; 3 ; 4 ; 5 ; 6 ; 1.1 ;
1 ; 4 ; 9 ; 16 ; 25 ; 36 ; 1.21 ;
1 ; 8 ; 27 ; 64 ; 125 ; 216 ; 1.331 ;
1 ; 16 ; 81 ; 256 ; 625 ; 1296 ; 1.4641 ;
1 ; 32 ; 243 ; 1024 ; 3125 ; 7776 ; 1.61051 ;
1 ; 64 ; 729 ; 4096 ; 15625 ; 46656 ; 1.77156 ;

Cond: 1.99768e+07
Answers: 5.12857 ; -1.97223 ; 3.52866 ; -1.91276 ; 0.59341 ; -0.0804839 ; -2.74487 ;

X: 1 ; 2 ; 3 ; 4 ; 5 ; 6 ; 1.01 ;
Original matrix:
1 ; 1 ; 1 ; 1 ; 1 ; 1 ; 1 ;
1 ; 2 ; 3 ; 4 ; 5 ; 6 ; 1.01 ;
1 ; 4 ; 9 ; 16 ; 25 ; 36 ; 1.0201 ;
1 ; 8 ; 27 ; 64 ; 125 ; 216 ; 1.0303 ;
1 ; 16 ; 81 ; 256 ; 625 ; 1296 ; 1.0406 ;
1 ; 32 ; 243 ; 1024 ; 3125 ; 7776 ; 1.05101 ;
1 ; 64 ; 729 ; 4096 ; 15625 ; 46656 ; 1.06152 ;

Cond: 1.33154e+08
Answers: 12076.6 ; 392.728 ; -287.833 ; 151.449 ; -46.8948 ; 6.38835 ; -12287.3 ;

X: 1 ; 2 ; 3 ; 4 ; 5 ; 6 ; 1.001 ;
Original matrix:
1 ; 1 ; 1 ; 1 ; 1 ; 1 ; 1 ;
1 ; 2 ; 3 ; 4 ; 5 ; 6 ; 1.001 ;
1 ; 4 ; 9 ; 16 ; 25 ; 36 ; 1.002 ;
1 ; 8 ; 27 ; 64 ; 125 ; 216 ; 1.003 ;
1 ; 16 ; 81 ; 256 ; 625 ; 1296 ; 1.00401 ;
1 ; 32 ; 243 ; 1024 ; 3125 ; 7776 ; 1.00501 ;
1 ; 64 ; 729 ; 4096 ; 15625 ; 46656 ; 1.00602 ;

Cond: 1.27261e+09
Answers: 6.17589e+07 ; 145535 ; -94226.4 ; 46299.3 ; -13725.2 ; 1813.8 ; -6.18325e+07
     ;

X: 1 ; 2 ; 3 ; 4 ; 5 ; 6 ; 1.0001 ;
Original matrix:
1 ; 1 ; 1 ; 1 ; 1 ; 1 ; 1 ;
1 ; 2 ; 3 ; 4 ; 5 ; 6 ; 1.0001 ;
1 ; 4 ; 9 ; 16 ; 25 ; 36 ; 1.0002 ;
1 ; 8 ; 27 ; 64 ; 125 ; 216 ; 1.0003 ;
1 ; 16 ; 81 ; 256 ; 625 ; 1296 ; 1.0004 ;
1 ; 32 ; 243 ; 1024 ; 3125 ; 7776 ; 1.0005 ;
1 ; 64 ; 729 ; 4096 ; 15625 ; 46656 ; 1.0006 ;

Cond: 1.26679e+10
Answers: 3.66356e+12 ; 9.1174e+08 ; -6.06081e+08 ; 3.02368e+08 ; -9.05356e+07 ; 1.205
    e+07 ; -3.66403e+12 ;
--------------------------------------

N = 9
```

```
B: 2.5403 ; 3.58385 ; 7.01001 ; 15.3464 ; 32.2837 ; 64.9602 ; 128.754 ; 255.854 ;
    511.089 ;
X: 1 ; 2 ; 3 ; 4 ; 5 ; 6 ; 7 ; 8 ; 1.1 ;
Original matrix:
1 ; 1 ; 1 ; 1 ; 1 ; 1 ; 1 ; 1 ; 1 ;
1 ; 2 ; 3 ; 4 ; 5 ; 6 ; 7 ; 8 ; 1.1 ;
1 ; 4 ; 9 ; 16 ; 25 ; 36 ; 49 ; 64 ; 1.21 ;
1 ; 8 ; 27 ; 64 ; 125 ; 216 ; 343 ; 512 ; 1.331 ;
1 ; 16 ; 81 ; 256 ; 625 ; 1296 ; 2401 ; 4096 ; 1.4641 ;
1 ; 32 ; 243 ; 1024 ; 3125 ; 7776 ; 16807 ; 32768 ; 1.61051 ;
1 ; 64 ; 729 ; 4096 ; 15625 ; 46656 ; 117649 ; 262144 ; 1.77156 ;
1 ; 128 ; 2187 ; 16384 ; 78125 ; 279936 ; 823543 ; 2.09715e+06 ; 1.94872 ;
1 ; 256 ; 6561 ; 65536 ; 390625 ; 1.67962e+06 ; 5.7648e+06 ; 1.67772e+07 ; 2.14359 ;

Cond: 1.50774e+10
Answers: -3.27322 ; -8.04376 ; 11.4984 ; -9.8966 ; 5.99565 ; -2.4051 ; 0.572881 ;
    -0.0613482 ; 8.1534 ;

X: 1 ; 2 ; 3 ; 4 ; 5 ; 6 ; 7 ; 8 ; 1.01 ;
Original matrix:
1 ; 1 ; 1 ; 1 ; 1 ; 1 ; 1 ; 1 ; 1 ;
1 ; 2 ; 3 ; 4 ; 5 ; 6 ; 7 ; 8 ; 1.01 ;
1 ; 4 ; 9 ; 16 ; 25 ; 36 ; 49 ; 64 ; 1.0201 ;
1 ; 8 ; 27 ; 64 ; 125 ; 216 ; 343 ; 512 ; 1.0303 ;
1 ; 16 ; 81 ; 256 ; 625 ; 1296 ; 2401 ; 4096 ; 1.0406 ;
1 ; 32 ; 243 ; 1024 ; 3125 ; 7776 ; 16807 ; 32768 ; 1.05101 ;
1 ; 64 ; 729 ; 4096 ; 15625 ; 46656 ; 117649 ; 262144 ; 1.06152 ;
1 ; 128 ; 2187 ; 16384 ; 78125 ; 279936 ; 823543 ; 2.09715e+06 ; 1.07214 ;
1 ; 256 ; 6561 ; 65536 ; 390625 ; 1.67962e+06 ; 5.7648e+06 ; 1.67772e+07 ; 1.08286 ;

Cond: 7.58843e+10
Answers: 58162.5 ; 2860.09 ; -3232.45 ; 2876.8 ; -1798.08 ; 739.525 ; -179.663 ;
    19.546 ; -59451.5 ;

X: 1 ; 2 ; 3 ; 4 ; 5 ; 6 ; 7 ; 8 ; 1.001 ;
Original matrix:
1 ; 1 ; 1 ; 1 ; 1 ; 1 ; 1 ; 1 ; 1 ;
1 ; 2 ; 3 ; 4 ; 5 ; 6 ; 7 ; 8 ; 1.001 ;
1 ; 4 ; 9 ; 16 ; 25 ; 36 ; 49 ; 64 ; 1.002 ;
1 ; 8 ; 27 ; 64 ; 125 ; 216 ; 343 ; 512 ; 1.003 ;
1 ; 16 ; 81 ; 256 ; 625 ; 1296 ; 2401 ; 4096 ; 1.00401 ;
1 ; 32 ; 243 ; 1024 ; 3125 ; 7776 ; 16807 ; 32768 ; 1.00501 ;
1 ; 64 ; 729 ; 4096 ; 15625 ; 46656 ; 117649 ; 262144 ; 1.00602 ;
1 ; 128 ; 2187 ; 16384 ; 78125 ; 279936 ; 823543 ; 2.09715e+06 ; 1.00702 ;
1 ; 256 ; 6561 ; 65536 ; 390625 ; 1.67962e+06 ; 5.7648e+06 ; 1.67772e+07 ; 1.00803 ;

Cond: 6.91747e+11
Answers: 2.80625e+08 ; 748040 ; -634808 ; 474151 ; -263013 ; 99161.8 ; -22543 ;
    2327.17 ; -2.8097e+08 ;

X: 1 ; 2 ; 3 ; 4 ; 5 ; 6 ; 7 ; 8 ; 1.0001 ;
Original matrix:
1 ; 1 ; 1 ; 1 ; 1 ; 1 ; 1 ; 1 ; 1 ;
1 ; 2 ; 3 ; 4 ; 5 ; 6 ; 7 ; 8 ; 1.0001 ;
1 ; 4 ; 9 ; 16 ; 25 ; 36 ; 49 ; 64 ; 1.0002 ;
1 ; 8 ; 27 ; 64 ; 125 ; 216 ; 343 ; 512 ; 1.0003 ;
1 ; 16 ; 81 ; 256 ; 625 ; 1296 ; 2401 ; 4096 ; 1.0004 ;
1 ; 32 ; 243 ; 1024 ; 3125 ; 7776 ; 16807 ; 32768 ; 1.0005 ;
1 ; 64 ; 729 ; 4096 ; 15625 ; 46656 ; 117649 ; 262144 ; 1.0006 ;
1 ; 128 ; 2187 ; 16384 ; 78125 ; 279936 ; 823543 ; 2.09715e+06 ; 1.0007 ;
1 ; 256 ; 6561 ; 65536 ; 390625 ; 1.67962e+06 ; 5.7648e+06 ; 1.67772e+07 ; 1.0008 ;

Cond: 6.8511e+12
```

```
Answers: 2.20643e+13 ; 7.67775e+09 ; -7.65622e+09 ; 6.36991e+09 ; -3.81793e+09 ;
    1.52604e+09 ; -3.63142e+08 ; 3.88912e+07 ; -2.20678e+13 ;
```

3 Выводы

Для всех вариантов матриц значение обусловленности очень большое (> 50000). Следовательно, доверять полученным результатам мы не можем.

4 Приложение

4.1 Код lab2.hpp

```
#ifndef COMPUTATIONAL_MATH_LAB2
#define COMPUTATIONAL_MATH_LAB2

#include "matrix.hpp"
#include "../cmath.h"

void fillMatrixOfVandermond(Matrix<double>& matrix, const std::vector<double>& x);
void fillXvector(std::vector<double>& x, double start, double end);
void solveSystem(std::vector<double>& x, std::vector<double>& b, std::ostream& out);
void solveSystem(int size, std::ostream& out);
#endif // COMPUTATIONAL_MATH_LAB2
```

4.2 Код lab2.cpp

```
#include "lab2.hpp"

#include <iostream>
#include <algorithm>
#include <numeric>

#include "matrix.hpp"

void fillMatrixOfVandermond(Matrix<double>& matrix, const std::vector<double>& x)
{
  if (matrix.size() != x.size())
  {
    throw std::invalid_argument("Size of matrix douesn't match the size of x vector.");
  }
  int exp = 0;
  for (auto it = matrix.begin(); it != matrix.end(); ++exp, it += matrix.size())
  {
    std::transform(x.begin(), x.end(), it,
        [=](double x){ return pow(x, exp); });
  }
}

void fillBvector(std::vector<double>& b)
{
  int k = 0;
  std::generate(b.begin(), b.end(), [&](){++k; return pow(2, k) + cos(k);});
}

void solveSystem(std::vector<double>& x, std::vector<double>& b, std::ostream& out)
{
  out << "\nX: ";
```

```
  std::copy(x.begin(), x.end(), std::ostream_iterator<double>(out, " ; "));

  size_t size = x.size();
  Matrix<double> matrix(size);
  fillMatrixOfVandermond(matrix, x);

  out << "\nOriginal matrix:\n" << matrix;

  double cond = 0;
  int flag = 0;

  std::vector<int> pivot(size);
  decomp(size, size, matrix.data(), &cond, pivot.data(), &flag);
  if (flag != 0)
  {
    throw std::runtime_error("Error occured in program decomp.\n");
  }

  out << "\nCond: " << cond;

  solve(size, size, matrix.data(), b.data(), pivot.data());
  std::cout << "\nAnswers: ";
  std::copy(b.begin(), b.end(), std::ostream_iterator<double>(std::cout, " ; "));
  std::cout << "\n";
}

void solveSystem(int size, std::ostream& out)
{
  std::vector<double> b(size);
  fillBvector(b);

  out << "\nN = " << size
      << "\nB: ";
  std::copy(b.begin(), b.end(), std::ostream_iterator<double>(out, " ; "));

  std::vector<double> x(size);
  std::iota(x.begin(), x.end(), 1);
  double divider = 1;
  while (divider *= 10, divider != 100000)
  {
    x.back() = 1 + 1 / divider;
    solveSystem(x, b, out);
  }
}

int main()
{
  solveSystem(5, std::cout);
  std::cout << "---------------------------------------\n";
  solveSystem(7, std::cout);
  std::cout << "---------------------------------------\n";
  solveSystem(9, std::cout);
  return 0;
}
```

4.3 Код matrix.hpp

```
#ifndef COMPUTATIONAL_MATH_LAB2_MATRIX
#define COMPUTATIONAL_MATH_LAB2_MATRIX

#include <vector>
#include <iosfwd>
```

```cpp
#include <cmath>
#include <stdexcept>
#include <iterator>

template<typename T>
class Matrix
{
public:
  using iterator = typename std::vector<T>::iterator;
  using const_iterator = typename std::vector<T>::const_iterator;
  class Row
  {
  public:
    Row(T* begin, size_t size);
    T& operator[](size_t index);
  private:
    T* begin_;
    size_t size_;
  };

  Matrix(size_t size);
  Row operator[](size_t index);
  iterator begin();
  iterator end();
  const_iterator begin() const;
  const_iterator end() const;
  size_t size() const;
  void resize(size_t size);
  T* data();

private:
  std::vector<T> data_;
};

template<typename T>
Matrix<T>::Row::Row(T* begin, size_t size):
  begin_(begin),
  size_(size)
{
}
template<typename T>
T& Matrix<T>::Row::operator[](size_t index)
{
  if (index > size_)
  {
    throw std::invalid_argument("Index is out of bounds.");
  }
  return begin_[index];
}
template<typename T>
typename Matrix<T>::Row Matrix<T>::operator[](size_t index)
{
  return Row(&data_[0] + index * size(), size());
}
template<typename T>
Matrix<T>::Matrix(size_t size):
  data_(size * size)
{
}
template<typename T>
typename Matrix<T>::const_iterator Matrix<T>::begin() const
{
  return data_.begin();
```

```
}
template<typename T>
typename Matrix<T>::const_iterator Matrix<T>::end() const
{
  return data_.end();
}
template<typename T>
typename Matrix<T>::iterator Matrix<T>::begin()
{
  return data_.begin();
}
template<typename T>
typename Matrix<T>::iterator Matrix<T>::end()
{
  return data_.end();
}
template<typename T>
size_t Matrix<T>::size() const
{
  return sqrt(data_.size());
}
template<typename T>
void Matrix<T>::resize(size_t size)
{
  data_.resize(size * size);
}
template<typename T>
T* Matrix<T>::data()
{
  return &data_[0];
}

template <typename T>
std::ostream& operator<<(std::ostream& out, Matrix<T>& matrix)
{
  for (auto it = matrix.begin(); it != matrix.end(); it += matrix.size())
  {
    std::copy(it, it + matrix.size(), std::ostream_iterator<T>(out, " ; "));
    out << "\n";
  }
  return out;
}

#endif // COMPUTATIONAL_MATH_LAB2_MATRIX
```