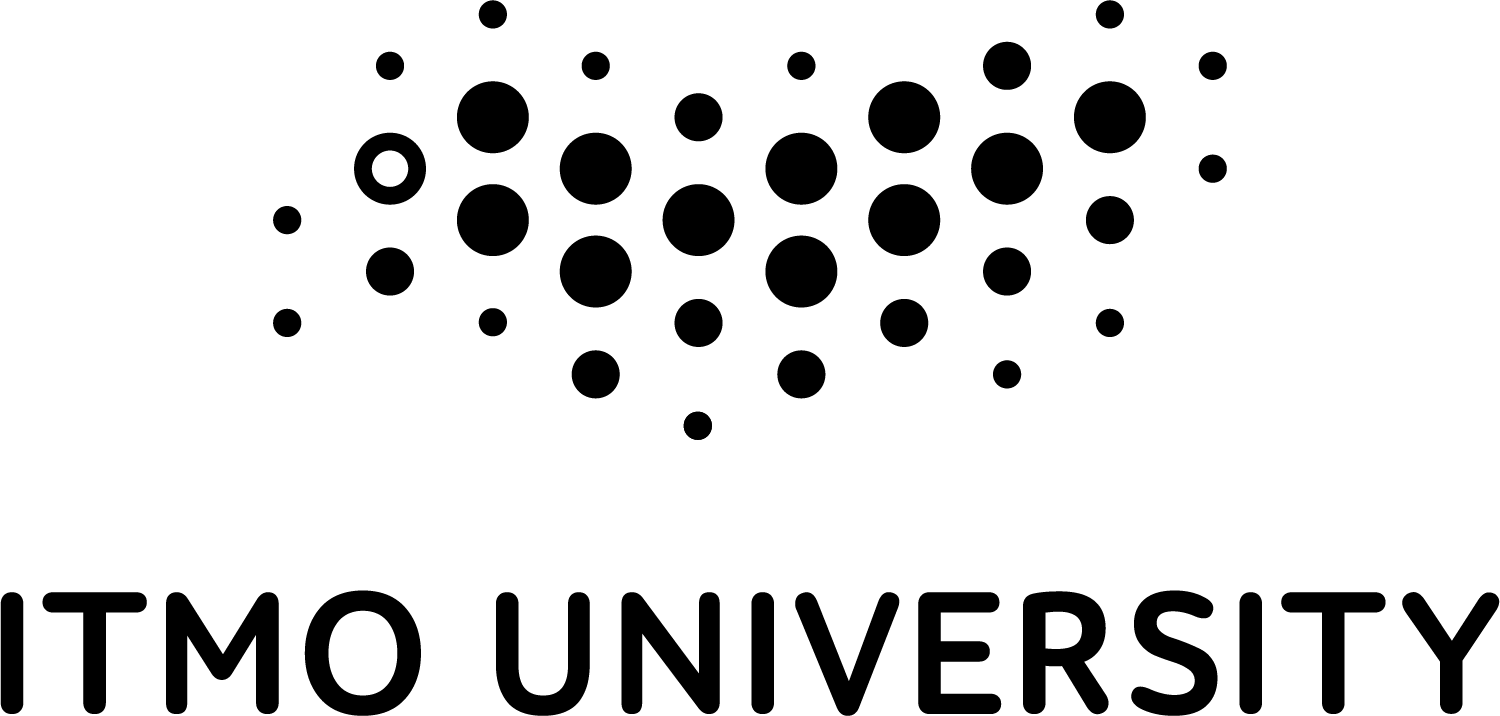
Лабораторная работа №3

Методы решения СЛАУ

**Курдюков Кирилл, Стрельников Илья, Харёв Павел**

**Группа M3237**



Университет ИТМО

# Постановка задания

* 1. Реализовать прямой метод решения СЛАУ на основе LU-разложения с учетом следую- щих требований:
     + формат матрицы профильный
     + размерность матрицы, элементы матрицы и вектор правой части читать из файлов, результаты записывать в файл
     + в программе резервировать объём памяти, необходимый для хранения в нем только одной матрицы и необходимого числа векторов
     + элементы матрицы обрабатывать в порядке, соответствующем формату хранения
  2. Провести исследование реализованного метода на матрицах, число обусловленности ко- торых регулируется за счёт изменения диагонального преобладания (то есть оценить влияние увеличения числа обусловленности на точность решения)
     + Для каждого *k*, для которого система вычислительно разрешима, оценить погреш- ность найденного решения.
     + Для одного из значений *k* попытаться найти операцию, вызывающую скачкообраз- ное накопление погрешности, пояснить полученные результаты.
  3. Провести аналогичные исследования на матрицах Гильберта различной размерности. Матрица Гильберта размерности k строится следующим образом:

*aij* = 1 *, i, j* = 1*..k*

*i*+*j−*1

* 1. Реализовать метод Гаусса с выбором ведущего элемента для **плотных матриц**. Срав- нить метод Гаусса по точности получаемого решения и по количеству действий с реали- зованным прямым методом LU-разложения.
  2. Реализовать метод сопряженных градиентов для решения СЛАУ, матрица которых хра- нится в **разреженном строчно-столбцовом формате** и является симметричной.

Точность решения СЛАУ задавать как минимум 10*−*7.

1. Протестировать разработанную программу. Для тестирования использовать матри- цы небольшой размерности, при этом вектор правой части формировать умноже- нием тестовой матрицы на заданный вектор.
2. Провести исследование реализованного метода на матрице с диагональным преоб- ладанием. Для каждого полученного решения с помощью невязки и погрешности

оценить число обусловленности *cond*(*A*) *≥ ||x−x∗|| / ||f−Ax||*

*||x∗|| ||f||*

1. Провести аналогичные п. (b) исследования на матрице с обратным знаком внедиа- гональных элементов
2. Повторить аналогичные п. (b) исследования для плотной матрицы Гильберта для различных размерностей (размерность *n* для СЛАУ выбирать от 10 до 103).

# Вычислительная схема алгоритмов

a11*x*1 + *a*12*x*2 + *..* + *a*1*nxn* = *b*1



a21*x*1 + *a*22*x*2 + *..* + *a*2*nxn* = *b*2

*...*



a*n*1*x*1 + *an*2*x*2 + *..* + *annxn* = *bn*

*Ax* = *b A* = (*aij*)

*b* = (*b*1*, b*2*, .., bn*)

*x* = (*x*1*, x*2*, .., xn*)

## Метод Гаусса с выбором ведущего элемента

* + - Прямой ход метода Гаусса Основная идея метода заключается в приведении матрицы к треугольному виду путём последовательного исключения неизвестных.:



a11*x*1 + *a*12*x*2 + *a*13*x*3 + *..* + *a*1*nxn* = *b*1

a22*x*2 + *..* + *a*23*x*3 + *a*2*nxn* = *b*2

*k*-номер этапа

...



a*kkxk* + *...* + *aknxn* = *bk*

(*k*) (*k−*1)

*a*(*k−*1)

(*k−*1)

*aij* = *aij − ik ∗ akj*

*kk*

*a*(*k−*1)

(*k*)

(*k−*1)

*a*(*k−*1)

(*k−*1)

*b* = *b ik b*

*— ∗i i k*

*a*(*k−*1)

*kk*

* + - Обратный ход метода Гаусса После приведения матрицы коэффициентов к верхнему тре- угольному виду становится возможным определение значений неизвестных. Из послед- него уравнения преобразованной системы может быть вычислено значение переменной *xn*, после этого из предпоследнего уравнения *xn−*1 и т.д *xn* = *bn/ann*

Далее для всех *k* = *n −* 1*..*1

*xk* = 1 (*b*(*k−*1) *−* P*n ak−*1*xj* )

*a*

(*k−*1)

*kk*

*k*

*j*=*k*+1

*kj*

## Метод Гаусса на основе LU-разложения

*Метод Гаусса*, использующий представление матрицы в виде нижнетреугольной матрицы и верхнетреугольной матрицы. *Ax* = *b*

*LU* (L - нижнетреугольная матрица, U - верхнетреугольная

*LUx* = *b y* = *Ux Ly* = *b*

Алгоритм действий:

* + - Получить из матрицы *A* нижнетреугольную матрицу *L* и верхнетреугольную матрицу

*U* .

* + - Решить уравнение *Ly* = *b* - прямой ход, найти *y*.
    - Обратный ход *Ux* = *y*, находи *x*.

Интерфейс, опписывающий поведение и свойства точки из пространства R*n* такие, как координаты, норму и алгебраические операции.

public interface Point { double getResult();

void setResult(double result);

static double getResult(Point p, ToDoubleFunction<Point> f) { if (Double.isNaN(p.getResult())) {

p.setResult(f.applyAsDouble(p));

}

return p.getResult();

}

double getX(int i);

void setX(int i, double x);

List<Double> getCoordinates();

void setCoordinates(List<Double> coordinates); int getDim();

double getNorm2(); double getNorm();

Point sum(Point p); Point sub(Point p); Point mul(double x);

double dotProduct(Point p);

}