# Лабораторная №3 по предмету методы оптимизации университет ИТМО.

Группа: М3237

Команда: пацаны на отSOSе

Участники: Курдюков Кирилл Алексеевич, Харёв Павел Андреевич, Стрельников Илья Денисович.

1 Постановка задачи

1. Реализовать прямой метод решение СЛАУ на основе LU – разложения с учетом следующих требований:

* Формат матрицы профильный
* Размерность матрицы, элементы матрицы и вектор правой части из файлов, результаты записывать в файл
* В программе резервировать объем памяти, необходимый для хранения в нем только одной матрицы и необходимого числа векторов
* Элементы матрицы обрабатывать в порядке, соответствующем формату хранения

1. Провести исследования реализованного метода на матрицах, число обусловленности которых регулируется за счет изменения диагонального преобладания (то есть оценить влияние увеличения числа обусловленности на точность решения)

* Для каждого k, для которого система вычислительно разрешима, оценить погрешность найденного решения.
* Для одного из значений k попытаться найти операцию, вызывающую скачкообразное накопление погрешности, пояснить полученные результаты.

1. Провести аналогичные исследования на матрицах Гильберта различной размерности. Матрица Гильберта размерности k строится следующим образом: 
2. Реализовать метод Гаусса с выбором ведущего элемента для плотных матриц. Сравнить метод Гаусса по точности получаемого решения и по количеству действий с реализованным прямым методом LU – разложения.
3. Реализовать метод сопряженных градиентов для решения СЛАУ, матрица которых хранится в **разреженном строчно – столбцовом** и является симметричной.

Точность решения СЛАУ задавать как минимум 10 ^ -7.

* Протестировать разработанную программу. Для тестирования использовать матрицы небольшой размерности, при этом вектор правой части формировать умножением тестовой матрицы на заданный вектор.
* Провести исследование реализованного метода на матрице с диагональным преобладанием. Для каждого полученного решения с помощью невязки и погрешности оценить число обусловленности 
* Провести аналогичные пункту выше исследования на матрице с обратным знаком внедиагональных элементов
* Повторить аналогичные пункту выше исследования для плотной матрицы Гильберта для различных размерностей (размерность n для СЛАУ выбирать от 10 до 10 ^ 3).

2 Форматы хранения матриц

Далее будет считаться, что матрицы А имеет размеры , а для элементов матрицы  будет использована индексация с 1.

2.1 Плотный формат

В **плотном формате** обычно хранятся матрицы малых размеров, либо матрицы с неизвестной структурой. Элементы хранятся в обычном двумерном массиве matrix[][]. Для матрицы A элементы  будет в элементе matrix[i – 1][j – 1]. Формат занимает  памяти.

2.2 Профильный формат

**Профильный формат** хранения матрицы предполагают, что все достаточно удаленные от диагонали элементы – нули, что помогает в некоторых случаях сэкономить память.

Матрица A представляется в виде четырех массивов:

* Массив диагональных элементов – вещественный массив длины n, в котором i - ый элемент равен значения матрицы 
* Портрет матрицы - целочисленный массив длины n + 1, в котором значения элементов определяются рекурсивно p[0] = 0 и p[i] = p[i-1] + (i-ki), где  , либо ki = i, если такого j не существует. Добавляемо значение , равно длине профиле i-ый строки (столбца) – минимального суффикса, содержащего ненулевые элементы.
* Массивы значений верхнего и нижнего треугольников upperColumns[] и lowerRows[] – массивы длины такие, что для  значения определяются, как lowerRoes[profile[i-1] + (j-ki)] = aij и upperColumns[profile[i-1] + (j-ki) = aji, таким образом, эти массивы хранят в себе подряд идущие профили соответствующего треугольника.

Таким образом, все ненулевые элементы матрицы будут содержаться в этих массивах. Понятно, что данный формат оптимальнее плотного, если ненулевые элементы сосредоточены около диагонали. В противном случае матрица может занять больше памяти, чем плотная.

2.3 Симметричный разряженный формат

Данный формат является улучшением прошлого. Он дополнительно предполагает, что матрица А часто имеет нулевые элементы между ненулевыми. Опишем данный формат для симметричной матрицы.

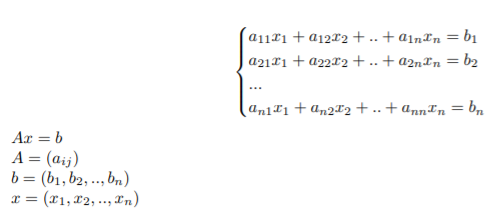
Аналогично прежнему формату, диагональ ранится в массиве diag[]. Массив profile[] имеет несколько отличающееся рекурсивное определение: profile[0] = 0;

profile[i] = profile[i-1] + ci, где ci = {j < i; aij != 0}. Так как матрицы симметричная, нам не понадобится массив upperColumns[]. Тогда lowerRows[profile[i-1] + k – 1] = bik для 1 <= k <= ci, где bik – kый ненулевой элемент iой строки матрицы.

Для доступа к данным элементам также понадобится массив nonNullIndex[] размера profile[n]. Пусть в массив lowerRows[] по индексу k был сохранен элемент аij (то есть lowerRows[k] = aij), тогда положим в nonNullIndices[k] = j-1.

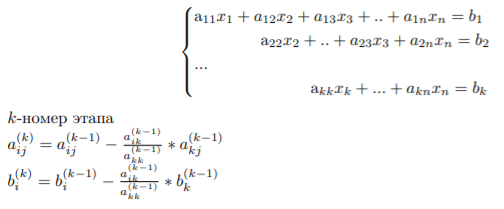
Понятно, что для плотных матриц разреженный формат будет занимать еще больше памяти, чем профильный. Но чем более разрежена матрица, то есть чем больше нулевых элементов, тем оптимальнее использовать именно этот формат.

Вычислительная схема алгоритмов 3



3.1 Метод Гаусса с выбором ведущего элемента

* Прямой ход метода Гаусса. Основная идея метода заключается в приведении матрицы к треугольному виду путем последовательно исключения неизвестных:



* Обратный ход метода Гаусса. После приведения матрицы коэффициентов к верхнему треугольному виду становится возможным определение значений неизвестных. Двумя способами:

1. Из последнего уравнения преобразованной системы может быть вычислено значение переменной xn, после этого из предпоследнего уравнения xn-1 и т.д 

Далее для все k = n – 1…1



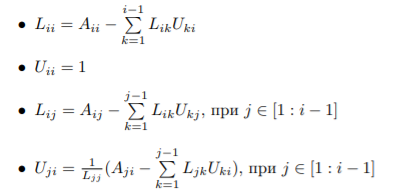
1. Можно преобразовать в диагональный вид и тогда xi = bi / aii

В ходе вычисления методом Гаусса возможны ситуации, когда элемент матрицы aii = 0. В этом случае для корректной работы алгоритма следует переставить строки матрицы. Однако среди возможных элементов могут оказаться **маленькие по абсолютной величине элементы**, тогда в ходе деления на такие элементы получается **большая погрешность вычислений.**

Чтобы избежать сильного влияния вычислительной погрешности на решение в каждом шаге в качестве ведущей выбирается строка с **наибольшим по модулю ведущим элементом.**

3.2 LU - разложение

Разложение A = LU на произведение нижне-треугольной матрицы и верхне-треугольной матрицы производилось по формулам:



3.2 Метод Гаусса на основе LU – разложения

Метод Гаусса, использующий представление матрицы в виде произведения нижне-треугольной матрицы и верхне-треугольной матрицы. Ах = b

LU (L – нижне-треугольная матрица, U – верхне-треугольная

LUx = b

y = Ux

Ly = b

Алгоритм действий:

* Получать из матрицы А нижне-треугольную матрицу L и верхне-треугольную матрицу.
* Решить уравнение Ly = b – прямой ход, найти y.
* Обратный ход Ux = y, находим x.

3.4 Метод сопряженных градиентов для решения СЛАУ

Для решения СЛАУ с симметричной положительно определенной матрицей А можно использовать методы многомерной оптимизации, а именно метод сопряженных градиентов. Рассмотрим функцию:



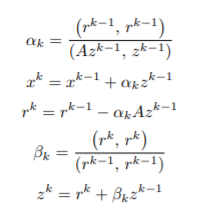
Эта функция является квадратичной и достигает минимума в некоторой точке min. Минимум достигается, когда обнуляется градиент , а это происходит в единственной точке x\*, как решение исходной СЛАУ Ax = b, а значит x\* = min. Поэтому поиск решения системы можно свести к минимизации приведенной квадратичной функции f(x).

**3.4.1 Описание алгоритма**

* Подготовка перед итерационным процессом:

****

* k – ая итерация метода:

****

* Критерий остановки.

В теории метод сопряженных градиентов для квадратичных функций завершается за n итераций, где n – размерность квадратичной функции f и, по совместительству, размер матрицы А, но из – за точности представления вещественных чисел может потребоваться и больше.

**3.4.2 Время работы алгоритма**

Наиболее дорогая операция умножения матрицы А на вектор производится на каждой итерации ровно один раз (вычисляем один раз и используем результат). Соответственно количество операций на каждой итерации в худшем случае будет составлять О(n^2)

Дополнительно для метода сопряженных градиентов известно, что на квадратичных функциях он завершается за <= n итераций. Поэтому худшее время работы метода сопряженных градиентов составляет О(n^3).

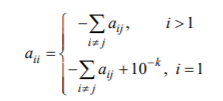
Стоит заметить, что для хорошо обусловленных функций данный метод совершает в разы меньше итераций. Из выводов лаб2.

4 Результаты вычислений

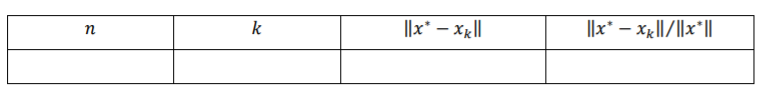
Далее будем считать, что x\* - решение заданной СЛАУ, x – найденное методом решение.

4.1 Метод Гаусса на основе LU – разложения

**4.1.1 Матрицы с диагональным преобладанием**

****



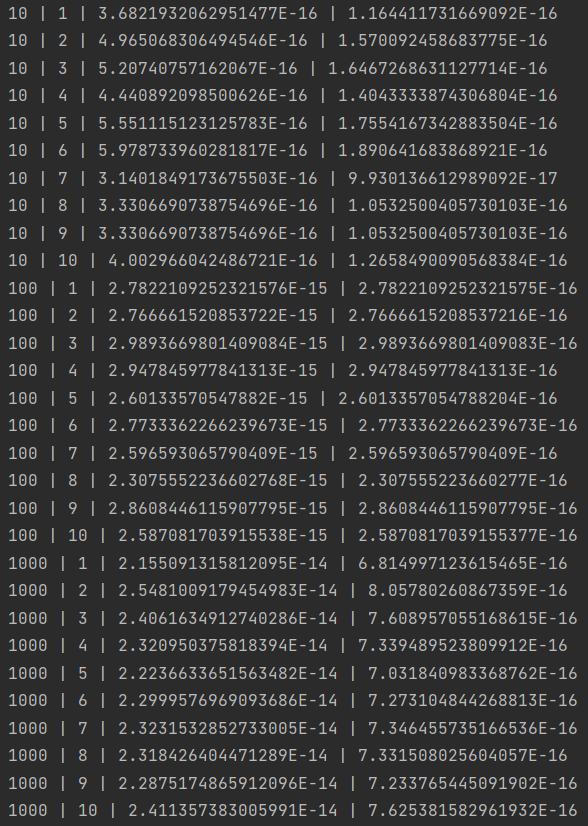
****

Результаты ниже.

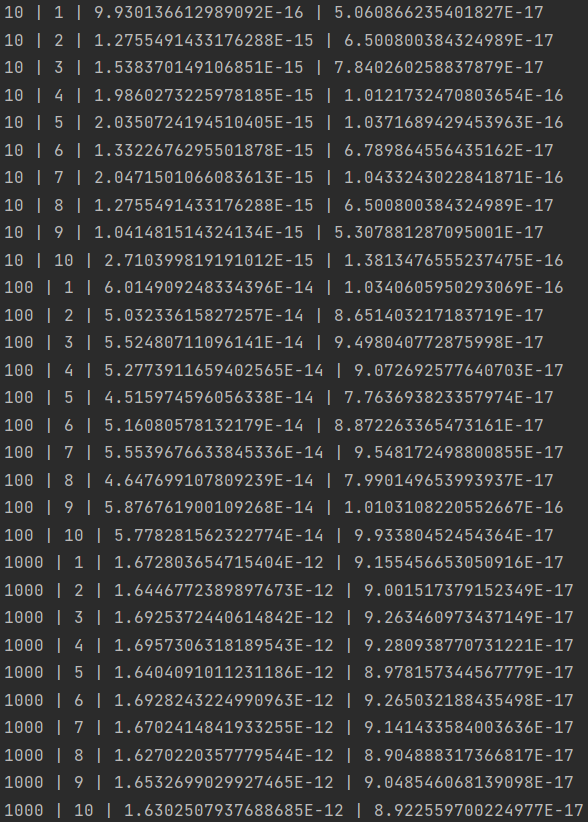
Для каждого k были проведены 10 итераций, погрешность усреднена. Можно заметить, что число обусловленности для каждой из таких матриц будет порядка 10 ^ k. Из – за этого погрешность также растет экспоненциально при увеличении k.

Можно заметить, что при увеличении n и k мы отклоняемся от ответа.

Брался вектор x\* = (1…1) размерности n



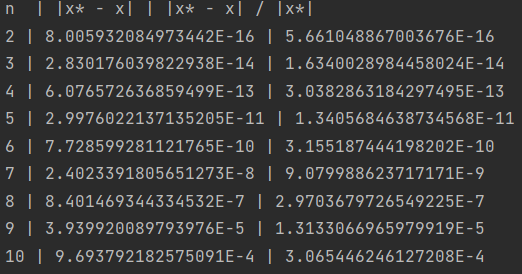
Для вектора x\* = (1 … n)

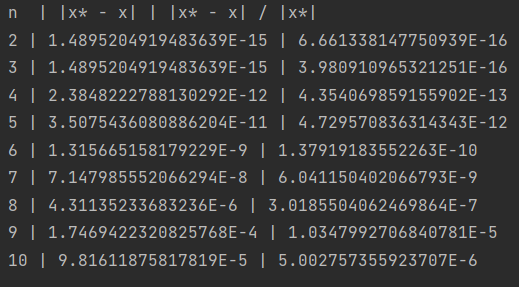


**4.1.2 Матрицы Гильберта.**

Для матриц Гилберта известно, что число обусловленности растет экспоненциально (с экспонентой равной . Из – за этого даже на маленьких размерностях метод показывает плохие результаты.

Вектор x\* = (1…1) 1 таблица и x\* = (1…n) 2 таблица

****

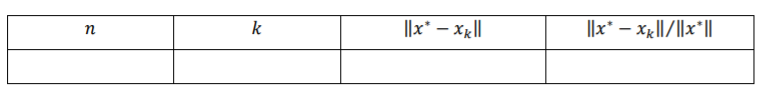
****

4.2 Метод Гаусса с выбором ведущего элемента

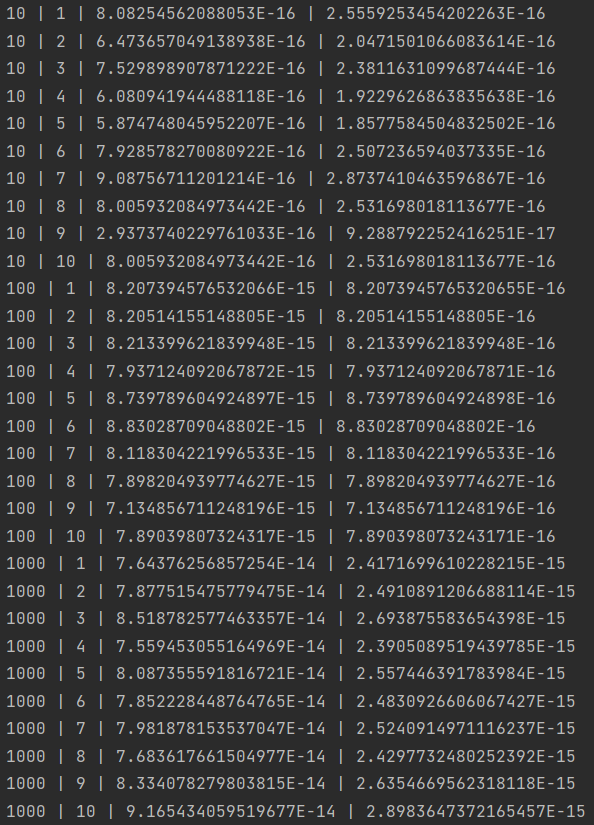
Для данного метода исследование проводилось аналогично прошлому методу. Комментарии для прошлых результатов также справедливы и здесь

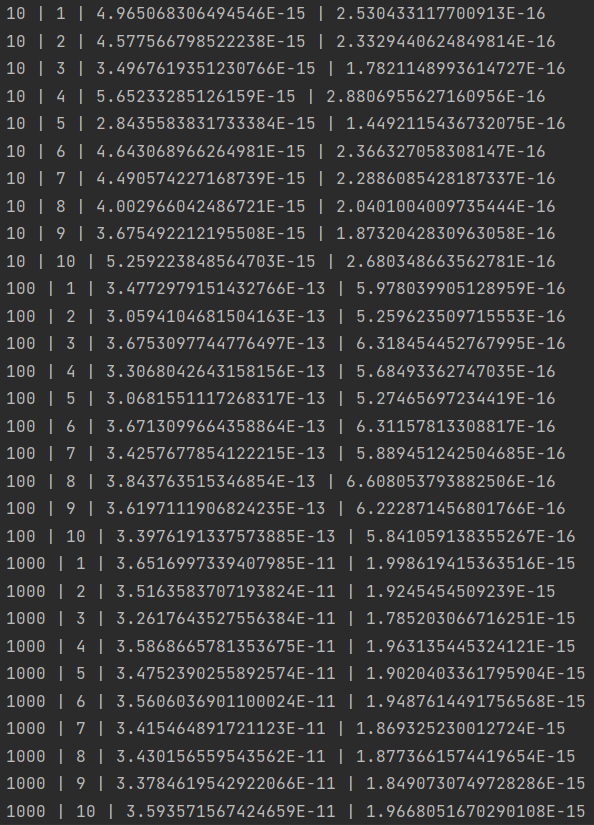
**4.2.1 Матрицы с диагональным преобладанием**

Eps = 10 ^ -7 брался для сравнения с 0.

****

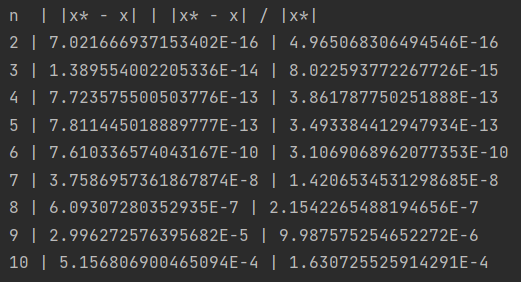
Соответственно для x\* = (1…1) и x\* = (1 … n)

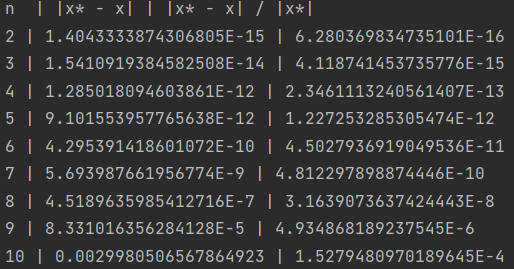
****

****

**4.2.2 Матрицы Гильберта**

Eps брался = 10 ^ -12. Чтобы хотя бы для n = 10 компенсировать плохую обусловленность.



****

4.3 Метод сопряженных градиентов