**Формулировка задания:**

Решить СЛАУ LU-разложением и исследовать зависимость возмущения (*норма разности исходного решения и решения полученного при помощи LU-разложения*) от числа обусловленности матрицы

**Алгоритм:**

Решение СЛАУ при помощи LU-разложение состоит из двух итераций

**LU-разложение:**

Предпошлем теорему:

Если все главные миноры матрицы отличны от нуля, то матрицу можно представить в виде произведения двух матриц единственным образом, если элементы главной диагонали одной из полученных матриц фиксированные(не нулевые)

Остановимся на случае когда:

L: U:

/ \ / \

| 1 0 …….………….0| |……….…|

| 1…..…………….0| | 0....……….|

|...................................| |...................................|

|…….………..1| | 0 0 ..……….…|

\ / \ /

Найдем элементы матриц:

(\*)

(\*\*)

Шаг 1: вычисляем по (\*) kую строку U

Шаг 2: вычисляем по (\*\*) kый столбец L(без первого элемента)

**Решение:**

представим матрицу А в виде произведения двух lower и upper, тогда, если исходное уравнение было Ах = b, то оно перепишется в виде LUx = b или введя вспомогательный вектор y:

| Ly = b

{

| Ux = y

вычислим вектор y:

вычислим вектор решений(*начинать надо с конца*):

**Анализ задачи + проверка условий:**

В требование к разложению - главные миноры не равны нулю, чтобы удовлетворить этому требованию я генерирую диагональную матрицу с ненулевыми элементами на главной диагонали, умножаю ее на ортогональную транспонированную матрицу слева и справа на ту же не транспонированную матрицу. Дальше будет необходимо генерировать матрицы с определенным числом обусловленности: определенного числа обусловленности можно добиться меня один диагональный элемент, при этом держа один равный 1

**Тестовый пример:**

A = [1 1 1; 1 2 3; 1 3 4]

b = [6; 14; 19]

Найдем LU разложение:

первая итерация:

Шаг 1:

Шаг 2:

вторая итерация:

Шаг 1:

Шаг 2:

третья итерация:

Шаг 1:

Решение:

Шаг 1:

8

5

Шаг 2:

**Перечень контрольных тестов:**

Чтобы получить зависимость на графике ниже(***численный анализ решения***) матрицы изменялись и генерировались как описано выше(***анализ задачи + проверка условий***) с некоторым шагом(*разница между ближайшими числами обусловленности*)

**Модульная структура программы:**

программа состоит из библиотеки работы с матрицами(определение типов: матрица, вектор, работа с памятью: выделение, удаление памяти, выделенной матрицы, вектора, простые операции: перемножение матриц, перемножение матрицы и вектора и т.д.)

matrix\_t genA(**int** size, **int** cond);

size размер матрицы

cond число обусловленности

возвращает матрицу с подходящими условиями и числом обусловленности равным cond

matrix\_t LU\_decomposition(matrix\_t matrix, **int** size);

matrix матрица, которую нужно разложить в произведение lower и upper

size размер входной матрицы

возвращает матрицу, в которую записаны lower и upper

vector\_t genroots(**int** size);

генерирует вектор решений размера size

vector\_t getrightpart(matrix\_t matrix, vector\_t roots, **int** size);

генерирует правую часть уравнения на основании roots(корней) и matrix(матрицы) размера size

vector\_t solve(matrix\_t matrix, vector\_t b, **int** size);

решает матричное уравнение с матрицей matrix и правой частью b размера size

возвращает вектор-решение

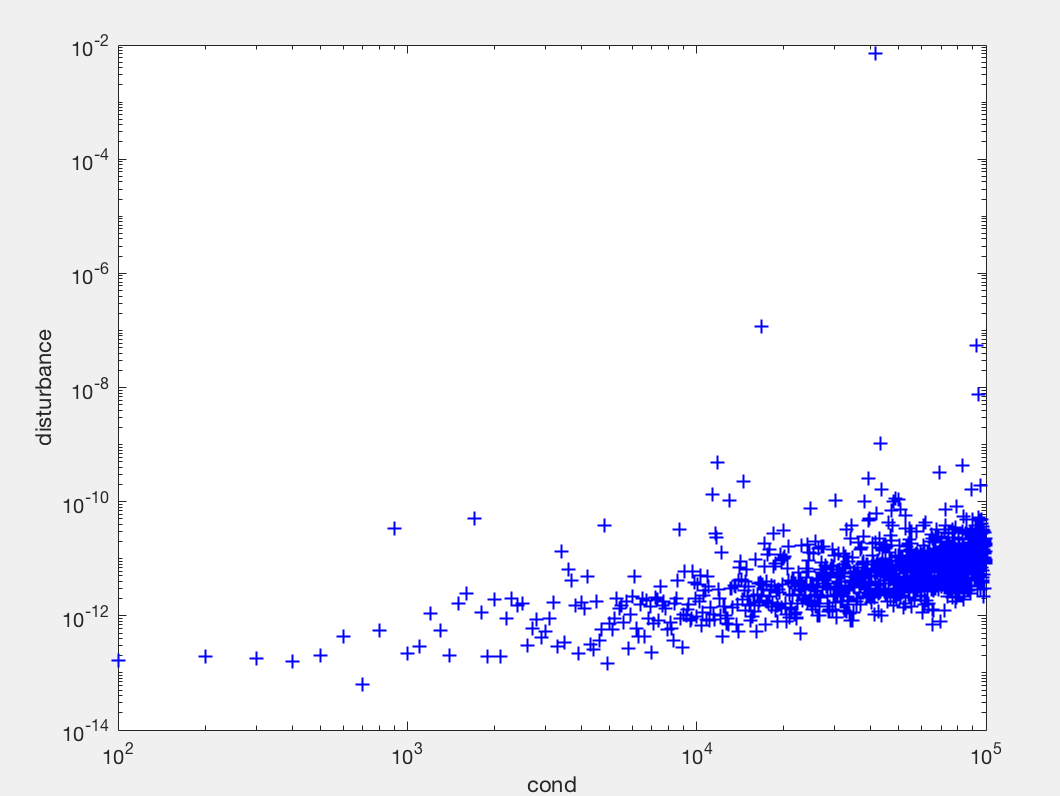
**int** main(**int** argc, **char**\* argv[]);

записывает данные в файл

function main()

считывает данные из файла и выводит на экран зависимость

**Численный анализ решения:**



большинство матриц показывают предсказанное возмущение(норма разности решения найденного при помощи LU-разложения и исходного решения)

но есть матрицы, которые я приложил в файлах: corruption\_type\_one.txt, corruption\_type\_two.txt

возможно причина одна и та же - проблемы с представлением чисел в компьютере

в первом файле матрицы получаются очень близкими к разреженным, это сказывается на lu-разложении(элементы lu матрицы получаются слишком большими)

во втором файле матрицы менее разрежены, но при этом матрицы из первого и второго файла имеют общие черты: во всех матрицах есть два элемента, превосходящие на несколько порядков остальные. Скорее всего это получается из-за выбранного способа изменения числа обусловленности и выбранной ортогональной матрицы(можно было бы добавить функции генерации ортогональных матриц, но это будет добавлено только к 3 лабе)

**Вывод:**

метод достаточно простой в реализации. Самая сложная(медленная) часть - представление в виде произведения двух матриц