**Федеральное государственное автономное образовательное учреждение высшего образования**

**“Санкт-Петербургский политехнический университет Петра Великого”**

Кафедра “Прикладная математика”

Дисциплина “Численные методы”

Отчет по учебной практике

Студента Хрипункова Д.В.

Группы 3630102/80001

Руководитель практики: Добрецова С.Б.

Санкт-Петербург 2019 г.

1. Формулировка задачи

Рассмотреть матричное уравнение вида Ax = b (систему линейных алгебраических уравнений) и найти решение методом LU-разложения.

По результатам решения необходимо проанализировать данные и найти зависимости точности решений от числа обусловленности матрицы через разность с точным решением ||x\* - x||(cond(A)) и нахождение невязки ||Ax – b||(cond(A)). После нужно внести ошибку в вектор b, найти решение и зависимость относительной погрешности (||x\* - x|| / ||x\*||) от внесённой ошибки в вектор b. Ошибку возьмём как случайное изменение значений в пределах 5% процентов.

2. Постановка задачи

В данной работе решается задача нахождения решения СЛАУ вида

Которое можно преобразовать в матричный вид как *Ax = b*, где b = (b1, b2, …, bn) – вектор свободных членов, x = (x1, x2, …, xn) – вектор неизвестных, – вещественная матрица коэффициентов размером n x n.

Чтобы существовало решение такой системы, необходимо и достаточно, чтобы определитель матрицы A не был равен нулю. Решение находится прямым методом, то есть за конечное число арифметических операций.

3. Алгоритм метода

В основе метода лежит представление исходной матрицы A как произведения двух матриц *A = LU*, где L – нижняя треугольная матрица, а U – верхняя треугольная матрица. Исходную систему *Ax = b* можно представить в виде *LUx = b* и решить в два шага. На первом шаге прямой подстановкой решается система *Ly = b*, а на втором обратной подстановкой решается система *Ux = y*, откуда находится искомый вектор *x.*

Таким образом алгоритм метода можно описать так:

1. Раскладываем матрицу A на L и U по формулам (где i ≤ j), (где i > j), при i = 1, 2, …, n; j = 1, 2, …, n
2. По формуле (где i = 1, 2, …, n) находим вектор y
3. По формуле (где i = 1, 2, …, n) находим вектор решений

4. Анализ задачи

Для применения метода LU-разложения достаточно, чтобы исходная матрица A была невырожденной и все её главные миноры были отличны от нуля. Если выполнены эти условия, то матрицу A всегда можно представить как произведение *A=LU*, а если диагональные элементы одной из матриц L или U фиксированы (ненулевые), то такое разложение единственно.

5. Тестовый пример

Возьмём СЛАУ вида *Ax = b* с данными входными параметрами:

, а ожидаемое решение .

Подставив входные данные, можно убедиться в правильности системы.

Теперь попробуем получить вектор *x* методом LU-разложения.

1. Начнём считать матрицы L и U по формулам. Исходная матрица A имеет размер 3х3, значит в каждой из матриц разложения будет по 9 элементов, по 3 из которых будут равны нулю, найдём их:

i = 1, j = 1: u11 =a11 = 1, l11 = 1

i = 1, j = 2: u12 = a12 = 5, l12 = 0

i = 1, j = 3: u13 = a13 = 3, l13 = 0

i = 2, j = 1: u21 = 0, l­21 = a21 / u11 = 7

i = 2, j = 2: u22 = a22 – l21 \*u12 = 6 – 7 \* 5 = -29, l22 = 1

i = 2, j = 3: u23 = a23 – l21 \* u13 = 11 – 7 \* 3 = -10, l23 = 0

i = 3, j = 1: u31 = 0, l31 = a31 / u11 = 2 / 1 = 2

i = 3, j = 2: u32 = 0, l31 = (a32 – l31 \* u12) / u22 = (9 – 2 \* 5) / -29 = 0.034 ≈ 0.03

i = 3, j = 3: u33 = a33 – (l31 \* u13 + l32 \* u23) = 4 – (2 \* 3 + 0.034 \* (-10)) = -1.66

Таким образом

1. *Ax = b* выражаем как *LUx = b* и решаем прямой подстановкой *Ly = b.* В векторе y будет 3 элемента, найдём их:

i = 1: y1 = b1 = 7

i = 2: y2 = b2 – l21 \* y1=20 – 7 \* 7 = -29

i = 3: y3 = b3 – (l31 \* y1 + l32 \* y2) = 13 – (2 \* 7 + 0.03 \* (-29)) = -0.13

Таким образом

1. Решим *Ux = y* обратной подстановкой, в векторе x будет 3 элемента, найдём их:

i = 3: x3 = y3 / u33 = -0.13 / -1.66 = 0.08

i = 2: x2 = (y2 – u23 \* x3) / u22 = (-29 + 10 \* 0.08) / -29 = 0.97

i = 1: x1 = (y1 – (u12 \* x2 + u13 \* x3)) / u11 = (7 – (5 \* 0.97 + 3 \* 0.08)) / 1 = 1.91

Таким образом,

Решение, полученное с помощью LU-разложения оказалось неточным. Это связано с тем, что в процессе решения производились операции деления и умножения, результаты которых округлялись (в данном случае округление было принято до второго знака после запятой, поэтому погрешность решения такая высокая).

6. Контрольные тесты

Для тестов будем использовать матрицы размером 12х12, элементы которых создаются с помощью псевдорандомной функции rand() с сидом, заданным по умолчанию, из библиотеки stdlib.h на языке си. Значение числа обусловленности таких матриц при этом будет меняться от 0 до 2000.

Число обусловленности считается как произведение нормы исходной матрицы и нормы обратной матрицы cond(A) = ||A|| \* ||A-1||

Вторая норма матрицы считается как корень из суммы квадратов всех её элементов

Обратная матрица рассчитывается по методу Гаусса-Жордано. Исходная матрица A сводится к единичной, в то время как над единичной матрицей B проводятся аналогичные преобразования. Когда матрица A станет единичной, матрица B примет вид обратной к исходной матрице.

Весь анализирующий функционал реализован на языке си в visual studio 2019, а графическая интерпретация в пакете MATLAB.

Для рассмотрения зависимости значения вычислительной ошибки и невязки от числа обусловленности матрицы, рассмотрим 100 различных матриц.

Пример работы программы, где первый столбец – число обусловленности, второй – вычислительная ошибка, третий – невязка:

Для рассмотрения зависимости значения относительной погрешности от внесённой в вектор b ошибки, возьмём матрицу с собственным числом равным 1030 и будем менять значения вектора случайно в пределах 0-5% по формуле: res[i] = vect[i] \* (1 + (double)((double)rand() / (double)(RAND\_MAX / (double)(rand()%5 + 1)))), где res – изменённый вектор, vect – исходный вектор b, RAND\_MAX – константа из библиотеки stdlib.h, а i = 1, 2, …, n.

Пример работы программы, где первый столбец – относительная погрешность, второй – ошибка, внесённая в вектор b:

7. Модульная структура программы

double\* CreateMatrix(int size) – функция создания квадратичой матрицы с заданным размером. Возвращает полученную матрицу

double\* CreateVector(int size) – функция создания вектора с заданным размером. Возвращает полученный вектор.

void CreateLUMatrix(double\* matr, int size, double\*\* L, double\*\* U) – функция разложения матрицы matr с размерностью size на матрицы L и U, ссылки на которые передаются во входных параметрах.

double\* Solution (double\* L, double\* U, int size, double\* b) – функция, решающая систему LUx = b в два шага с вычислением вектора y. Во входных параметрах принимает L и U матрицы, на которые разложена исходная, размерность обеих матриц и вектор свободных членов b. Возвращает вектор решений.

double GetCond(double\* matr, int size) – функция, считающая число обусловленности матрицы по второй норме. Принимает на вход матрицу и её размерность. Возвращает вещественное значение числа обусловленности.

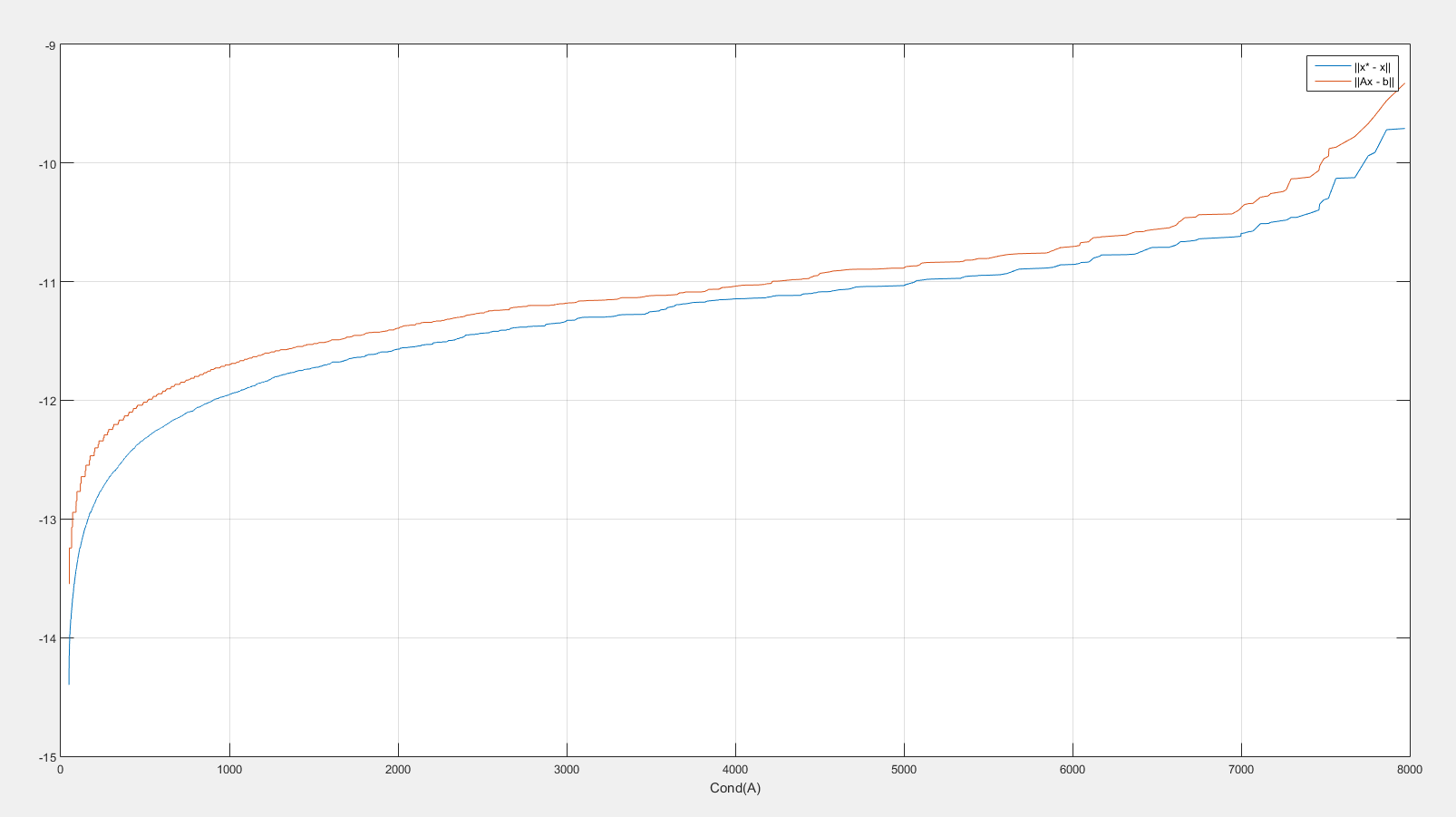
double\* VectSub(double\* v1, double\* v2, int size) – функция вычисления вектора разности двух векторов. Принимает на вход два вектора и их размер. Возвращает вектор разности.

double FindNorm(double\* vect, int size) – функция вычисления нормы вектора. Принимает на вход вектор и его размер. Возвращает вещественное значение нормы.

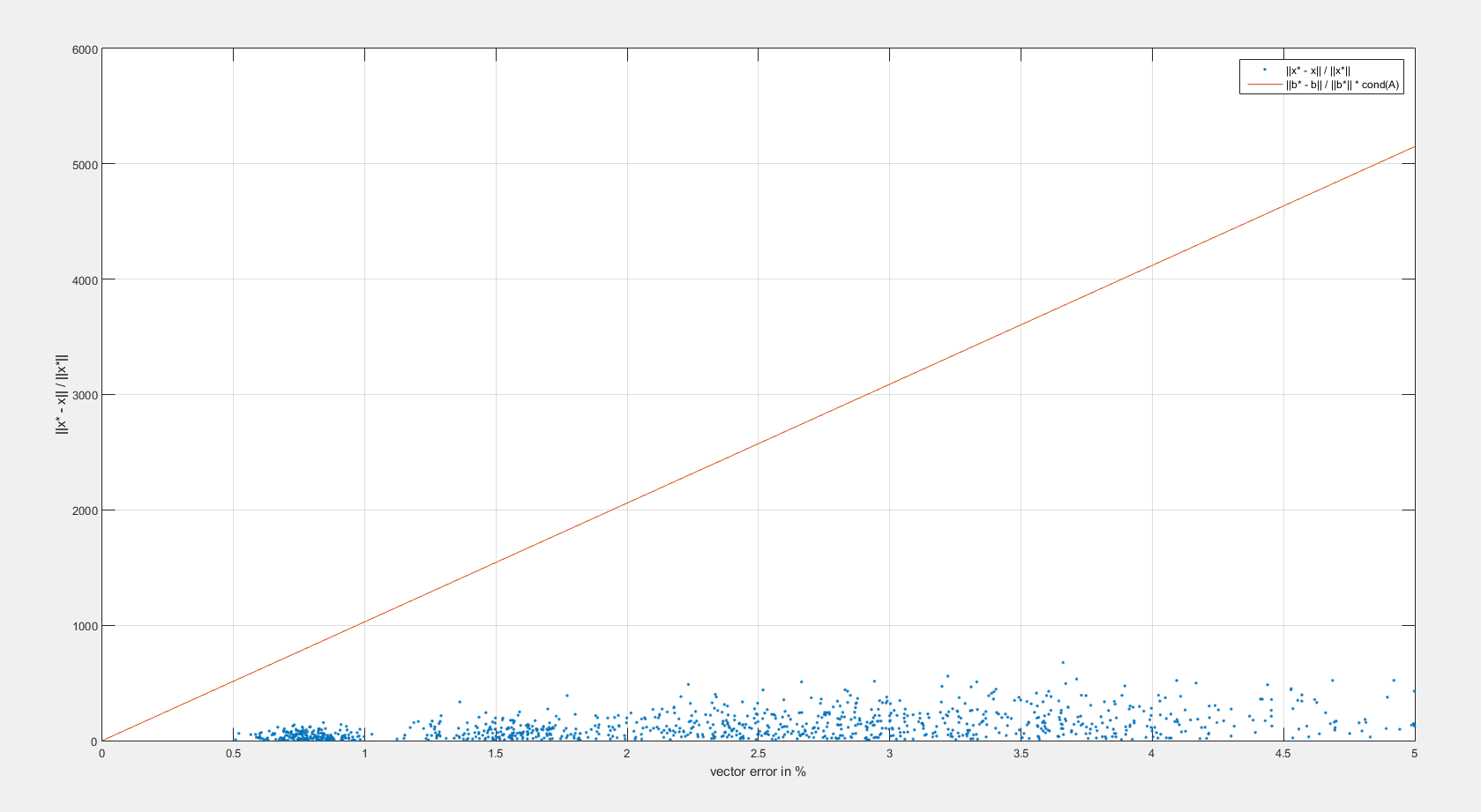
void ChangeVector(double\*\* vect, int size, double num) – функция изменения вектора свободных членов. Принимает на вход ссылку на вектор b, его размер и величину изменения.

8. Численный анализ

Графики зависимостей вычислительной ошибки и невязки от cond(A) и относительной погрешности от внесённой ошибки, построенные в MATLAB:



||x\* - x||(cond(A)) и ||Ax – b||(cond(A))



||x\* - x|| / ||x\*|| от внесённой ошибки

На первых двух графиках по оси y взят логарифмический масштаб, мы можем наблюдать линейное возрастание. ||x\* - x|| и ||Ax – b|| - связанные понятия и соответственно дают очень схожие графики зависимости. Данные зависимости показывают, что чем больше число обусловленности матрицы, тем менее точным становится ответ, полученный с помощью прямого метода LU-разложения.

Третий график не даёт чёткой зависимости. Попробуем оценить её возможные значения:

. Тогда получим и итоговая оценка относительной погрешности:

Полученная зависимость полностью удовлетворяет данной теоретической оценке.

9. Вывод

Метод решения СЛАУ с помощью LU-разложения оказался пригоден для применения и дал достаточно точный ответ.

Точность полученного решения зависит от числа обусловленности матрицы и для получения наиболее точного ответа стоит выбирать матрицы с наименьшим cond. Тенденция графика показывает, что для матриц с очень большим числом обусловленности (порядка десятков тысяч) абсолютная погрешность ответа может оказаться уже гораздо более заметной, а для матриц с ещё большим cond она может оказаться критической.

При внесении ошибки в вектор свободных членов в рамках 5% относительная погрешность укладывается в рамки и в целом сопоставима с относительной величиной ошибки.