**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

В данной работе будут рассматриваться уравнения в матричном виде . Необходимо решить данное уравнение с использованием метода LU-разложения и метода простых итераций, считая количество использованных операций +/- и \*/ в каждом из методов.

По итогу исследования изучить зависимость точности решения итерационным методом, если число операций в нём равно числу операций в прямом, в два раза больше и в два раза меньше. Также исследовать зависимость числа операций в обоих методах от точности вычисления решения ε.

По зависимостям построить графики в среде MATLAB.

2. Постановка задачи

В работе решается задача о нахождении системы уравнений вида

Или соответствующего матричного уравнения , 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 поочерёдно находится сначала прямым методом LU-разложения за конечное число операций, а потом итерационным методом простых итераций с фиксированным числом операций за одну итерацию.

3. Анализ задачи

Для применения метода LU-разложения достаточно, чтобы исходная матрица A была невырожденной и все её главные миноры были отличны от нуля. А для применения метода простых итераций необходимо и достаточно чтобы все собственные числа матрицы *B* были по модулю меньше 1. Так как условия сходимости метода простых итераций включают в себя условия метода LU, то для обоих методов будем использовать условия метода прямых итераций.

Для того, чтобы вести подсчёт числа математических операций, создадим четыре переменные, которые будут хранить число операций сложения/вычитания и умножений/делений в прямом и итерационном методах. С этой же целью модифицируем все функции, участвующие в вычислении решения обоими методами, чтобы они считали число проделанных операций.

4. Тестовый пример

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

, а ожидаемое решение .

Подставив входные данные, можно убедиться в правильности системы.

Посчитаем 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 = 7, l11 = 1

i = 1, j = 2: u12 = a12 = 0.4, l12 = 0

i = 1, j = 3: u13 = a13 = 0.4, l13 = 0

i = 2, j = 1: u21 = 0, l21 = a21 / u11 = 0.03

i = 2, j = 2: u22 = a22 – l21 \*u12 = 8 – 0.03 \* 0.4 = 7.99, l22 = 1

i = 2, j = 3: u23 = a23 – l21 \* u13 = 0.4 – 0.03 \* 0.4 = 0.39, l23 = 0

i = 3, j = 1: u31 = 0, l31 = a31 / u11 = 0.1 / 7 = 0.01

i = 3, j = 2: u32 = 0, l32 = (a32 – l31 \* u12) / u22 = (0.1 – 0.01 \* 0.4) / 7.99 = 0.01

i = 3, j = 3: u33 = a33 – (l31 \* u13 + l32 \* u23) = 5 – (0.01 \* 0.4 + 0.01 \* 0.39) = 4.99, l33 = 1

Таким образом

Общее число операций сложения/вычитания: 5

Общее число операций умножения/деления: 7

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 = 9.8

i = 2: y2 = b2 – l21 \* y1 = 41 – 0.03 \* 9.8 = 40.7

i = 3: y3 = b3 – (l31 \* y1 + l32 \* y2) = 10.6 – (0.01 \* 9.8 + 0.01 \* 40.7) = 10.1

Таким образом

Общее число операций сложения/вычитания: 8

Общее число операций умножения/деления: 10

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

i = 3: x3 = y3 / u33 = 10.1 / 4.99 = 2.02

i = 2: x2 = (y2 – u23 \* x3) / u22 = (40.7 – 0.39 \* 2.02) / 7.99 = 5

i = 1: x1 = (y1 – (u12 \* x2 + u13 \* x3)) / u11 = (9.8 – (0.4 \* 5 + 0.4 \* 2.02)) / 7 = 1

Таким образом,

Общее число операций сложения/вычитания: 11

Общее число операций умножения/деления: 16

Теперь посчитаем одну итерацию МПИ:

1. По формуле ci = bi / Aii

c1 = b1 / A11 = 9.8 / 7 = 1.4

c2 = b2 / A22 = 41 / 8 = 5.125

c3 = b3 / A33 = 10.6 / 5 = 2.12

Общее число операций сложения/вычитания: 0

Общее число операций умножения/деления: 3

1. По формуле Bij = -Aij / Aii, при i ≠ j и Bij = 0 при i = j рассчитаем значения B:

B11 = 0

B12 = A12 / A11 = 0.4 / 7 = 0.05714

B13 =A13 / A11 = 0.4 / 7 = 0.05714

B21 = A21 / A22 = 0.2 / 8 = 0.025

B22 = 0

B23 = A23 / A22 = 0.4 / 8 = 0.05

B31 = A31 / A33 = 0.1 / 5 = 0.02

B32 = A32 / A33 = 0.1 / 5 = 0.02

B33 = 0

Общее число операций сложения/вычитания: 0

Общее число операций умножения/деления: 9

1. Возьмём вектор x0 = c
2. Сделаем итерацию по итеративной формуле *x(k+1)= Bx(k) + c*

x(1) = Bx(0) + c =

Общее число операций сложения/вычитания: 9

Общее число операций умножения/деления: 18

Таким образом, для матрицы 3 *х* 3 метод LU-разложения затрачивает на 2 операции сложения/вычитания больше и на 2 операции умножения/деления больше.

5. Контрольные тесты

Для тестов будем использовать матрицу размером 12х12, элементы которой создаются с учётом диагонального преобладания с помощью псевдорандомной функции rand() с сидом, заданным по умолчанию, из библиотеки stdlib.h на языке си. В каждой строке матрицы сначала вычисляются все элементы, кроме диагонального, как случайные числа от 1 до 9 включительно, а потом диагональный элемент считается как

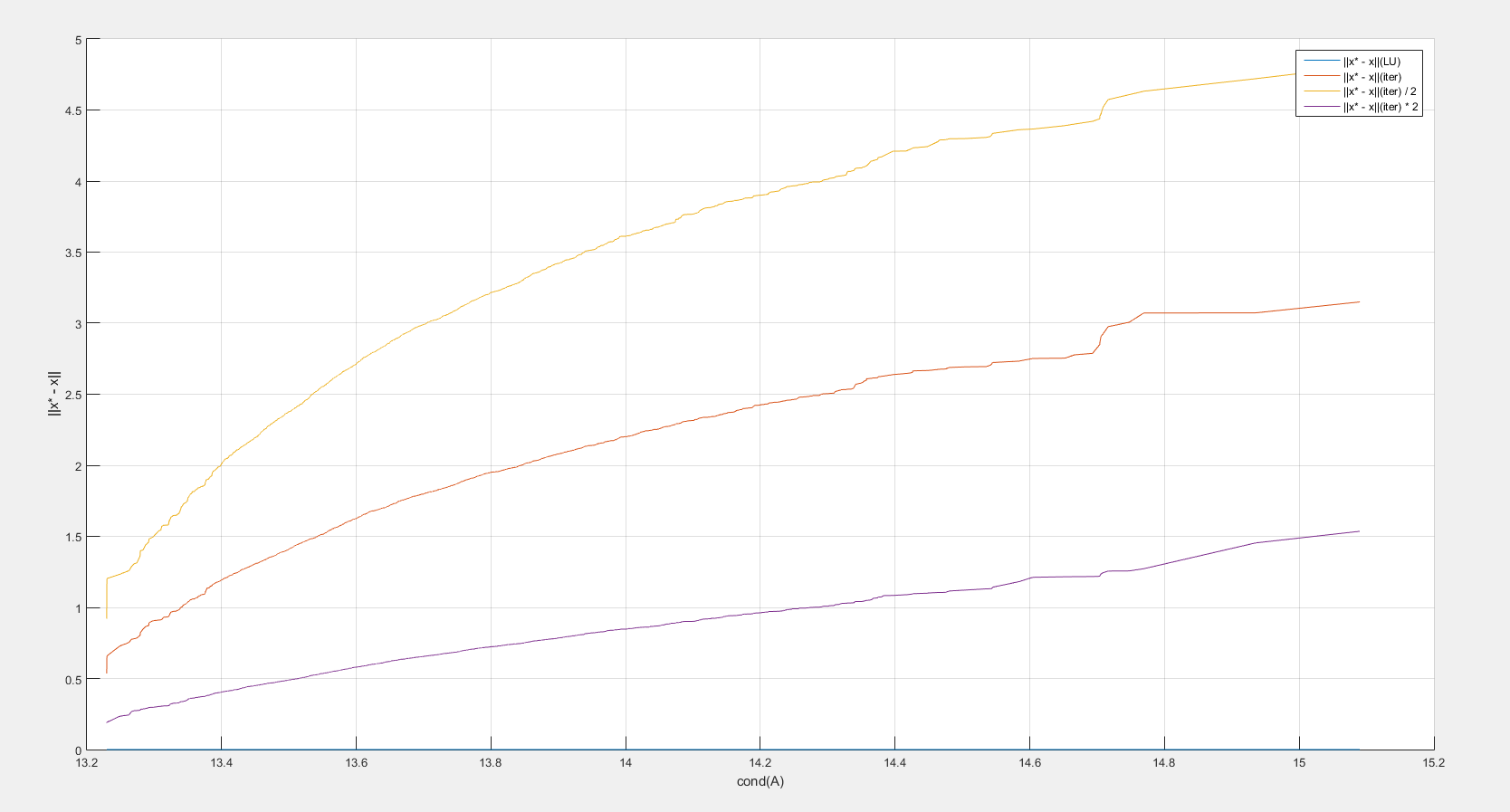
Весь анализирующий функционал реализован на языке си в visual studio 2019, а графическая интерпретация в пакете MATLAB.

В силу диагонального преобладания в матрице, при рассмотрении зависимости от числа обусловленности cond(A) оно будет изменяться в пределах от 13.2 до 15.2 при фиксированном ε = 10-14. Число опытов – 3000.

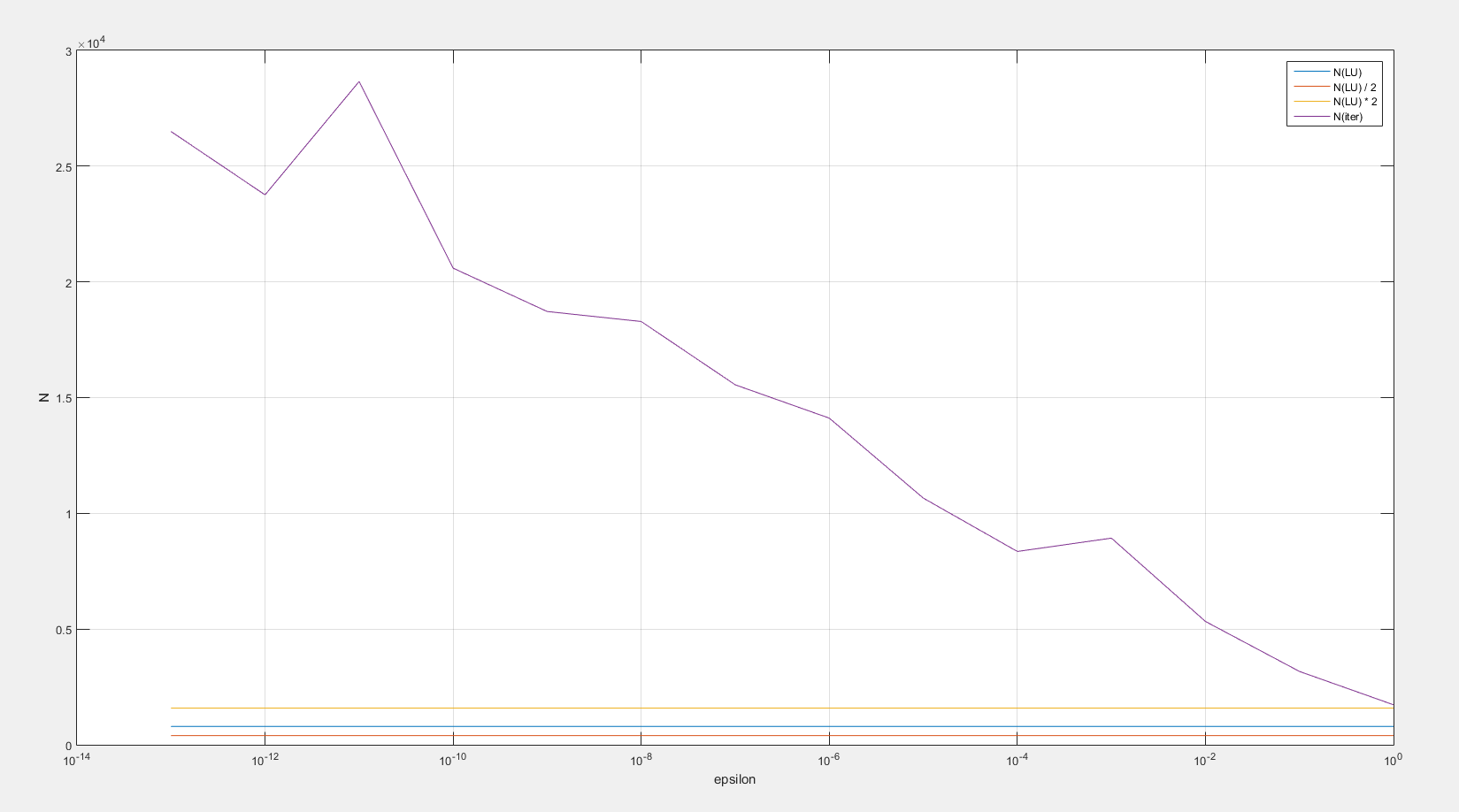
При рассмотрении зависимости от приближения решения ε оно будет изменяться от 10-13 до 10-1 с домножением на 10 на каждом шаге.

6. Численный анализ

Графики зависимостей, построенные в MATLAB:



На этом графике показана зависимость вычислительной ошибки ||x\* - x|| от числа обусловленности cond(A) исходной матрицы для метода LU-разложения (синим цветом), метода прямых итераций, выполняемого до тех пор, пока количество операций умножения/деления не превзойдёт количество этих операций в прямом методе (оранжевым цветом), не превзойдёт делёное пополам количество операций прямого метода (жёлтым цветом) и не превзойдёт удвоенное количество операций прямого метода (фиолетовым цветом). Из графика видно, что большее число проведённых операций в МПИ приводит к большему числу итераций данного метода, а, следовательно, и к большей точности вычислений. Однако, для достижения точности метода LU-разложения нужно значительное большее число математических операций.



На этом графике показана зависимость числа проделанных математических операций умножения/деления N обоими методами от необходимой точности вычислений ε. Фиолетовым цветом показано изменяющееся значение метода простых итераций, которое падает с уменьшением требуемой точности. Синим показано число операций метода LU-разложения, которое не меняется независимо от требуемой точности. Жёлтым и оранжевым показаны удвоенное и делёное пополам число операций соответственно. График показывает, что только при значении ε = 10-1 число операций итерационного метода становится чуть меньшим, чем удвоенное число операций прямого. Во всех остальных случаях метод простых итераций требует в разы большее количество операций.

7. Вывод

На используемых входных данных метод LU-разложения показывает гораздо лучшие результаты, чем метод простых итераций. Он требует в разы меньшего числа математических операций умножения/деления при любых значениях приближения ε, и одновременно даёт во много раз более точный результат.

Плюсом метода простых итераций можно отметить, что его можно использовать для нахождения решения с разной точностью, в том числе и гораздо большей, чем может предоставить метод LU-разложения. Однако, он является гораздо более затратным по числу операций и для небольших значений точности ε рекомендуется использовать прямой метод.