Санкт – Петербургский политехнический университет Петра Великого

Физико-механический институт

Отчет по лабораторной работе №1 по Численным методам «Разреженные матрицы и предобусловленный метод сопряженных градиентов» ILU(p), «Бабочка»

Выполнил студент

Группы 5030301/10002 Тугай В.В.

Преподаватель: К.Н. Козлов

Санкт-Петербург

2022

Исследования, проводимые в ходе работы:

1. Ручной расчет
2. Предобусловленный метод сопряженных градиентов
3. ILU(p) предобуславливание
4. Структура программы
5. Зависимость точности от числа обусловленности
6. Зависимость времени выполнения от числа обусловленности
7. Зависимость относительной погрешности от возмущения
8. Зависимость погрешности от заданной точности
9. Требуемое число итераций от заданной точности
10. Уменьшение погрешности с ходом итерации
11. Заполнения множителей разложения Холецкого
12. Временные затраты на разложение Холецкого

Вариант 21

ILU(p), «Бабочка»

Ручной расчет

$$x_0=\begin{pmatrix}0\\0\\0\\0\\0\\0\\0\end{pmatrix},\quad A=\begin{pmatrix}4&-1&-1&0&0&0&0\\-1&4&0&-1&0&0&0\\-1&0&4&-1&0&0&0\\0&-1&-1&4&-1&-1&0\\0&0&0&-1&4&0&-1\\0&0&0&-1&0&4&-1\\0&0&0&0&-1&-1&4\end{pmatrix},\quad x_{\text{точн}}=\begin{pmatrix}1\\1\\1\\1\\1\\1\\1\end{pmatrix},\quad B=\begin{pmatrix}2\\2\\2\\0\\2\\2\\2\end{pmatrix}$$

$$M=\begin{pmatrix}0{,}029&0{,}079&0{,}079&0{,}045&0{,}012&0{,}012&0{,}006\\0{,}079&0{,}29&0{,}024&0{,}091&0{,}024&0{,}024&0{,}012\\0{,}079&0{,}024&0{,}29&0{,}0907&0{,}024&0{,}024&0{,}012\\0{,}045&0{,}091&0{,}091&0{,}34&0{,}089&0{,}089&0{,}045\\0{,}012&0{,}024&0{,}024&0{,}089&0{,}2904&0{,}021&0{,}078\\0{,}012&0{,}024&0{,}024&0{,}089&0{,}021&0{,}2904&0{,}078\\0{,}0059&0{,}012&0{,}012&0{,}045&0{,}078&0{,}078&0{,}29\end{pmatrix}$$

$$r_0=b-Ax_0=\begin{pmatrix}2\\2\\2\\0\\2\\2\\2\end{pmatrix};\quad z_0=M\cdot r_0=\begin{pmatrix}0{,}954\\0{,}907\\0{,}907\\0{,}903\\0{,}898\\0{,}898\\0{,}949\end{pmatrix};\quad p_0=z_0;\quad \varepsilon=0{,}001;\quad \|r_0\|=4{,}899$$

$$q_0=A\cdot p_0=\begin{pmatrix}2\\1{,}77\\1{,}77\\0\\1{,}74\\1{,}74\\2\end{pmatrix};\quad \alpha_0=\frac{r_0^T z_0}{p_0^T q}=1{,}086;\quad x_1=x_0+\alpha_0 p_0=\begin{pmatrix}1{,}0361\\0{,}986\\0{,}986\\0{,}981\\0{,}976\\0{,}976\\1{,}031\end{pmatrix};$$

$$r_1=r_0-\alpha_0 q=\begin{pmatrix}-0{,}173\\0{,}074\\0{,}074\\0\\0{,}109\\0{,}109\\-0{,}173\end{pmatrix};\quad \frac{\|r_1\|}{\|r_0\|}=0{,}0626>\varepsilon;\quad z_1=M\cdot r_1=\begin{pmatrix}-0{,}037\\0{,}013\\0{,}013\\0{,}0173\\0{,}022\\0{,}022\\-0{,}032\end{pmatrix};\quad \beta_0=\frac{r_1^T z_1}{r_0^T\cdot z_0}=0{,}0[illegible]$$

$$p_1=z_1+\beta_0 p_0=\begin{pmatrix}-0{,}035\\0{,}014\\0{,}014\\0{,}0188\\0{,}0234\\0{,}0234\\-0{,}0307\end{pmatrix};\quad q_1=Ap_1=\begin{pmatrix}-0{,}1675\\0{,}0739\\0{,}0739\\0\\0{,}1053\\0{,}1053\\-0{,}169\end{pmatrix};\quad \alpha_1=\frac{r_1^T z_1}{p_1 q}=1{,}021;\quad x_2=x_1+\alpha_1 p_1=\begin{pmatrix}1{,}0001\\1{,}0004\\1{,}0004\\1\\0{,}9997\\0{,}9997\\0{,}9998\end{pmatrix}$$

$$r_2=r_1-\alpha_1 q=\begin{pmatrix}0{,}0002\\-0{,}0014\\-0{,}0014\\0{,}0011\\0{,}0011\\0{,}0002\end{pmatrix};\quad \frac{\|r_2\|}{\|r_0\|}=0{,}000516<0{,}001 \Rightarrow \text{конец расчета}$$

$$x=\begin{pmatrix}1{,}0001\\1{,}0004\\1{,}0004\\1\\0{,}9997\\0{,}9997\\0{,}9998\end{pmatrix}.$$

Предобусловленный метод сопряженных градиентов

Алгоритм метода сопряженных градиентов:

1. Выбираем начальное приближение x_0, вычисляем невязку и вектор направления $r_0 = b - Ax_0, p_0 = r_0$
2. Для k = 0,1,2… и до тех пор, пока итерационный процесс не сойдется:

$$1.\ q_k = Ap_k,\ a_k = \frac{{r_k}^T r_k}{{p_k}^T q_k}$$
$$2.\ x_{k+1} = x_k + a_k p,\ r_{k+1} = r_k - a_k q_k$$
$$3.\ \beta_k = \frac{{r_{k+1}}^T r_{k+1}}{{r_k}^T r_k},\ p_{k+1} = r_{k+1} + \beta_k p_k$$

Идея предобсуловливания для ускорения итерационных методов:

$Ax = b, \mathrm{cond}(A)$ велико $\Leftrightarrow$ $M^{-1}Ax = M^{-1}b, \mathrm{cond}(M^{-1}A)$ небольшое, где M — предобусловливатель

Алгоритм предобусловенного метода сопряженных градиентов:

x_0 – начальное приближение, $r_0 = b - Ax_0$; решаем $Mz_0 = r_0$; $p_0 = z_0$;
Пока $\|r_k\|/\|r_0\| > \varepsilon$:

$q = Ap_k$; $\alpha_k = r^T{}_k z_k/(p^T{}_k q)$

$x_{k+1} = x_k + \alpha_k p_k$; $r_{k+1} = r_k - \alpha_k q$
$Mz_{k+1} = r_{k+1}$
$\beta_k = r^T{}_{k+1} z_{k+1}/(r^T{}_k z_k)$; $p_{k+1} = z_{k+1} + \beta_k p_k$

ILU(0) предобуславливание

Пусть A – разреженная матрица NZ(A)={(i,j): $a_{ij} \neq 0$}

Пусть найдено разложение A в форме A=LU – R, где L и U – нижняя (с единичной диагональю) и верхняя треугольные матрицы;

NZ(L) $\cup$ NZ(U) = NZ(A); $r_{ij} = 0$ для всех (i,j) $\in$ NZ(A).
Тогда ILU(0)-предобуславливатель M=LU≈A. Более точное ILU-разложение можно получить, заполнив p дополнительных диагоналей. Начальное значение уровня заполнения $l_{ij} = \begin{cases} 0, \text{если } a_{ij} \neq 0 \text{ или } i = j \\ \infty, \text{иначе} \end{cases}$
На i-м шаге гауссова исключения: $l_{ij} = \min\{l_{ij}, l_{ik} + l_{kj} + 1\}$

Стратегия ILU(p) – обнулить все элементы с уровнем заполнения, большим p.

for i = 2,…,n do

for k = 1, . . . , i − 1 and if $a_{ij} \neq 0$ do

$a_{ik} = a_{ik} / a_{jj}$

$a_{i*} = a_{i*} - a_{ik} * a_{i*}$

обновить уровни заполнения для a_{i*} : $l_{ij} = \min\{l_{ij}, l_{ik} + l_{kj} + 1\}$

для i-й строки: if $l_{ij} > p$ then $a_{ij} = 0$

end k

end i

Структура программы

Программа matrix.m нужна для создания файлов, в которых будут записаны матрицы A 7x7…2000x2000, и в дальнейшем именно эти файлы будет считывать программа lab1.f95. Так же данная программа создает файлы matrix_size.txt, в которых записаны размеры матриц. Программа lab1.f95 считывает эти файлы и задает переменную и массивы: переменная содержит размеры исходных матриц, а массивы – и есть матрицы A.

Программа lab1.f95 состоит из основного тела и двух модулей:

matrix_fluct.mod - модуль, содержащий функции для внесения случайных возмущений в элементы вектора правой части в пределах процента, заданного пользователем.

gradient_method_for_preconditioned_system.mod - модуль, содержащий подпрограмму с реализацией метода, подпрограмму с неполной LU-факторизацией, а также функцию для перевода натуральных чисел в строку для более удобной записи данных в файлы, которые будут созданы по окончанию программы.

Программа lab1.m исследует эффективность обратного алгоритма Катхилла-Макки и минимальной степени в зависимости от степени заполнения разреженной матрицы. После выполнения в папке будет создано порядка 32 рисунков, все они представлены далее в отчете.

Зависимость точности от числа обусловленности

На рисунке 1 изображен график зависимости точности в логарифмических осях от числа размера исходной матрицы A. Как можно увидеть по рисунку, с увеличением размера матрицы растет и ошибка.

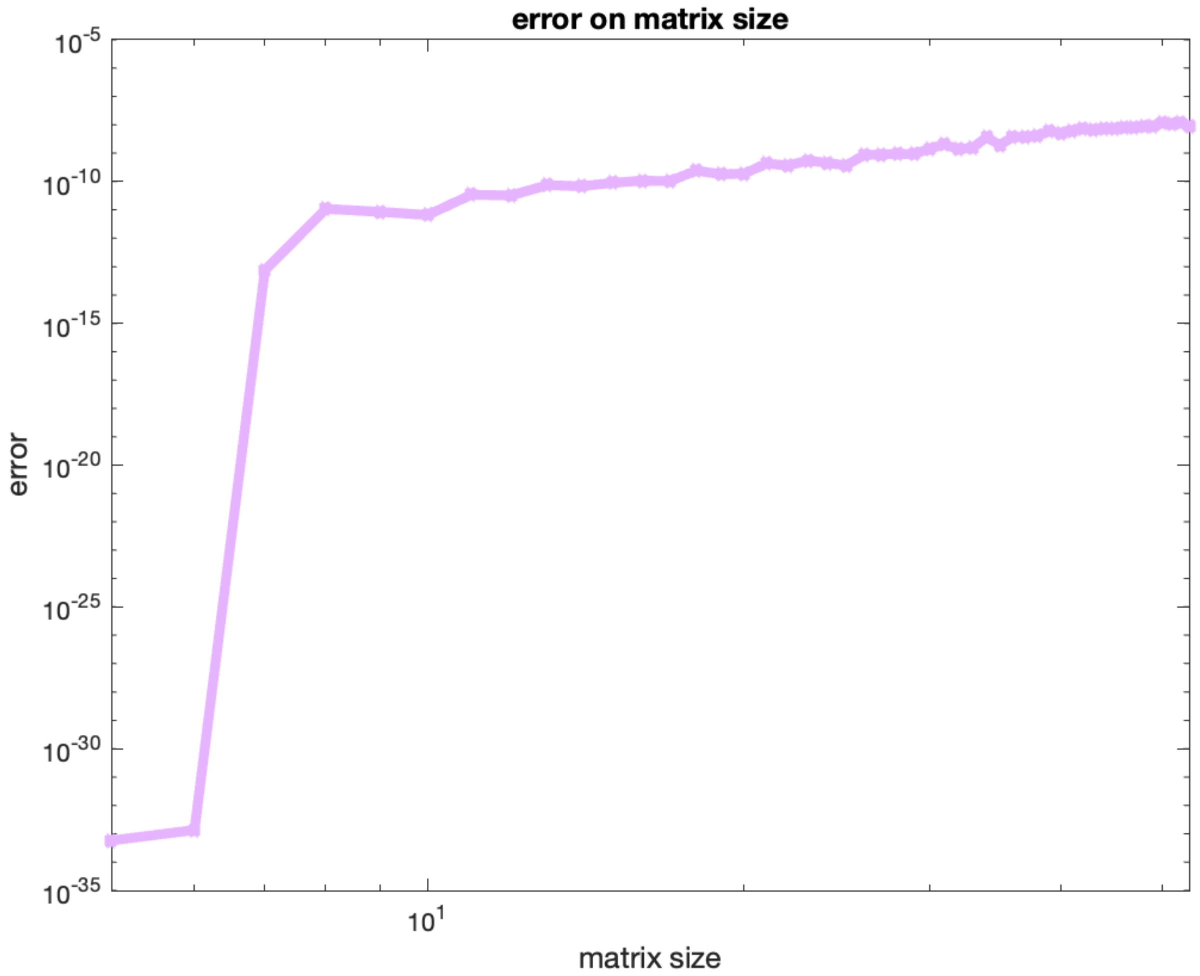

Рисунок 1
Зависимость точности от
числа обусловленности

Зависимость времени выполнения от числа обусловленности

На рисунке 2 изображен график зависимости времени выполнения программы в логарифмических осях от размера исходной матрицы А. Как можно увидеть по рисунку, время уменьшается с увеличением размера матрицы.

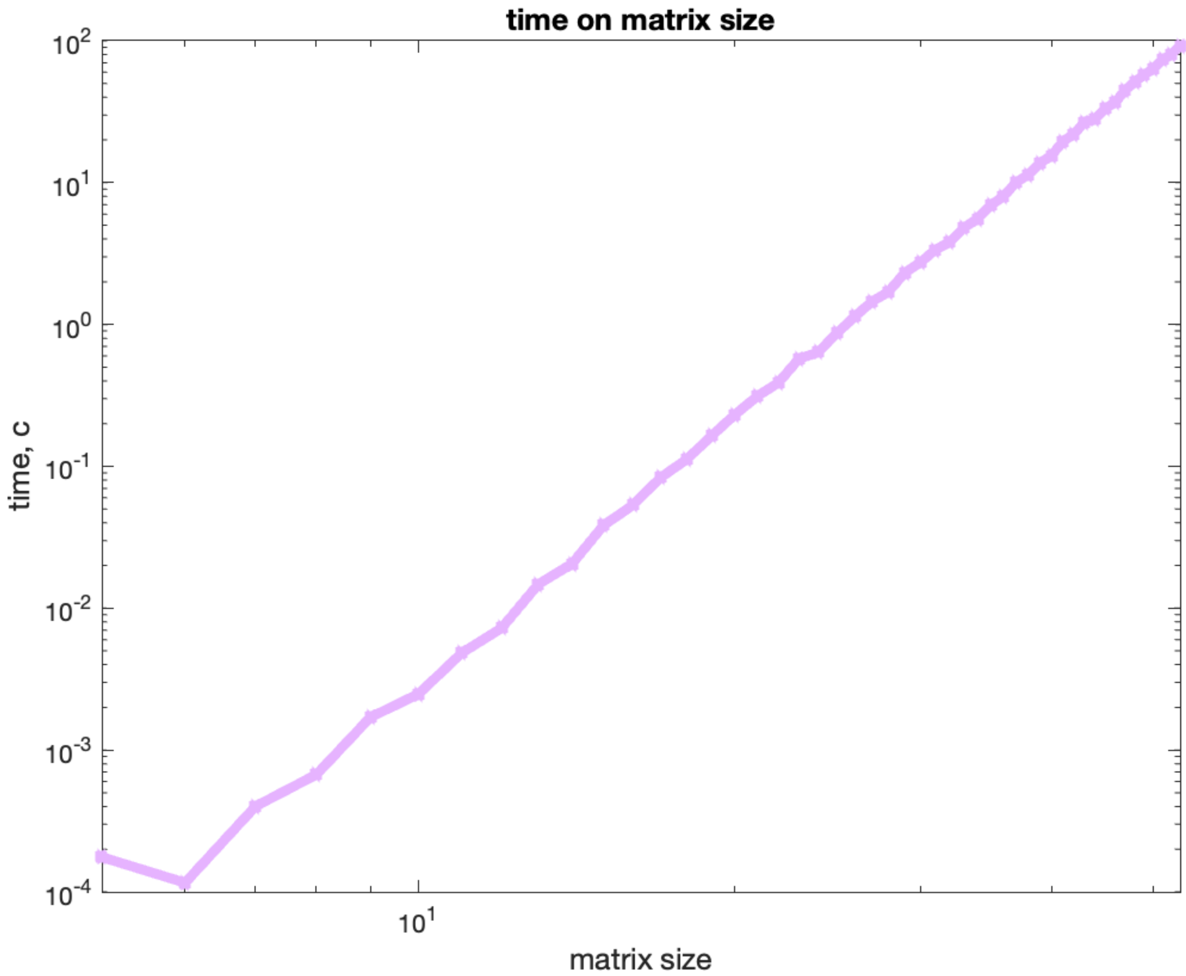

Рисунок 2
Зависимость времени от
числа обусловленности

Зависимость относительной погрешности от возмущения

На рисунке 3 изображены графики зависимости относительной погрешности от возмущения элементов вектора правой части. Погрешность для матрицы размеров 855x855 имеет значения от 100% до 300%, что является высокими значениями, но это связано с тем, что возмущения вносятся в множество элементов.

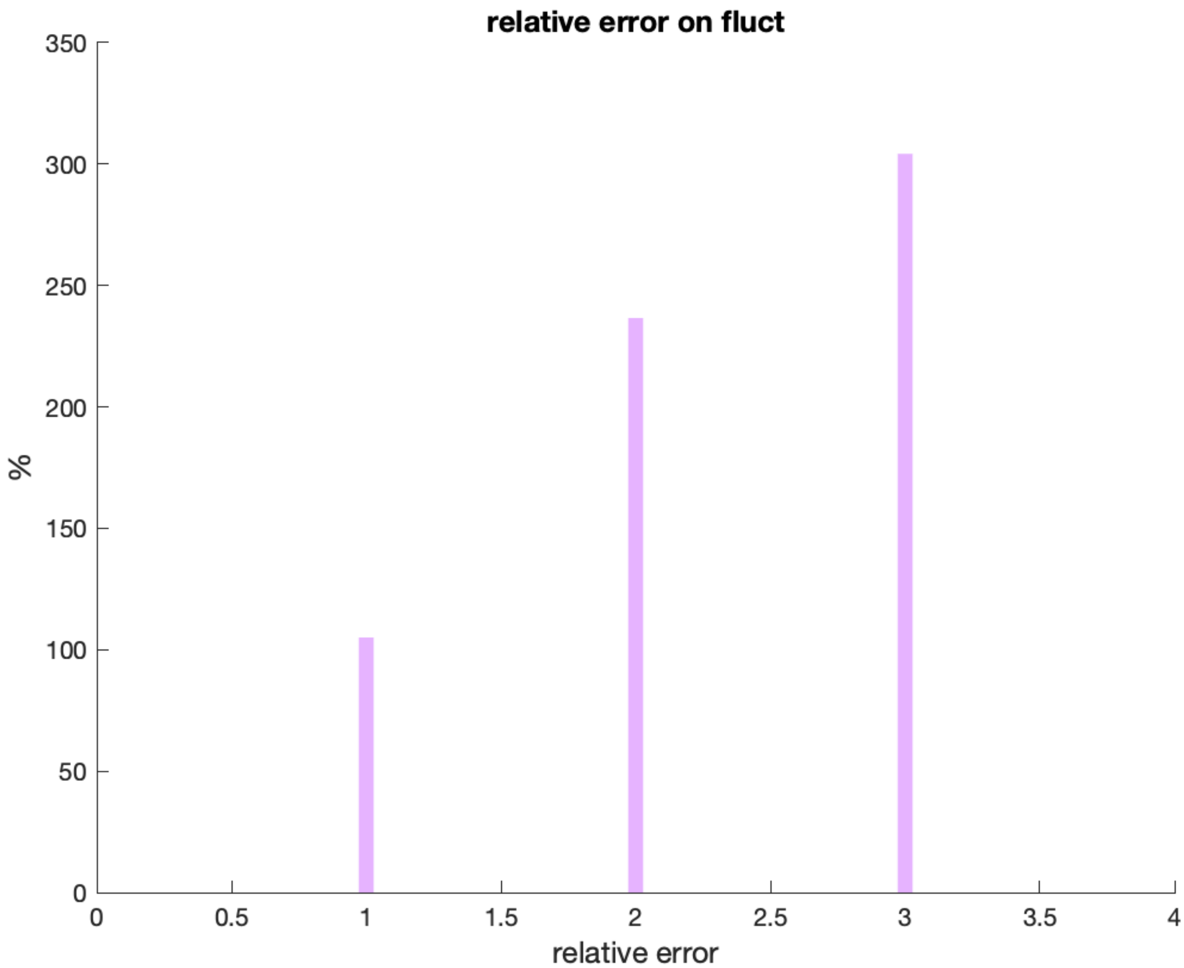

Рисунок 3
Зависимость относительной погрешности
от возмущения для матрицы B

Зависимость погрешности от заданной точности

На рисунке 4 изображен график зависимости погрешности в логарифмических осях от заданной точности. Как можно увидеть по рисунку, с уменьшением значения заданной точности уменьшается и ошибка на выходе, что соответствует смыслу изменения данного параметра.

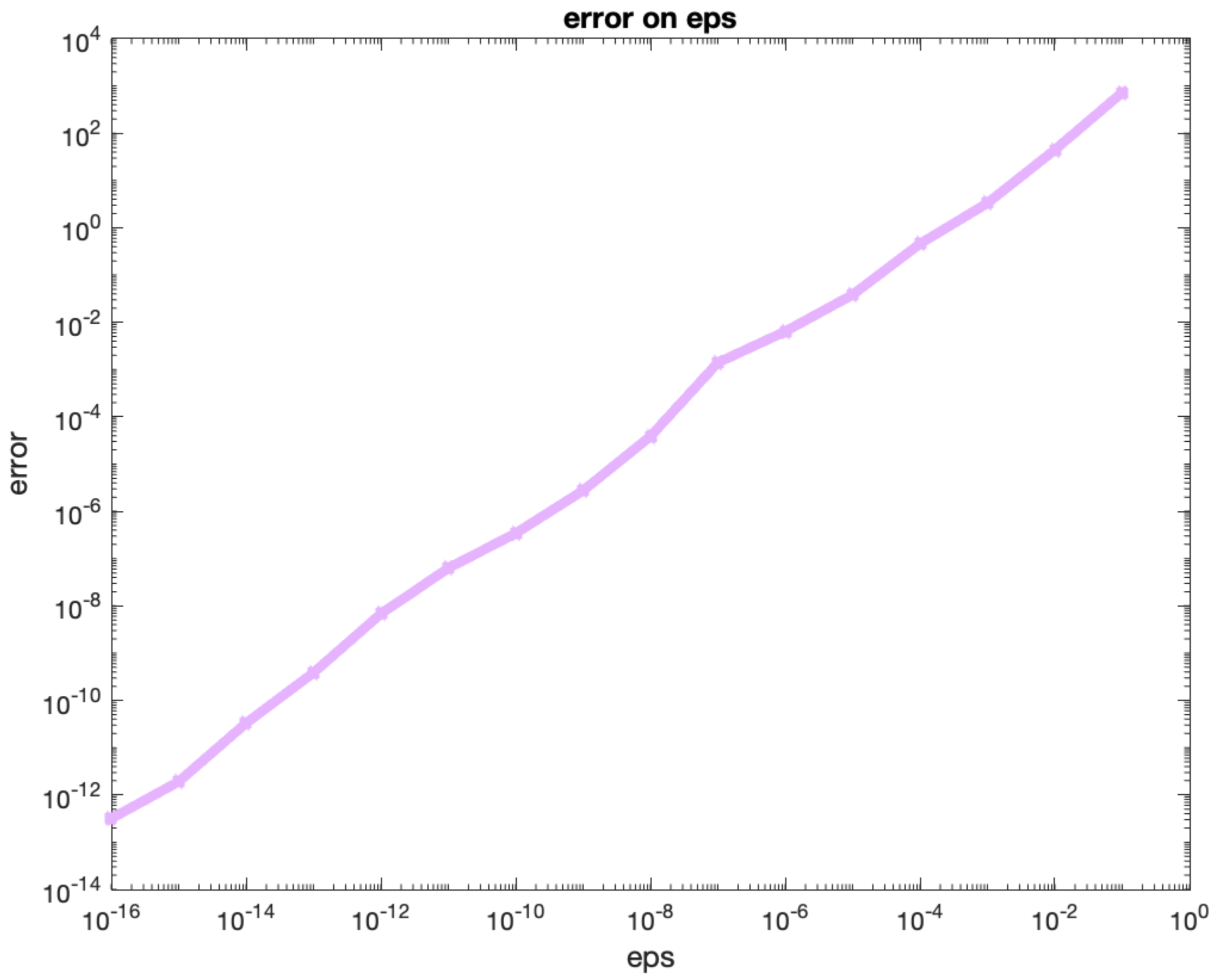

Рисунок 4
Зависимость погрешности от
заданной точности

Требуемое число итераций от заданной точности

На рисунке 5 изображен график зависимости требуемого числа итераций от заданной точности. Как можно увидеть по рисунку, с уменьшением значения заданной точности увеличивается количество итераций, что является логичным выводом для итерационных методов.

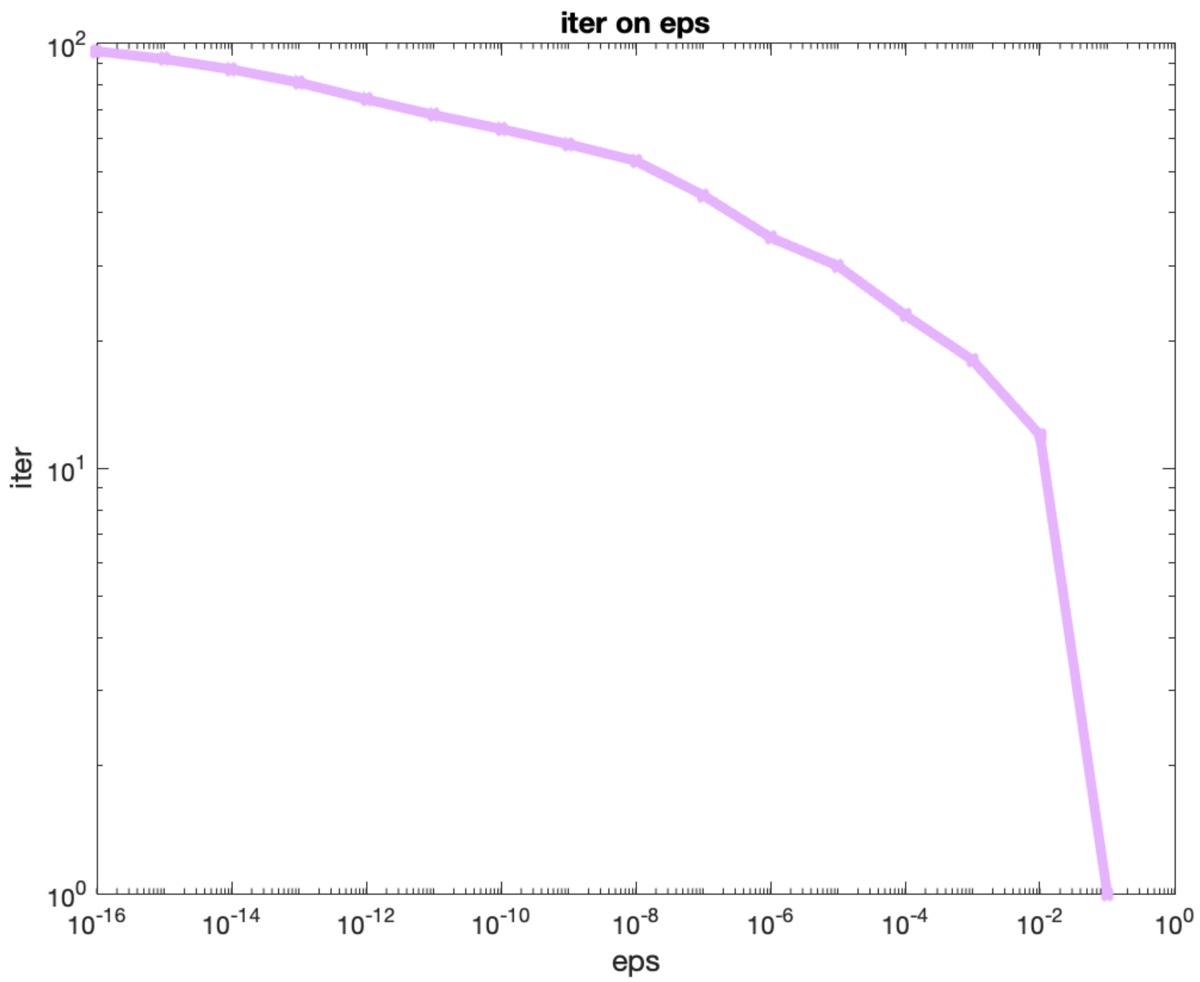

Рисунок 5
Зависимость требуемого числа
итераций от заданной точности

Уменьшение погрешности с ходом итерации

На рисунке 6 изображен график зависимости погрешности от хода итераций. Как можно увидеть по рисунку, с каждым шагом итерации уменьшается и погрешность вычислений, что свидетельствует о правильном ходе программы.

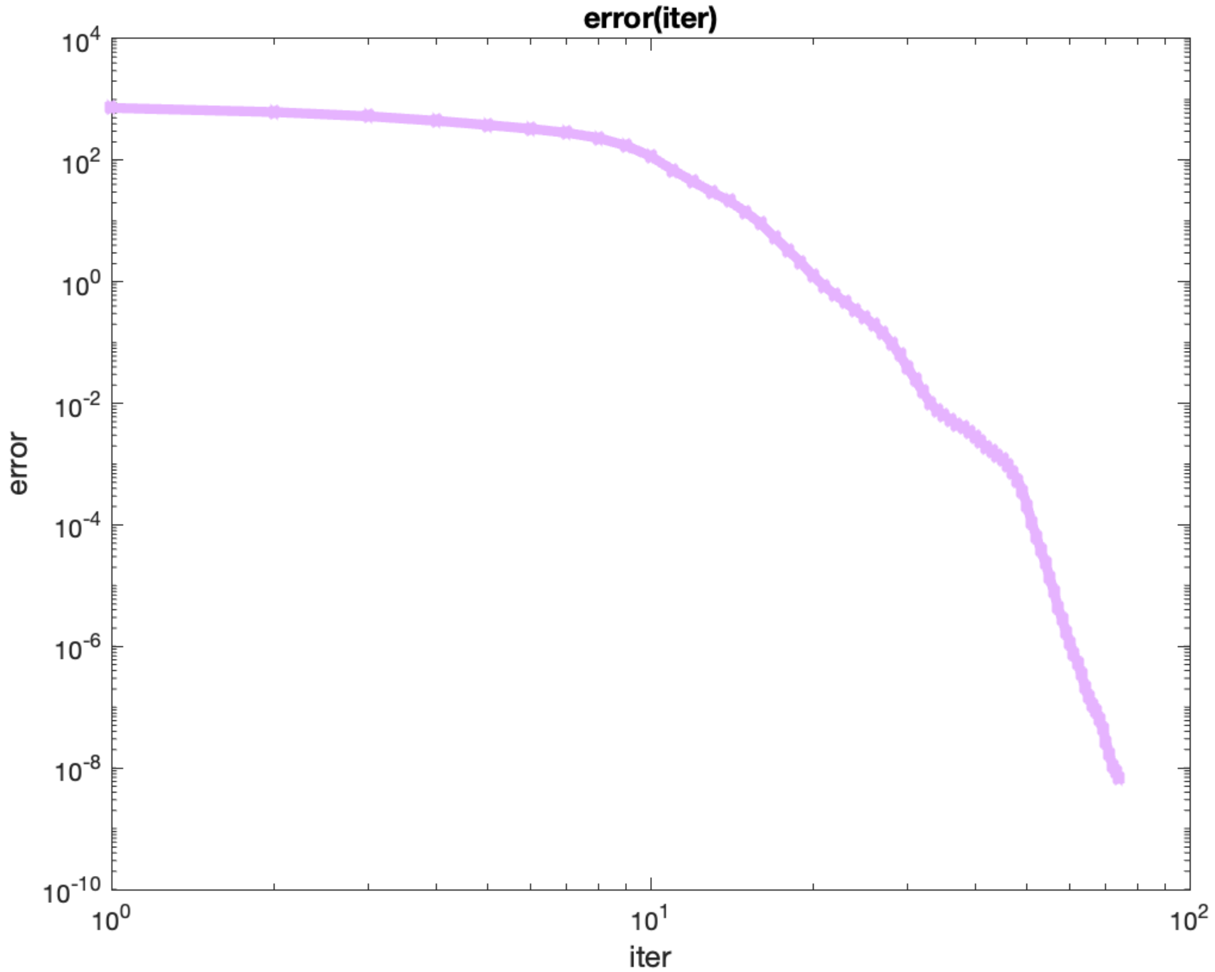

Рисунок 6
Зависимость погрешности
от хода итерации

Изменение относительной погрешности в зависимости от p

На рисунке 7 изображен график зависимости погрешности от значения параметра p для ILU(p). Как можно увидеть по рисунку, с увеличением значения данного параметра увеличивается и ошибка, что связано с увеличением количества нулевых элементов в матрицах L и U.

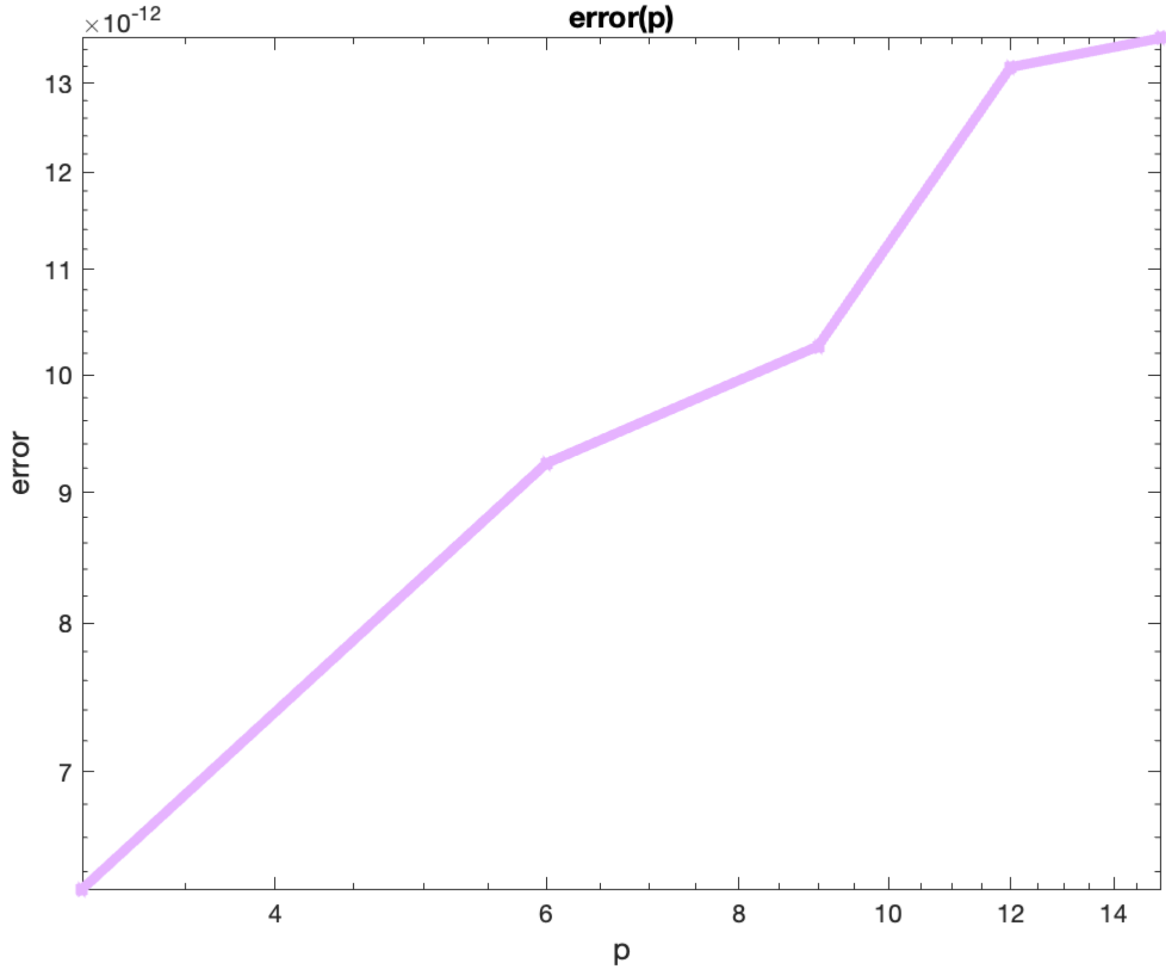

Рисунок 7
Зависимость погрешности
от величины p

Заполнения множителей

На рисунках 8–34 изображены графики заполнения множителей разложения Холецкого при применении функции chol к исходной матрице и к матрицам после применения symrcm и столбцами при помощи symrcm и symamd для степеней разложения от 1 до 9. Анализируя рисунки можно прийти к выводу, что заполнения множителей с помощью функции symrcm (обратный алгоритм Катхилла-Макки) при малых степенях заполнения более плотное, нежели заполнение с помощью функции symamd (минимальная степень), однако при степени заполнения, равной 9, плотность заполнения множителей выше у матрицы после применения функции symamd. Так же очевиден вывод, что с увеличением степени заполнения растет и плотность заполнения множителей, что соответствует смыслу определения степени заполнения.

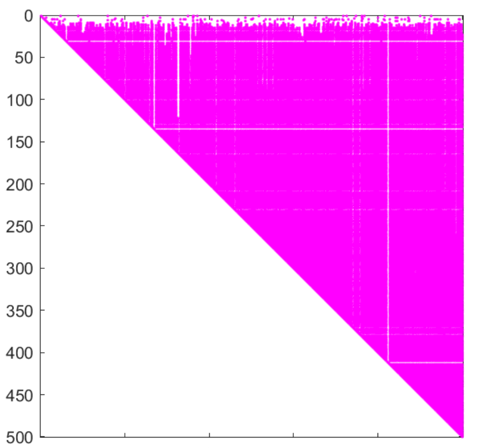

Рисунок 8
Заполнение множителей разложения Холецкого к исходной матрице при степени заполнения 1

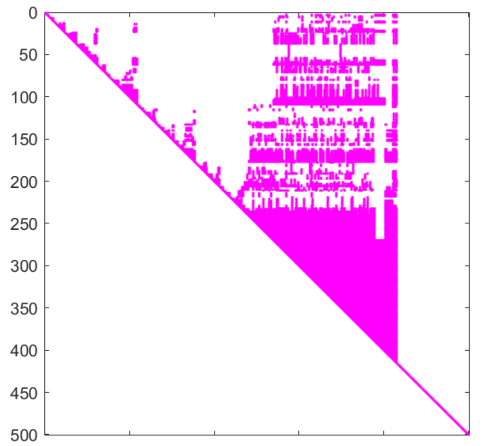

Рисунок 9
Заполнение множителей разложения Холецкого к матрице после применения symamd при степени заполнения 1

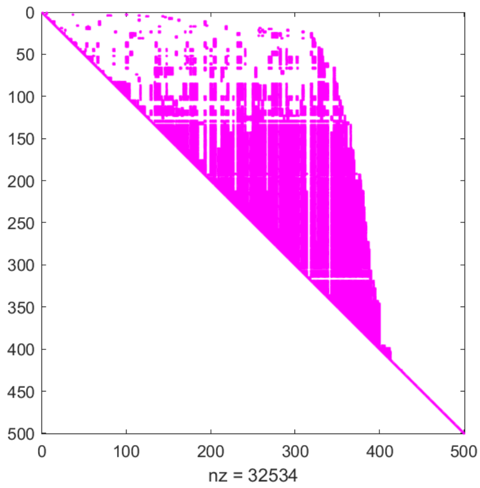

Рисунок 10
Заполнение множителей разложения Холецкого к матрице после применения symrcm при степени заполнения 1

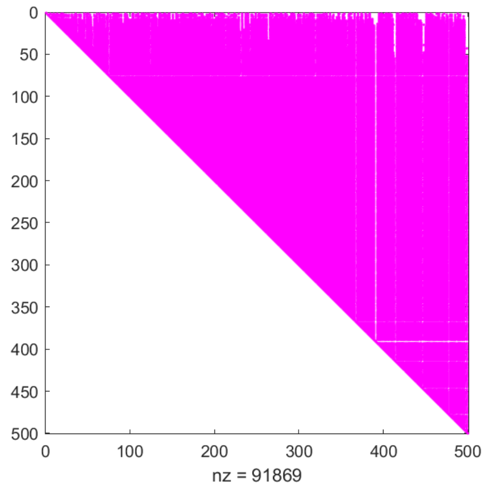

Рисунок 11
Заполнение множителей разложения Холецкого к исходной матрице при степени заполнения 2

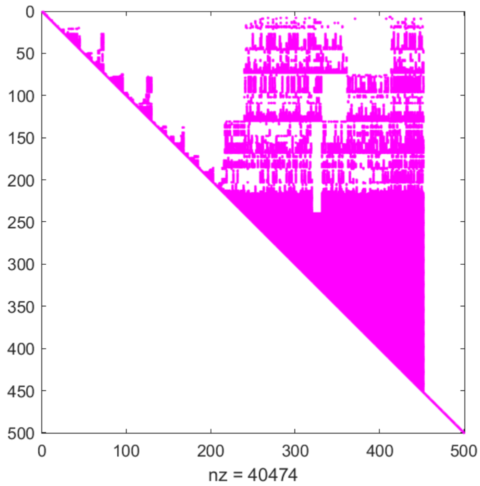

Рисунок 12
Заполнение множителей разложения Холецкого к матрице после применения symamd при степени заполнения 2

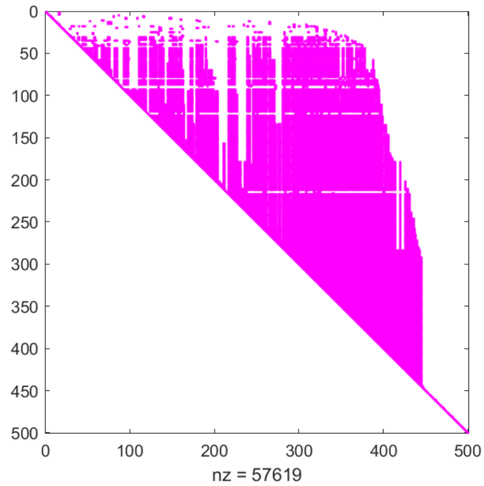

Рисунок 13
Заполнение множителей разложения Холецкого к матрице после применения symrcm при степени заполнения 2

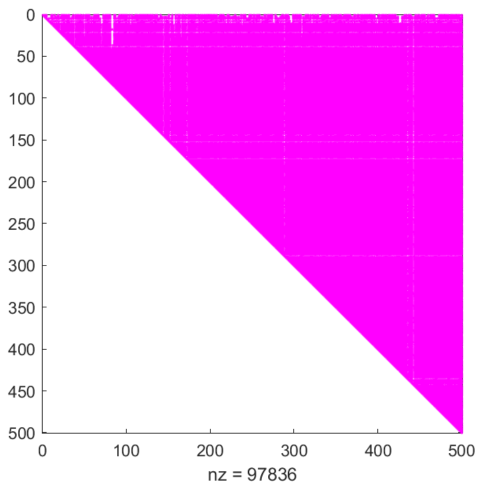

Рисунок 14
Заполнение множителей разложения Холецкого к исходной матрице при степени заполнения 3

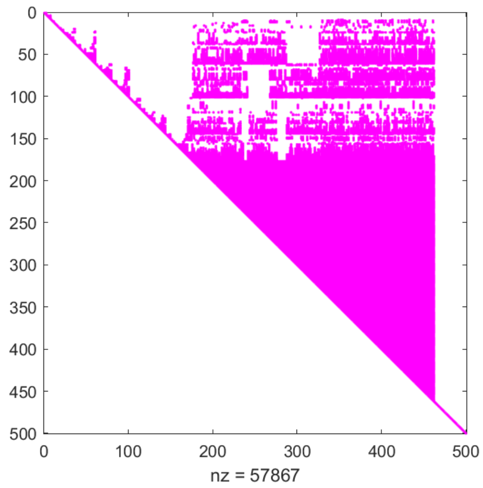

Рисунок 15
Заполнение множителей разложения Холецкого к матрице после применения symamd при степени заполнения 3

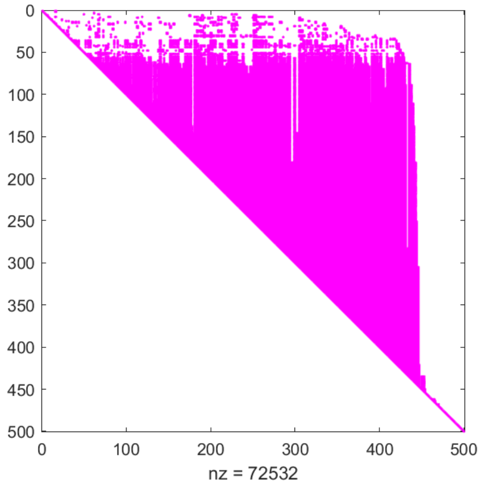

Рисунок 16

Заполнение множителей разложения Холецкого к матрице после применения symrcm при степени заполнения 3

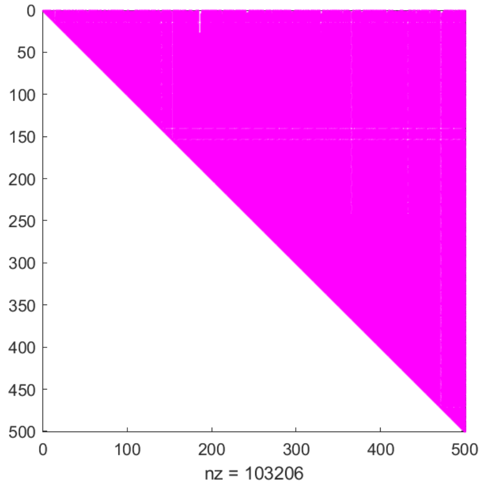

Рисунок 17

Заполнение множителей разложения Холецкого к исходной матрице при степени заполнения 4

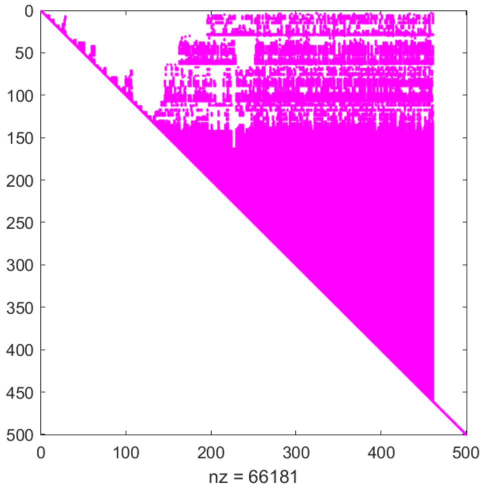

Рисунок 18

Заполнение множителей разложения Холецкого к матрице после применения symamd при степени заполнения 4

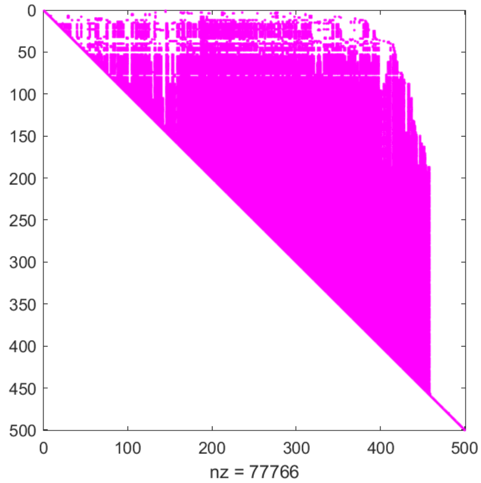

Рисунок 19

Заполнение множителей разложения Холецкого к матрице после применения symrcm при степени заполнения 4

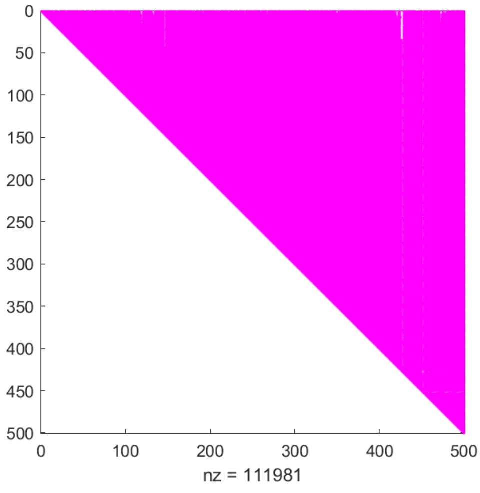

Рисунок 20

Заполнение множителей разложения Холецкого к исходной матрице при степени заполнения 5

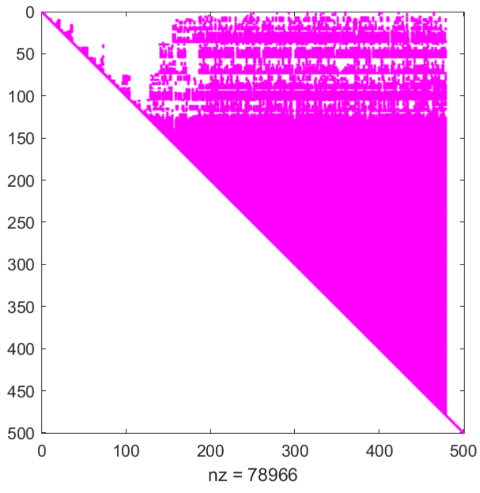

Рисунок 21

Заполнение множителей разложения Холецкого к матрице после применения symamd при степени заполнения 5

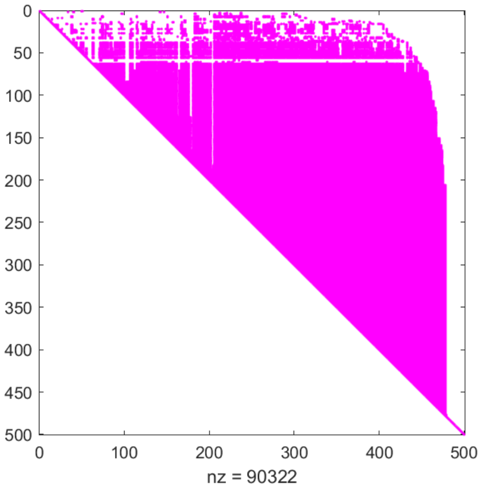

Рисунок 22
Заполнение множителей разложения Холецкого к матрице после применения symrcm при степени заполнения 5

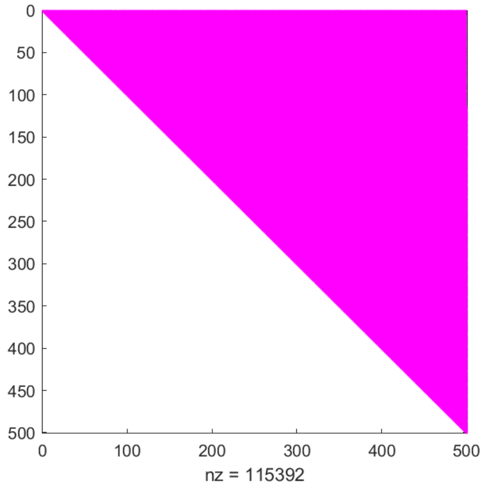

Рисунок 23
Заполнение множителей разложения Холецкого к исходной матрице при степени заполнения 6

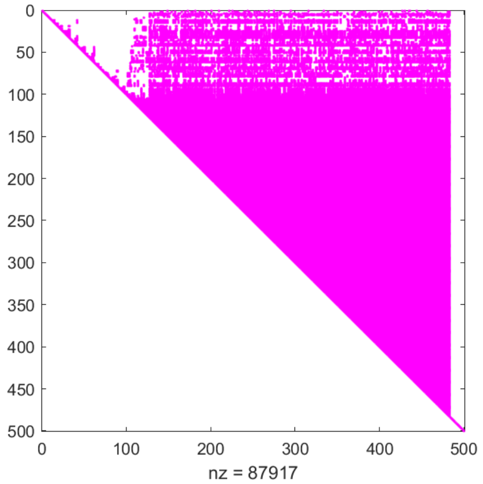

Рисунок 24
Заполнение множителей разложения Холецкого к матрице после применения symamd при степени заполнения 6

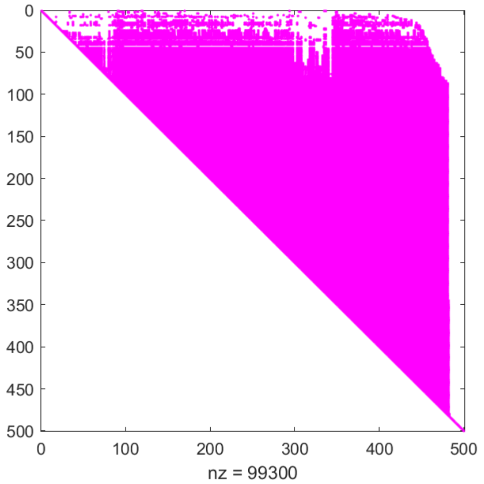

Рисунок 25
Заполнение множителей разложения Холецкого к матрице после применения symrcm при степени заполнения 6

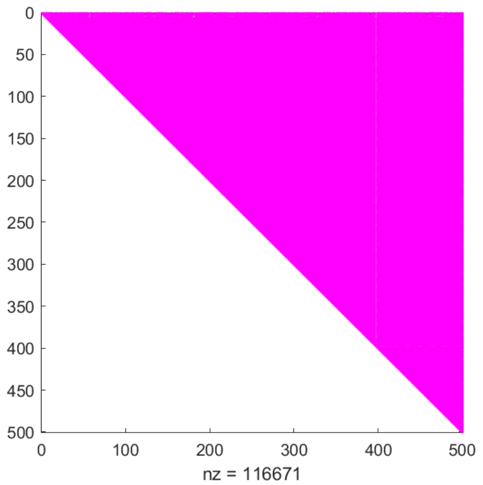

Рисунок 26
Заполнение множителей разложения Холецкого к исходной матрице при степени заполнения 7

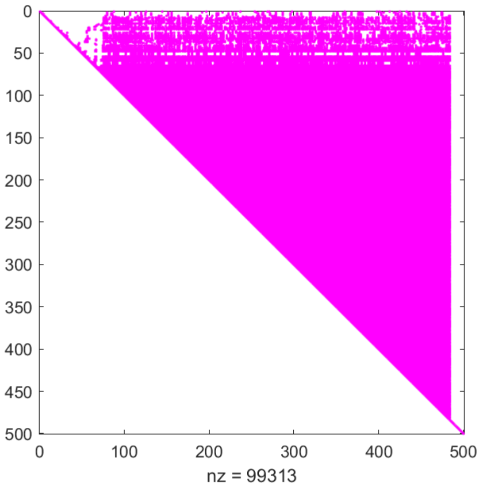

Рисунок 27
Заполнение множителей разложения Холецкого к матрице после применения symamd при степени заполнения 7

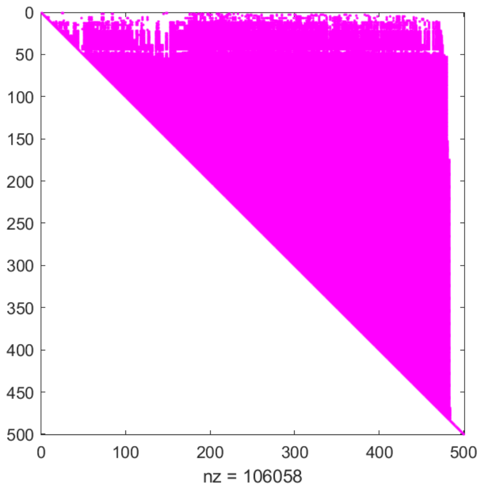

Рисунок 28
Заполнение множителей разложения Холецкого к матрице после применения symrcm при степени заполнения 7

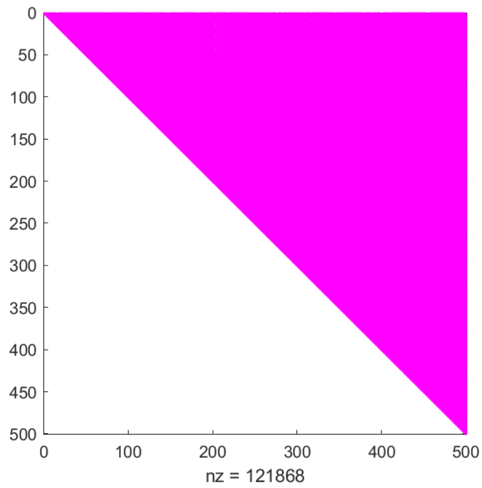

Рисунок 29
Заполнение множителей разложения Холецкого к исходной матрице при степени заполнения 8

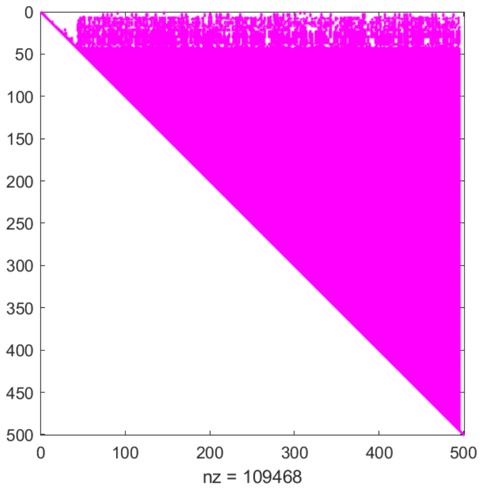

Рисунок 30
Заполнение множителей разложения Холецкого к матрице после применения symamd при степени заполнения 8

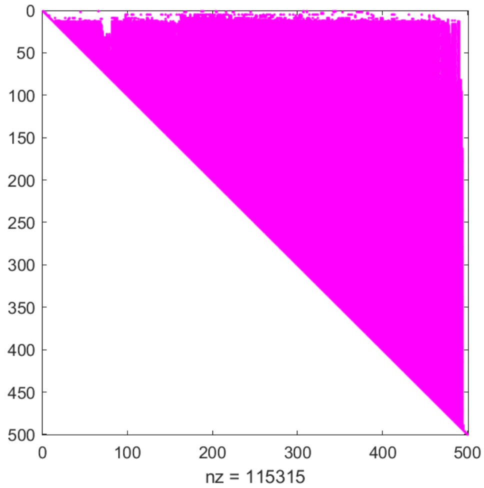

Рисунок 31
Заполнение множителей разложения Холецкого к матрице после применения symrcm при степени заполнения 8

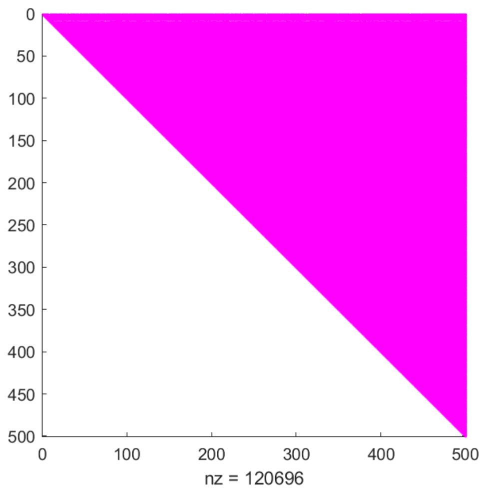

Рисунок 32
Заполнение множителей разложения Холецкого к исходной матрице при степени заполнения 9

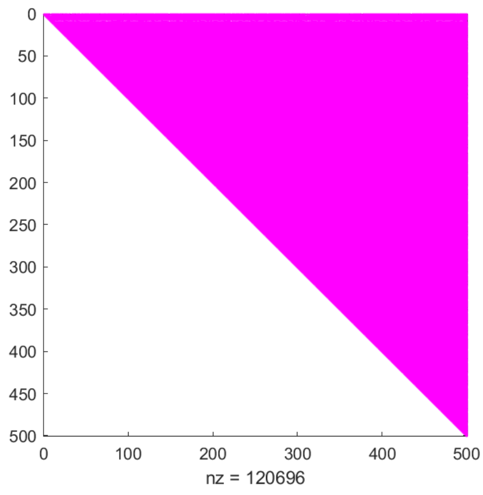

Рисунок 33
Заполнение множителей разложения Холецкого к матрице после применения symamd при степени заполнения 9

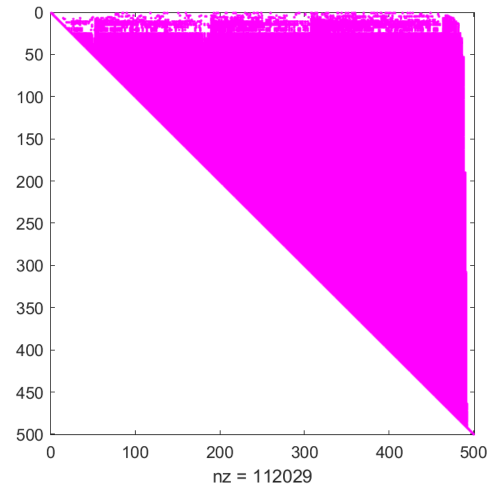

Рисунок 34
Заполнение множителей разложения Холецкого к матрице
после применения symrcm при степени заполнения 9

Временные затраты

На рисунках 35–39 изображены графики временных затрат на разложение Холецкого исходной матрицы, предварительное применение функций symrcm, symamd и разложение Холецкого матрицы после применения symrcm и столбцами при помощи symrcm и symamd. Из рисунка 33 видно, что время, необходимое на разложение Холецкого, не имеет корреляции со степенью заполнения матрицы. На рисунках 34-37 видно, что затрачиваемое время растет вместе с ростом степени заполнения, что связано с увеличением количества ненулевых элементов и, как следствие, увеличение количества вычислений.

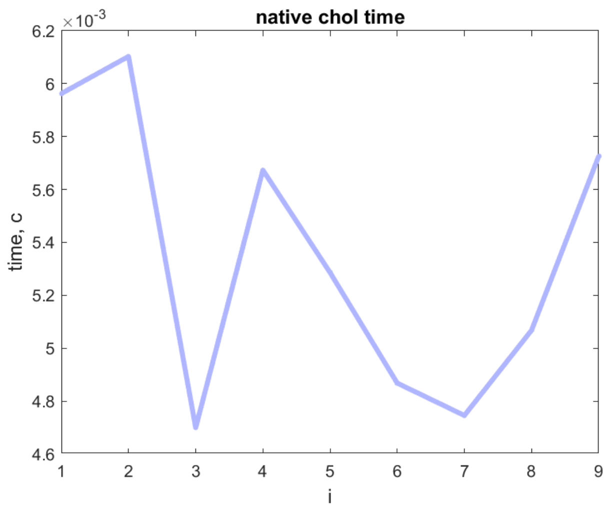

Рисунок 35
Временя выполнения разложения Холецкого исходной матрицы от степени заполнения

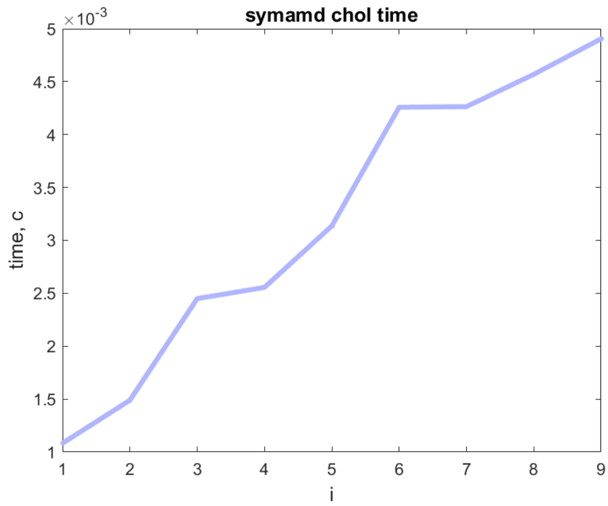

Рисунок 36
Временя выполнения разложения Холецкого матрицы после применения symamd от степени заполнения

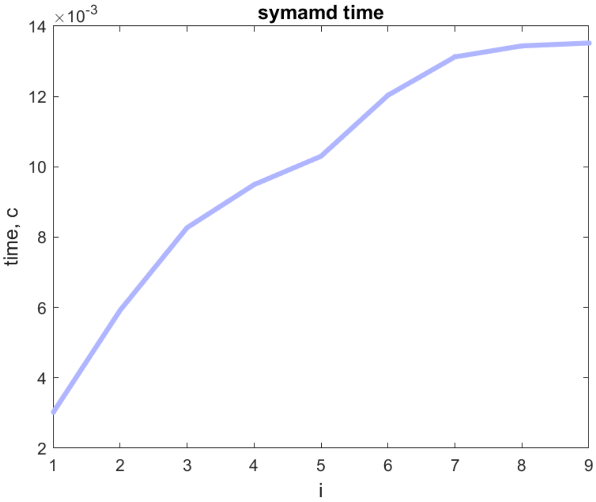

Рисунок 37
Временя выполнения функции symamd к исходной матрице от степени заполнения

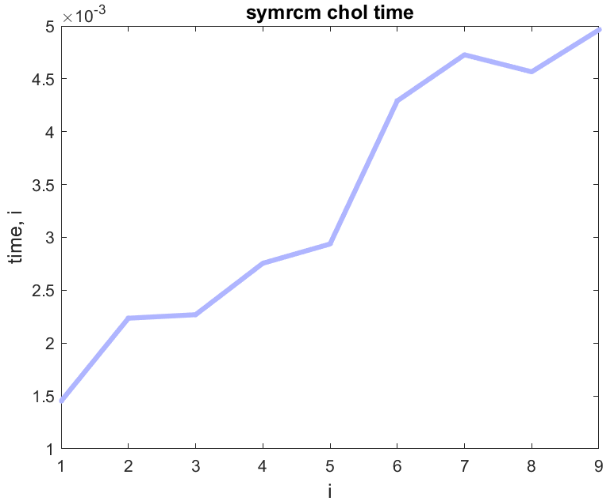

Рисунок 38
Временя выполнения разложения Холецкого матрицы после применения symrcm от степени заполнения

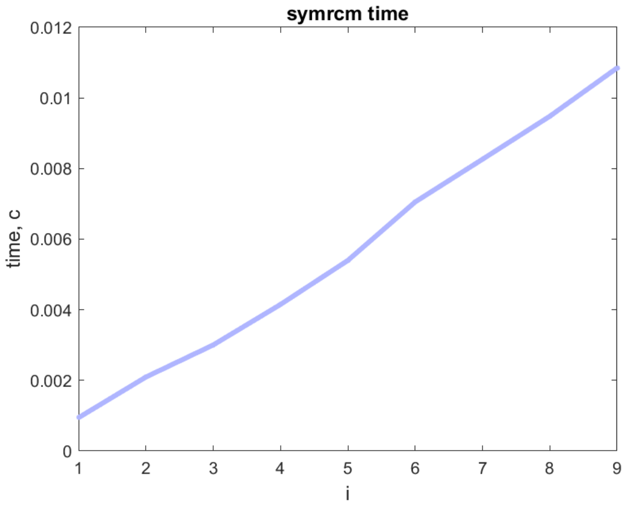

Рисунок 39
Временя выполнения функции symrcm к исходной матрице от степени заполнения

Вывод

В ходе проделанной лабораторной работы нами был исследован предобусловленный метод сопряженных градиентов и исследована эффективность обратного алгоритма Катхилла-Макки и минимальной степени в зависимости от степени заполнения разреженной матрицы. Метод был реализован на языке Fortran, затем по полученным результатам с помощью пакета Matlab были построены графики зависимостей. Исследование же (вместе с построением графиков) было проведено с помощью пакета Matlab.