

Санкт – Петербургский политехнический университет Петра Великого
Физико-механический институт

Отчет по лабораторной работе №2 по Численным методам
«Алгебраическая проблема собственных значений»
Метод обратных итераций (+ сдвиг), LR-алгоритм

Выполнил студент

Группы 5030301/10002

Тугай В.В.

Преподаватель:

К.Н. Козлов

Санкт-Петербург

2023

Содержание отчета:

1. Ручной расчет
2. Метод обратных итераций (+ сдвиг)
3. LR-алгоритм
4. Структура программы
5. Зависимость погрешности от числа обусловленности
6. Зависимость погрешности от возмущения
7. Проверка работы метода для симметричных и несимметричных матриц
8. Зависимость погрешности от точности
9. Зависимость числа итераций от точности
10. Создание матриц с различной отделимостью собственных чисел
11. Проверка «на ноль»
12. Работоспособность, время выполнения и погрешность для функции eig
13. Работоспособность и время выполнения для функции eigs
14. Реализация QR-алгоритма с приведением матрицы к форме Хессенберга
15. Выводы

Вариант 21

Метод обратных итераций (+ сдвиг), LR-алгоритм

Ручной расчет

Метод обратных итераций со сдвигом

$$A = \begin{pmatrix} 1 & -2 & -23 & -4 \\ 4 & 3 & 2 & -31 \\ -5 & -3 & -1 & 5 \\ -3 & -5 & -12 & 1 \end{pmatrix}, \quad \bar{x}^0 = \begin{pmatrix} 0,9 \\ 0,9 \\ 0,9 \\ 0,9 \end{pmatrix}, \quad \varepsilon = 0,01$$

$$\bar{x}^{1+}: \quad (A - 2E) = \begin{pmatrix} 1 & -2 & -23 & -4 \\ 4 & 1 & 2 & -31 \\ -5 & -3 & -1 & 5 \\ -3 & -5 & -12 & 1 \end{pmatrix}, \quad L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0,39 & 1 & 0 & 0 \\ -0,89 & -1,38 & 1 & 0 \\ -0,29 & -0,39 & 0,39 & 1 \end{pmatrix}, \quad U = \begin{pmatrix} 1,09 & -2 & -23 & -4 \\ 0 & 2,89 & 1,11 & -2,94 \\ 0 & 0 & 2,04 & -3,46 \\ 0 & 0 & 0 & 2,1 \end{pmatrix}$$

$$x_1 = \frac{0,47}{-0,89}, \quad x_2 = \frac{9}{11,91}, \quad x_3 = \frac{-3,96}{1,11}, \quad x_4 = \frac{1,08}{0,0031}, \quad \bar{x}^1 = \left(\frac{x_1}{g_1}, \dots, \frac{x_4}{g_4} \right)^T, \quad \bar{x}^1 = 5,064, \quad |x^1 - x^0| > \varepsilon$$

$$\bar{x}^{2+}: \quad (A - 2E) = \begin{pmatrix} 5,936 & -2 & -23 & -4 \\ 4 & -2,04 & 2 & -31 \\ -5 & -3 & -6,06 & 5 \\ -3 & -5 & -12 & -4,063 \end{pmatrix}, \quad L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0,64 & 1 & 0 & 0 \\ -0,89 & 6,54 & 1 & 0 \\ -0,5 & 6,39 & 0,11 & 1 \end{pmatrix}, \quad U = \begin{pmatrix} 5,936 & -2 & -23 & -4 \\ 0 & -0,94 & 1,15 & -2,83 \\ 0 & 0 & 1,22 & -1,563 \\ 0 & 0 & 0 & 3,85 \end{pmatrix}$$

$$x_1 = \frac{0,43}{-0,89}, \quad x_2 = \frac{9}{11,91}, \quad x_3 = \frac{-3,96}{1,11}, \quad x_4 = \frac{1,08}{0,0031}, \quad \bar{x}^2 = 1,443, \quad |x^2 - x^1| > \varepsilon$$

$$\bar{x}^{3+}: \quad (A - 2E) = \begin{pmatrix} 9,53 & -2 & -23 & -4 \\ 4 & 6,527 & 2 & -31 \\ -5 & -3 & -2,47 & 5 \\ -3 & -5 & -12 & -0,493 \end{pmatrix}, \quad L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0,412 & 1 & 0 & 0 \\ -0,52 & -6,21 & 1 & 0 \\ -0,32 & -0,15 & 0,52 & 1 \end{pmatrix}, \quad U = \begin{pmatrix} 9,53 & -2 & -23 & -4 \\ 0 & 2,26 & 1,16 & -2,94 \\ 0 & 0 & 5,4 & -4,4 \\ 0 & 0 & 0 & 0,38 \end{pmatrix}$$

$$x_1 = \frac{0,43}{-0,89}, \quad x_2 = \frac{9}{11,91}, \quad x_3 = \frac{-3,96}{1,11}, \quad x_4 = \frac{1,08}{0,0031}, \quad \bar{x}^3 = 1,244, \quad |x^3 - x^2| > \varepsilon$$

$$\bar{x}^{4+}: \quad (A - 2E) = \begin{pmatrix} 9,48 & -2 & -23 & -4 \\ 4 & 1,48 & 2 & -31 \\ -5 & -3 & -2,2 & 5 \\ -3 & -5 & -12 & -0,22 \end{pmatrix}, \quad L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0,411 & 1 & 0 & 0 \\ -0,51 & -1,51 & 1 & 0 \\ -0,31 & -0,15 & 0,52 & 1 \end{pmatrix}, \quad U = \begin{pmatrix} 9,48 & -2 & -23 & -4 \\ 0 & 2,6 & 1,4 & -2,94 \\ 0 & 0 & 3,6 & -4,4 \\ 0 & 0 & 0 & 0,39 \end{pmatrix}$$

$$x_1 = \frac{0,43}{-0,89}, \quad x_2 = \frac{9}{11,91}, \quad x_3 = \frac{-3,96}{1,11}, \quad x_4 = \frac{1,08}{0,0031}, \quad \bar{x}^4 = 1,241, \quad |x^4 - x^3| > \varepsilon$$

$$\bar{x}^{5+}: \quad (A - 2E) = \begin{pmatrix} 9,45 & -2 & -23 & -4 \\ 4 & 1,45 & 2 & -31 \\ -5 & -3 & -2,2 & 5 \\ -3 & -5 & -12 & -0,24 \end{pmatrix}, \quad L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0,411 & 1 & 0 & 0 \\ -0,51 & -0,61 & 1 & 0 \\ -0,31 & -0,15 & 0,52 & 1 \end{pmatrix}, \quad U = \begin{pmatrix} 9,45 & -2 & -23 & -4 \\ 0 & 2,6 & 1,4 & -2,94 \\ 0 & 0 & 3,6 & -4,4 \\ 0 & 0 & 0 & 0,40 \end{pmatrix}$$

$$x_1 = \frac{0,43}{-0,89}, \quad x_2 = \frac{9}{11,91}, \quad x_3 = \frac{-3,96}{1,11}, \quad x_4 = \frac{1,08}{0,0031}, \quad \bar{x}^5 = 1,241, \quad |x^5 - x^4| > \varepsilon$$

$$\Rightarrow x = 1,241$$

LR - algorithm

$$A_0 = \begin{pmatrix} 3 & -1 & 10 \\ 2 & 4 & -6 \\ 2 & -1 & 5 \end{pmatrix}, \text{eps} = 0.1$$

$$i=1: L_0 = \begin{pmatrix} 1 & 0 & 0 \\ \frac{2}{3} & 1 & 0 \\ \frac{2}{3} & -\frac{1}{3} & 1 \end{pmatrix}, U_0 = \begin{pmatrix} 3 & -1 & 10 \\ 0 & 19 & -38 \\ 0 & 0 & \frac{1}{3} \end{pmatrix}, A_1 = U_0 L_0 = \begin{pmatrix} 9 & -1\frac{5}{3} & 10 \\ -5\frac{1}{3} & 5\frac{4}{3} & -12\frac{2}{3} \\ -\frac{1}{3} & \frac{2}{3} & -2\frac{2}{3} \end{pmatrix}$$

$$\cancel{10 = 3 + 2 \cdot 0.1} / -2\frac{2}{3} - 5/2 \text{eps}$$

$$i=2: L_1 = \begin{pmatrix} 1 & 0 & 0 \\ -\frac{16}{27} & 1 & 0 \\ -\frac{4}{21} & -\frac{2}{242} & 1 \end{pmatrix}, U_1 = \begin{pmatrix} 9 & -\frac{12}{3} & 10 \\ 0 & 242 & -182 \\ 0 & 63 & -\frac{121}{121} \end{pmatrix}, A_2 = U_1 L_1 = \begin{pmatrix} 8\frac{1}{3} & -2\frac{27}{242} & 10 \\ -1\frac{704}{246} & 3\frac{828}{242} & -1\frac{61}{121} \\ \frac{140}{247} & \cancel{\frac{221}{242}} & -\frac{141}{121} \end{pmatrix}$$

$$\cancel{18\frac{1}{3} = 3 + 2\frac{4}{3}} / -\frac{141}{242} + 2\frac{4}{3} / 2 \text{eps}$$

$$i=3: L_2 = \begin{pmatrix} 1 & 0 & 0 \\ -\frac{146}{525} & 1 & 0 \\ \frac{112}{5179} & \frac{332}{2891} & 1 \end{pmatrix}, U_2 = \begin{pmatrix} \frac{43}{5} & -\frac{1646}{389} & 10 \\ 0 & \frac{8015}{2767} & \frac{1226}{2767} \\ 0 & 0 & \frac{-47}{41} \end{pmatrix}, A_3 = U_2 L_2 = \begin{pmatrix} 8\frac{12}{14} & -1\frac{17}{20} & 10 \\ -\frac{85}{105} & 3\frac{25}{89} & \frac{41}{43} \\ -\frac{528}{2131} & -\frac{17}{1062} & -\frac{16}{41} \end{pmatrix}$$

$$\cancel{17\frac{12}{14} = 3 + 1\frac{6}{41}} / -\frac{141}{242} + 1\frac{6}{41} / 2 \text{eps} \quad " -1,146 \frac{3}{41}$$

$$i=4: L_3 = \begin{pmatrix} 1 & 0 & 0 \\ -\frac{12}{186} & 1 & 0 \\ -\frac{26}{13062} & -\frac{70}{10541} & 1 \end{pmatrix}, U_3 = \begin{pmatrix} \frac{62}{4} & -\frac{39}{20} & 10 \\ 0 & \frac{2231}{2246} & \frac{7313}{3916} \\ 0 & 0 & \frac{-3411}{3083} \end{pmatrix}, A_4 = U_3 L_3 = \begin{pmatrix} 9,0047 & -2\frac{143}{10540} & 10 \\ -\frac{1332}{4467} & 3.2 & 7\frac{337}{392} \\ \frac{31}{9449} & \frac{1}{136} & -1,102 \end{pmatrix}$$

$$| -1,102 + 1,146 | < \text{eps}$$

$$\Rightarrow \lambda_1 = 29,0047; \lambda_2 = 3,2; \lambda_3 = -1,102 \quad (\text{zu } A_4)$$

für $A_0: \lambda_1 = 9; \lambda_2 = 4; \lambda_3 = -1$

$$\text{error}(\lambda_1) = 0,0047; \text{error}(\lambda_2) = 0,8; \text{error}(\lambda_3) = 0,102$$

Метод обратных итераций (+ сдвиг)

Алгоритм метода обратных итераций со сдвигом:

Пусть имеем собственное значение λ матрицы A . Запишем уравнение и построим по нему итерационный процесс: $(A - \lambda E)y^{(s+1)} = y^{(s)}$. На каждой итерации будем вычислять отношение скалярных произведений $\frac{(y^{(s+1)}, y^{(s+1)})}{(y^{(s)}, y^{(s+1)})}$, и это отношение будет стремиться к собственному числу матрицы $(A - \lambda E)^{-1}$, и если у матрицы A собственное значение λ_n , то собственные значения матрицы $(A - \lambda E)$ будут $\lambda_n - \lambda$. Тогда у матрицы $(A - \lambda E)^{-1}$ собственные значения будут $\frac{1}{\lambda_n - \lambda}$. Отсюда можем записать $\frac{(y^{(s+1)}, y^{(s+1)})}{(y^{(s)}, y^{(s+1)})} = \frac{1}{\lambda_n^{(s+1)} - \lambda}$, и если мы зададим начальное приближение $\lambda^{(0)}$, близкое к λ , то сможем записать итерационный процесс $(A - \lambda^{(s)} E)y^{(s+1)} = y^{(s)}$. Тогда получим $\lambda^{(s+1)} = \lambda^{(s)} + \frac{(y^{(s)}, y^{(s+1)})}{(y^{(s+1)}, y^{(s+1)})}$, т. к. вместо λ в итерационном процессе на 1 итерации будет использоваться $\lambda^{(0)}$, а на последующих итерациях вместо λ будут использоваться посчитанные на предыдущих шагах $\lambda^{(s)}$. Также для 1 итерации положим $y^{(0)} \neq 0$, а для упрощения решения системы $(A - \lambda^{(s)} E)y^{(s+1)} = y^{(s)}$ будет использоваться LU-разложение матрицы $(A - \lambda^{(s)} E)$, и на каждой итерации будем находить вектор $x^{(s)} = \frac{y^{(s)}}{\|y^{(s)}\|}$, который является собственным вектором матрицы A .

Алгоритм метода обратных итераций без сдвига отличается тем, что выражение $(A - \lambda E)$ и LU-разложение этой матрицы будет происходить перед итерационным процессом, и вместо λ там будет $\lambda^{(0)}$, $\lambda^{(s)}$ будет находиться как $\lambda^{(s)} = \frac{(y^{(s)}, y^{(s+1)})}{(y^{(s+1)}, y^{(s+1)})}$, и по завершению итерационного процесса собственное найдем как $\lambda = \frac{1}{\lambda^{(s)}} + \lambda^{(0)}$

LR-алгоритм

LR-алгоритм основан на разложении матрицы в произведение треугольных $A=LR$, где L – единичная нижняя треугольная, а R – верхняя треугольная. Предположим, что мы составим преобразование подобия $L^{-1}AL$ матрицы A , тогда $L^{-1}AL = L^{-1}(LR)L = RL$. Следовательно, если разложить A в произведение треугольных, а затем перемножить сомножители в обратном порядке, то получится матрица, подобная исходной. Алгоритм же определяется уравнениями $A_{s-1} = L_{s-1}R_{s-1}$, $R_{s-1}L_{s-1} = A_s$. Здесь A_s подобна A_{s-1} , а значит подобна A_1 . При больших s матрица A_s стремится к виду верхней треугольной матрицы, на диагонали которой лежат собственные числа исходной матрицы A .

Структура программы

Программа matrix.m нужна для создания файлов, в которых будут записаны матрицы А 10x10 с отдельностями от 0.01 до 10, и в дальнейшем именно эти файлы будет считывать программа lab2.f95. Программа lab2.f95 считывает эти файлы и задает переменную и массивы: переменная содержит значения отдельности, а массивы – и есть матрицы А.

Программа lab2.f95 состоит из основного тела и двух модулей:

matrix_fluct.mod - модуль, содержащий функции для внесения случайных возмущений в строку матрицы в пределах процента, заданного пользователем.

ort_givens_reverse_iter_lr.mod - модуль, содержащий подпрограмму с реализацией методов, подпрограмму с 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 lab2.m реализует решение алгебраической проблемы собственных значений с помощью средств Matlab. После выполнения в папке будут созданы рисунки, все они представлены далее в отчете.

Зависимость погрешности от отделимости

На рисунках 1 и 2 изображены графики зависимости погрешности от отделимости в логарифмических осях. Как можно увидеть по рисунку, с увеличением числа обусловленности растет и ошибка. Наименьшая ошибка у метода обратных итераций без сдвига, наибольшая же у LR-алгоритма

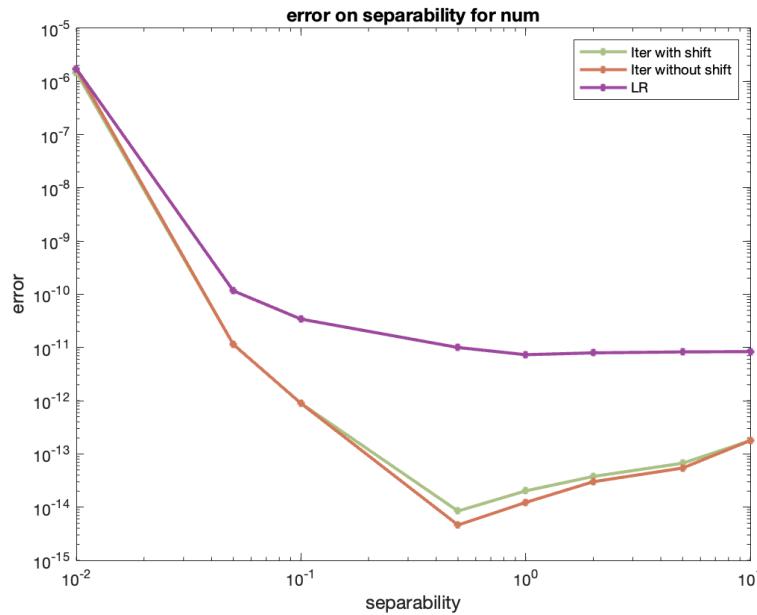


Рисунок 1
Зависимость погрешности от отделимости
для собственных значений

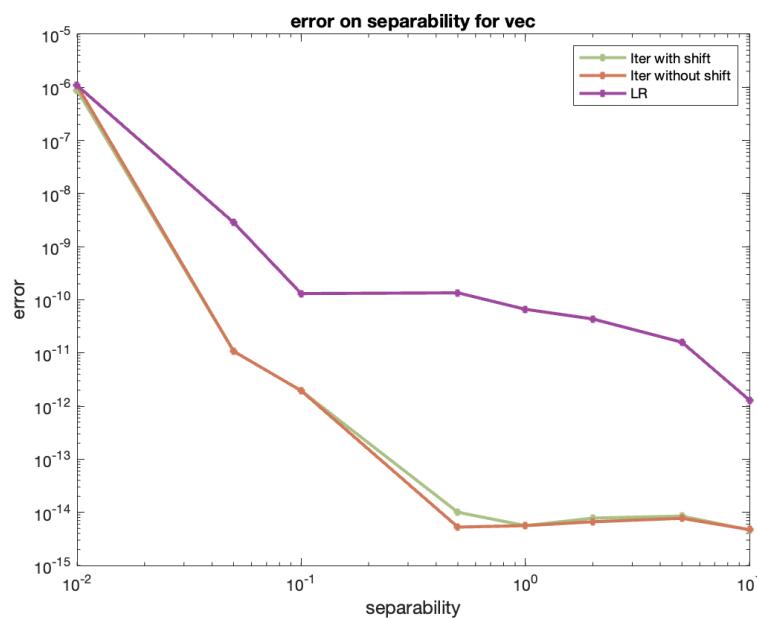


Рисунок 2
Зависимость погрешности от отделимости
для собственных векторов

Зависимость относительной погрешности от возмущения

На рисунках 3 и 4 изображены графики зависимости относительной погрешности от возмущения элементов строки матрицы A. Как видно из рисунков, ошибка растет вместе с увеличением возмущения, вносимого пользователем, для всех трех методов, самых чувствительных к возмущениям оказался метод обратных итераций со сдвигом.

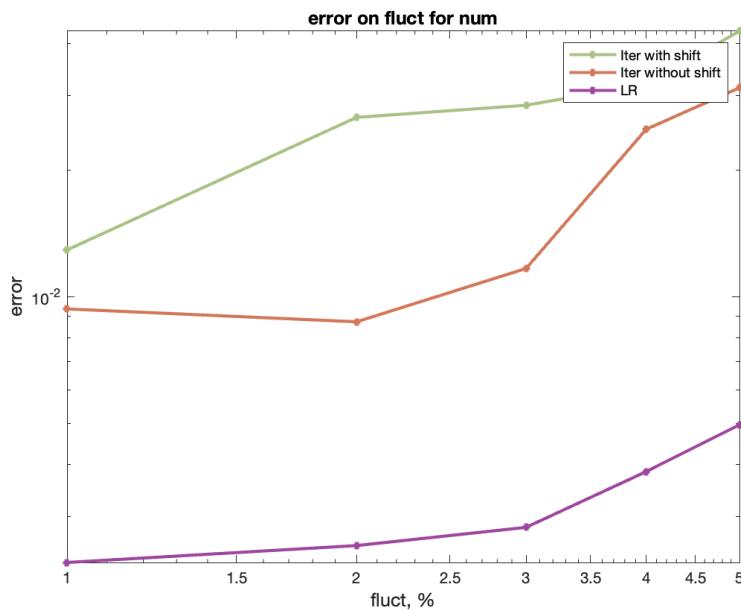


Рисунок 3
Зависимость погрешности от отделимости
для собственных значений

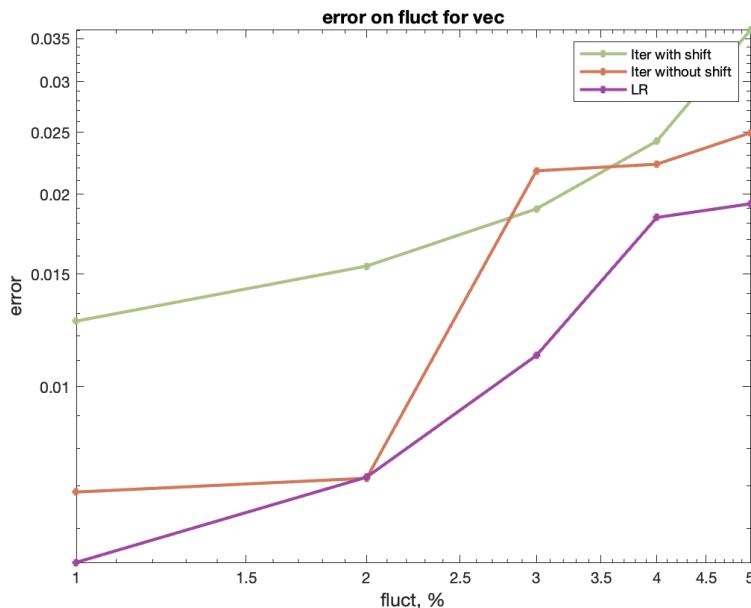


Рисунок 4
Зависимость погрешности от отделимости
для собственных векторов

Проверка работы метода для симметричных и несимметричных матриц

На рисунках 5 и 6 изображены графики зависимости погрешности от отдельности в логарифмических осях. Как можно увидеть по рисунку, все три метода исправно работают как с симметричными, так и несимметричными матрицами, и при отдельности 10 ошибка меньше, чем при отдельности 0.1.

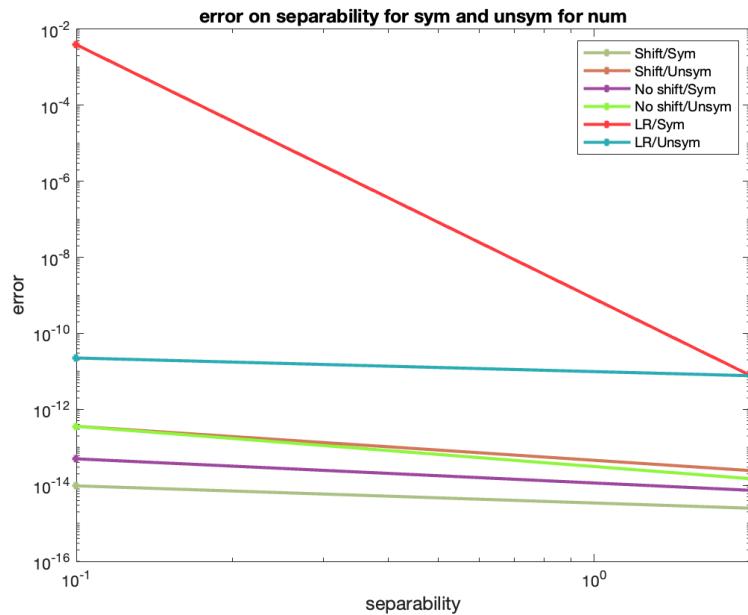


Рисунок 5
Зависимость погрешности от отдельности
для собственных значений

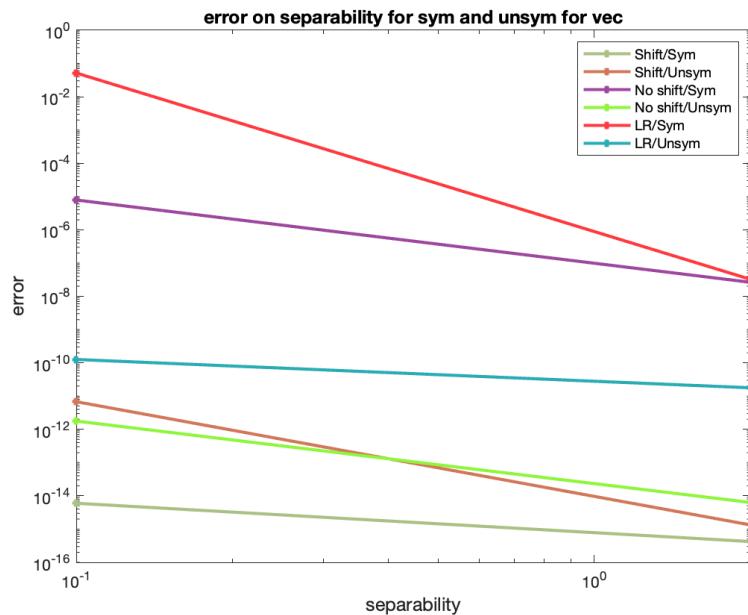


Рисунок 6
Зависимость погрешности от отдельности
для собственных векторов

Зависимость погрешности от точности

На рисунках 7 и 8 изображены графики зависимости погрешности от точности в логарифмических осях. Как можно увидеть по рисункам, с увеличением точности ошибка уменьшается, что соответствует смыслу данного параметра.

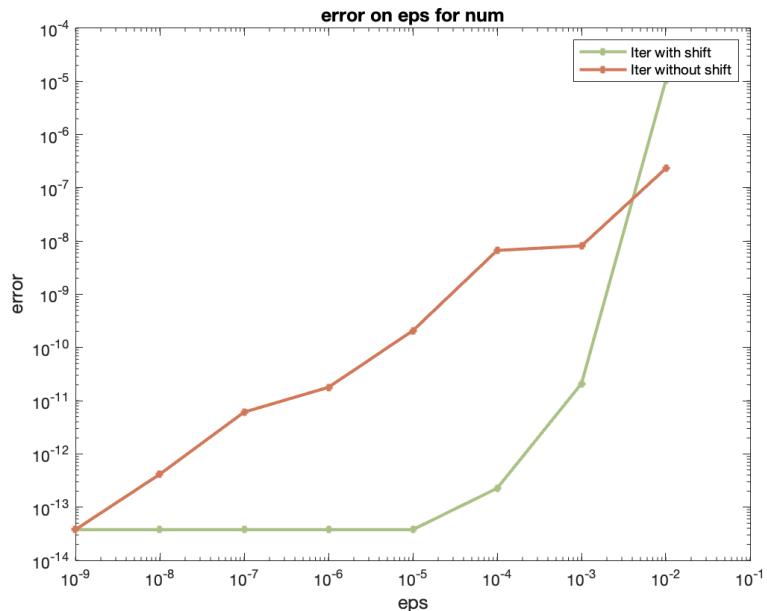


Рисунок 7
Зависимость погрешности от точности
для собственных значений

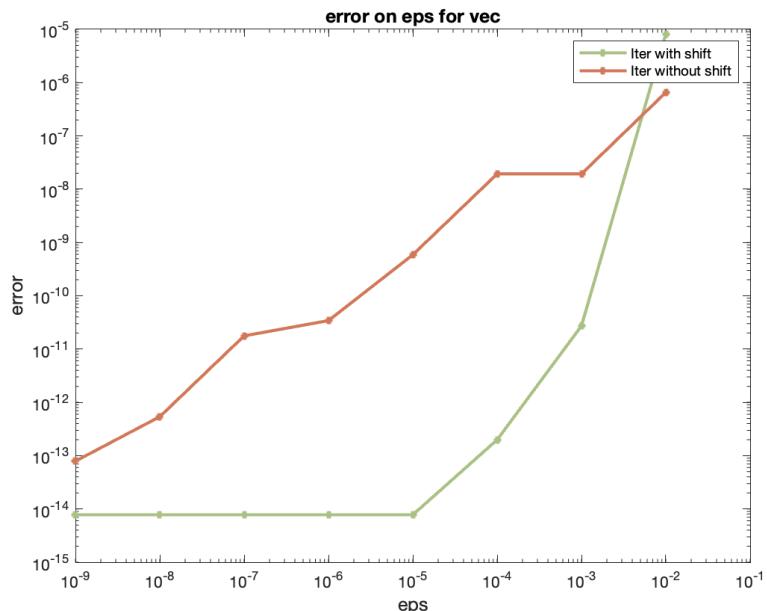


Рисунок 8
Зависимость погрешности от точности
для собственных векторов

Зависимость числа итераций от точности

На рисунках 9 и 10 изображены графики зависимости числа итераций от точности в логарифмических осях. Как можно увидеть по рисункам, с увеличением точности увеличивается и число итераций, необходимое для того, чтобы метод сошелся, что является логичным для итерационного процесса.

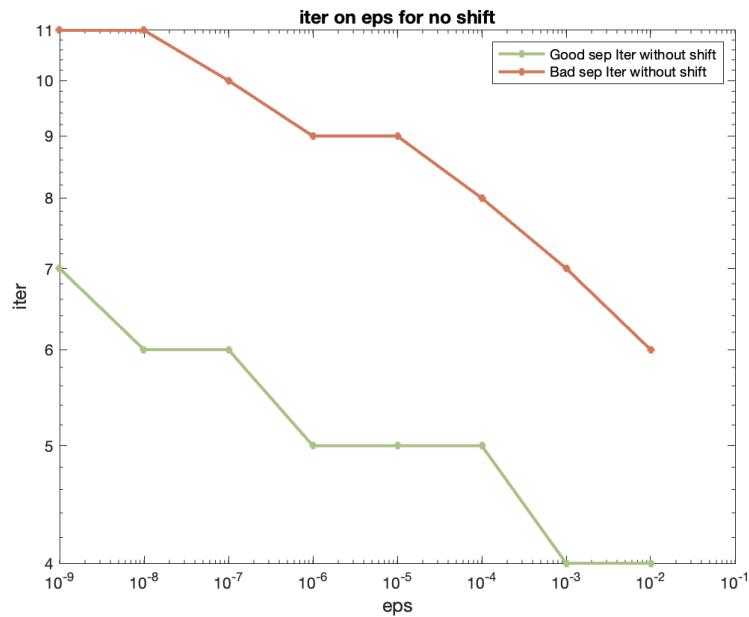


Рисунок 9
Зависимость погрешности от точности
для обратных итераций без сдвига

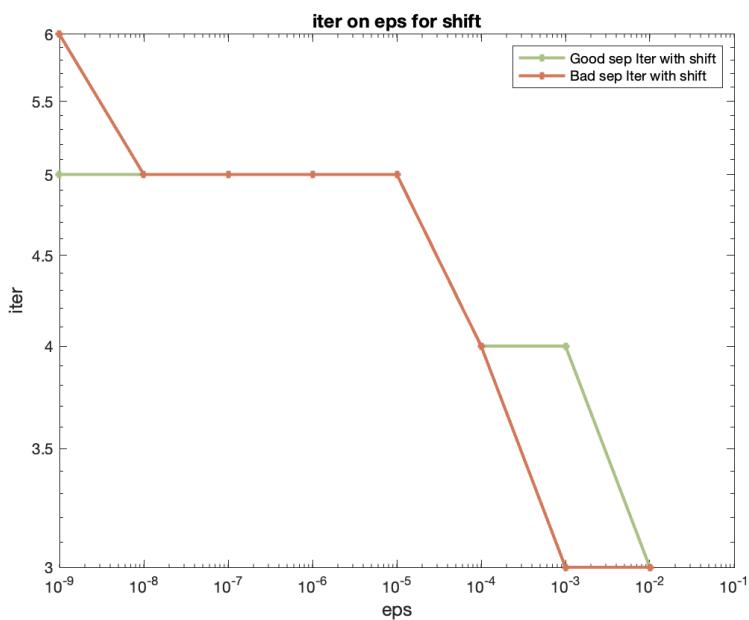


Рисунок 10
Зависимость погрешности от точности
для обратных итераций со сдвигом

Создание матриц с различной отделимостью СЧ

На рисунках 11-24 приведены матрицы через функцию spy, исходя из рисунков можно сделать вывод, что у заданных матриц нет нулевых элементов.

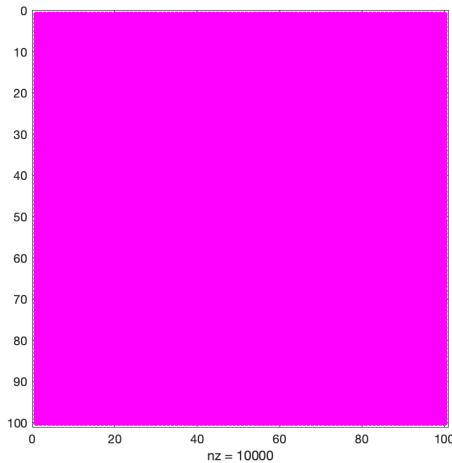


Рисунок 11

Все СЧ хорошо отделимы, вещественны, 100x100

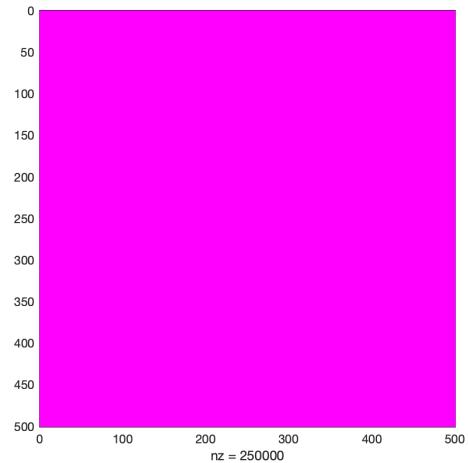


Рисунок 12

Все СЧ хорошо отделимы, вещественны, 500x500

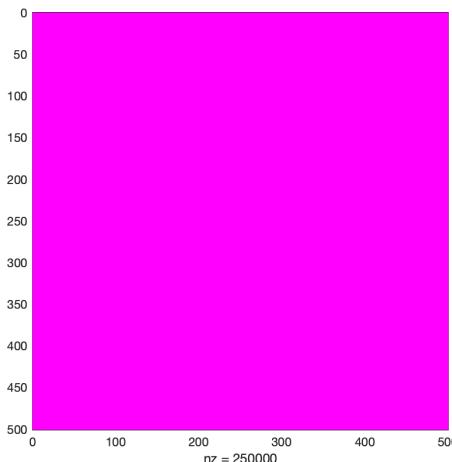


Рисунок 13

Все СЧ хорошо отделимы, вещественны, 500x500,
нулевые элементы заменены на EPS=1e-16

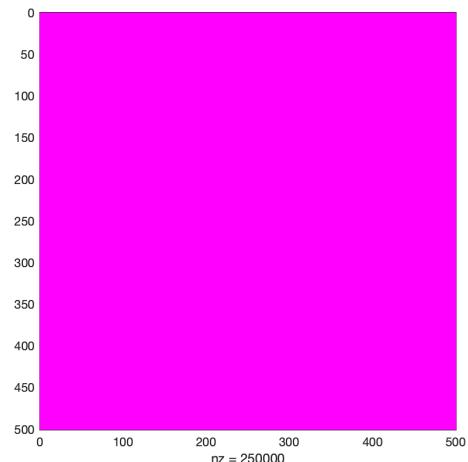


Рисунок 14

Все СЧ хорошо отделимы, вещественны, 500x500,
нулевые элементы заменены на EPS=1e-100

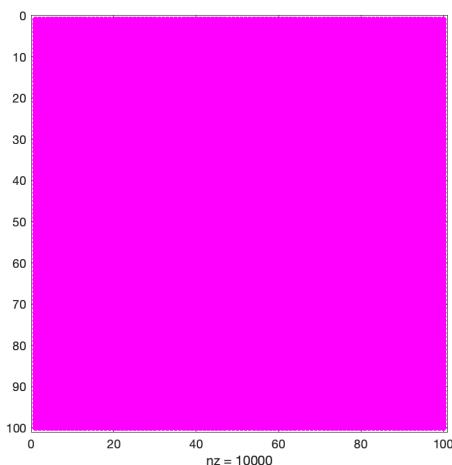


Рисунок 15

Все СЧ плохо отделимы, вещественны, 100x100

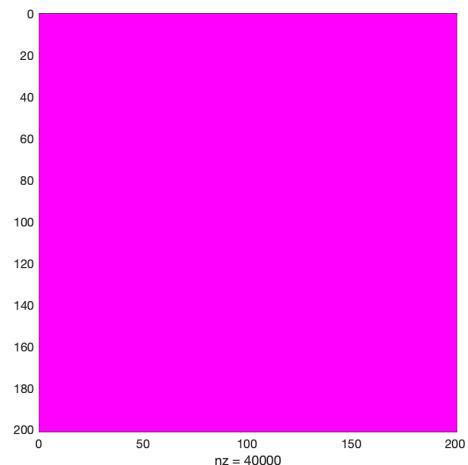


Рисунок 16

Все СЧ плохо отделимы, вещественны, 200x200

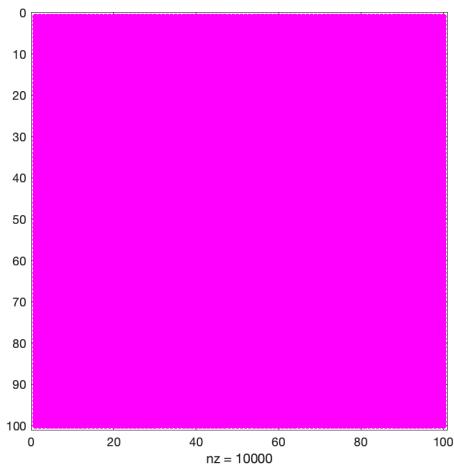


Рисунок 17

Макс. СЧ хорошо отделимы, остальные - нет,
вещественны, 100x100

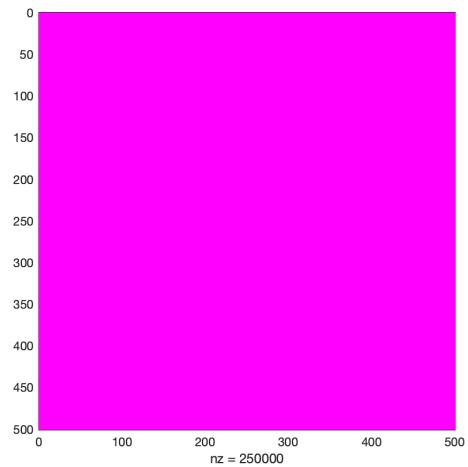


Рисунок 18

Макс. СЧ хорошо отделимы, остальные - нет,
вещественны, 500x500

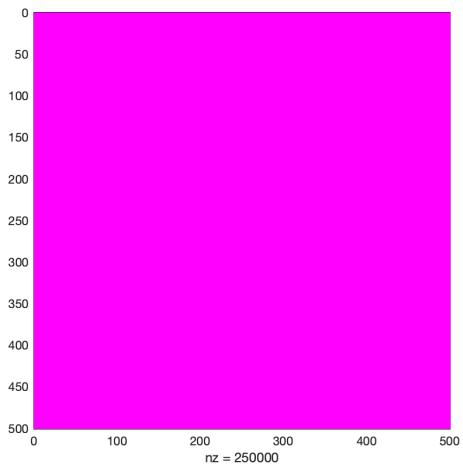


Рисунок 19

Макс. СЧ хорошо отделимы, остальные - нет,
вещественны, 100x100
нулевые элементы заменены на EPS=1e-16

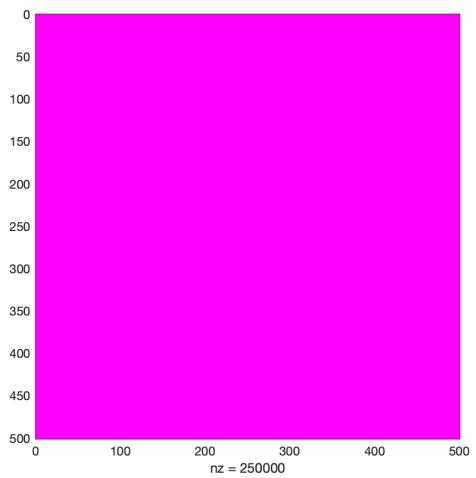


Рисунок 20

Макс. СЧ хорошо отделимы, остальные - нет,
вещественны, 500x500
нулевые элементы заменены на EPS=1e-100

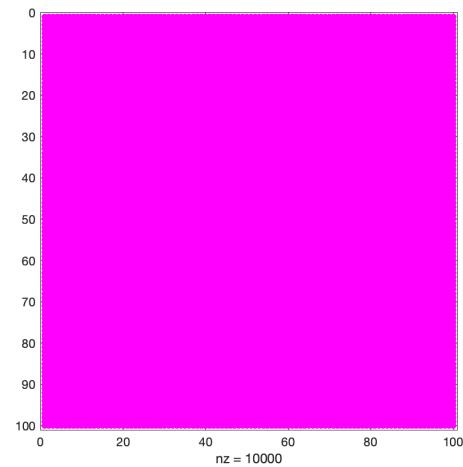


Рисунок 21

Мин. СЧ хорошо отделимы, остальные - нет,
вещественны, 100x100

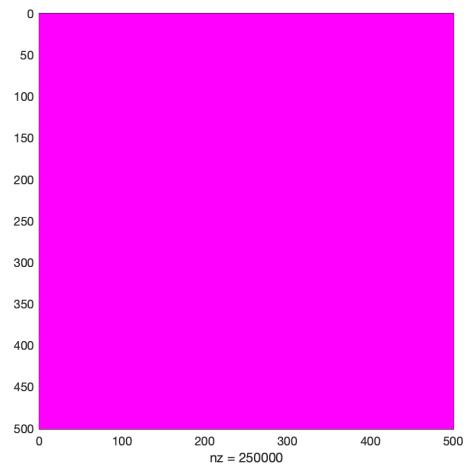


Рисунок 22

Мин. СЧ хорошо отделимы, остальные - нет,
вещественны, 500x500

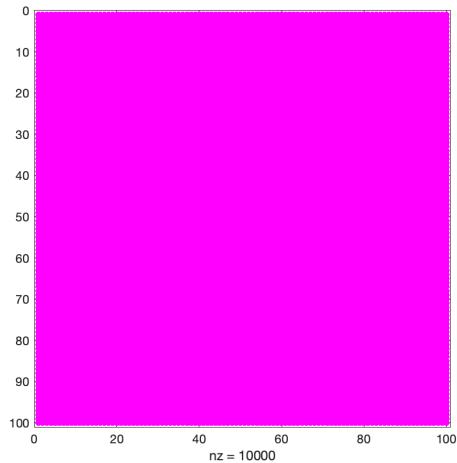


Рисунок 23

Все СЧ хорошо отделимы, несколько пар комплексно-сопряженных, 100x100

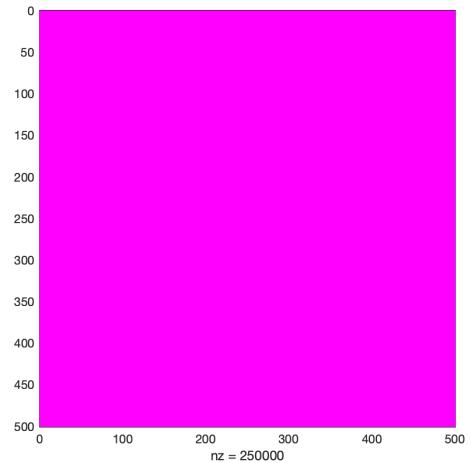


Рисунок 24

Все СЧ хорошо отделимы, несколько пар комплексно-сопряженных, 500x500

Проверка «на ноль»

Проверка «на ноль» происходила следующим образом :

```
[V,D] = eig(B);
disp(i)
zero_check=norm(B*V-V*D)
```

Вывод в консоль:

```
1
zero_check =
1.9394e-12

2
zero_check =
2.1045e-11

3
zero_check =
2.8586e-11

4
zero_check =
2.5905e-11

5
zero_check =
1.8621e-13

6
zero_check =
1.4219e-13

7
zero_check =
8.1719e-14

8
zero_check =
6.1313e-13

9
zero_check =
3.9771e-13

10
zero_check =
4.0156e-13

11
zero_check =
1.2392e-13

12
zero_check =
3.8751e-13

13
zero_check =
1.8290e-12

14
zero_check =
2.3372e-11
```

Значения не превышают 10^{-11} , что говорит об успешной проверки «на ноль» для всех 14 матриц

Работоспособность, время выполнения и погрешность для функции eig

Для всех 14 матриц были проверены возможные вызовы функции eig, такие как:

```

3.1) d = eig(B);
3.2) d = eig(B,C);
3.3) [V,D] = eig(B);
3.4) [V,D] = eig(B, 'nobalance');
3.5) [V,D] = eig(B,C);
3.6) [V,D] = eig(B,C, 'chol');

```

Результаты предоставлены на рисунке 25

3.1 time	3.2 time	3.3 time	3.3 err	3.4 time	3.4 err	3.5 time	3.5 err	3.6 time	3.6 err
0.013830666	0.008859125	0.006957708	1.89275144892915e-12	0.005930333	1.89275144892915e-12	0.007416084	4895.7178492968	0.00582	4895.7178492968
0.148711292	0.231186292	0.244206625	2.09799770570661e-11	0.30898825	2.09799770570661e-11	0.422374875	32080.7702005122	0.373792958	32080.7702005122
0.140577708	0.244744542	0.219463541	2.14417773490585e-11	0.2158015	2.14417773490585e-11	0.419375042	61644.3953986073	0.403900167	61644.3953986073
0.13651025	0.2677515	0.213527958	2.14444765446359e-11	0.220120667	2.14444765446359e-11	0.44143275	44145.2866923823	0.442228	44145.2866923823
0.004651375	0.003678625	0.006185792	1.02250693413511e-13	0.005206375	8.65381275265339e-14	0.005806833	100.818147211261	0.0046075	100.818147211261
0.022944416	0.019193625	0.055149291	1.77360634861632e-13	0.052053791	1.88644182745752e-13	0.030234292	217.350764452106	0.033188292	217.350764452106
0.003261292	0.0032875	0.005387083	6.17181412931108e-14	0.004114334	1.2746079182602e-13	0.005206334	87.8591298161617	0.004668584	87.8591298161617
0.144115625	0.218779834	0.220462333	4.15945814412065e-13	0.211786958	4.6991875477527e-13	0.374159167	875.696254765113	0.402209458	875.696254765113
0.1509785	0.232664833	0.220281083	3.40152251514578e-13	0.211715167	4.70483774314924e-13	0.415291834	736.736796939172	0.394158417	736.736796939172
0.157616583	0.227514708	0.241677583	4.46232912421253e-13	0.223368459	5.04629129193998e-13	0.39135325	861.219916174183	0.38411975	861.219916174183
0.004876	0.004039417	0.005775458	8.04043743613721e-14	0.006308083	1.13967042794303e-13	0.005160959	63.495096412488	0.004967375	63.495096412488
0.154196417	0.257479875	0.233038292	7.59713408841193e-13	0.221860666	4.99748417533508e-13	0.418070791	716.308552537616	0.413358083	716.308552537616
0.029004375	0.028260667	0.035581875	2.3011622253872e-12	0.041016625	2.3011622253872e-12	0.024880083	10102.4289534188	0.024544833	10102.4289534188
0.399106333	0.727474458	0.758704917	2.29877600187433e-11	0.743139917	2.29877600187433e-11	1.022469916	31302.7490628759	1.045750459	31302.7490628759

Рисунок 25
Время вычислений и погрешность функции eig

Работоспособность и время выполнения для функции eigs

Для всех 14 матриц были проверены возможные вызовы функции eigs в 5 вариациях: 'largestabs', 'smallestabs', 'largestreal', 'smallestreal', 'bothendsreal'. Время вычисления предоставлено на рисунке 26

'largestabs'	'smallestabs'	'largestreal'	'smallestreal'	'bothendsreal'
0.795621417	0.235512	0.541057334	0.532511208	0.517298542
0.521421375	0.063810291	0.521060042	0.514972792	0.447822042
0.505637375	0.025585416	0.515773625	0.512914625	0.483467375
0.472633333	0.018956334	0.519095916	0.491490583	0.458994958

Рисунок 26

Время вычислений функции eigs

Реализация QR-алгоритма с приведением матрицы к форме Хессенберга

Реализация классического QR-алгоритм для нахождения всех собственных значений с предварительным приведением матрицы к форме Хессенберга в MATLAB была осуществлена следующим образом:

```
H = hess(B);
err = inf;
k = 0;
while err > 1e-2
    k = k+1;
    [Q1, R1] = qr(H);
    H = R1 * Q1;
    err = norm(tril(H, -1), 'fro');
    if (k>=5000)
        break
    end
end
```

Время вычисления и погрешность предоставлено на рисунке 27

time	error
0.852800125	0.00999177960051607
81.385017042	0.717861830307335
85.81886825	1.32789019041082
81.925224334	0.153191615640133
3.152504	0.880803932330989
10.755861292	2.26493290179337
3.124025958	0.80124786206136
80.307636792	6.19427804343387
78.027772917	6.05135192055285
81.692645042	6.32433857747149
2.9393975	0.821085625278314
79.304802375	6.24293275359389
7.010188625	2.38407918146684
263.078534375	4.80962417222842

Рисунок 27
Время вычислений и погрешность QR-алгоритма

Выводы

В ходе проделанной лабораторной работы нами были исследованы метод обратных итераций без сдвига и со сдвигом и LR-алгоритм, а также проведено исследование решения алгебраической проблемы собственных значений средствами пакета Matlab. Метод и алгоритм были реализованы 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.