2 Прямые методы решения линейных систем

2.1 Метод Гаусса с выбором главного элемента по столбцу

При написании программы нельзя использовать готовые функции для матричного умножения и обращения матриц кроме случаев, когда это явно описано.

Напишите программу для решения линейной системы

$$Ax = b$$

методом Гаусса с выбором главного элемента по столбцу. Требования к программе

1. Программа должна содержать функцию, принимающую на вход матрицу A и правую часть b
2. Внутри функции нужно сначала вычислить матрицы L, U и матрицу перестановки P, соответствующую выбору главного элемента по столбцу (т.е. перестановке строк): $PA = LU$.
3. После этого в функции нужно решить системы с треугольными матрицами с помощью прямой и обратной подстановок
4. Функция должна возвращать матрицы и вектор решения: L, U, P, x.
5. Программа должна вызывать реализованную фукнцию для какой-то матрицы и правой части и выводить норму разницы между полученным решением, и решением, которое возвращает готовая библиотечная функция, например, `numpy.linalg.solve`.

2.2 Метод Гаусса с выбором главного элемента по строке

При написании программы нельзя использовать готовые функции для матричного умножения и обращения матриц кроме случаев, когда это явно описано.

Напишите программу для решения линейной системы

$$Ax = b$$

методом Гаусса с выбором главного элемента по строке. Требования к программе

1. Программа должна содержать функцию, принимающую на вход матрицу A и правую часть b
2. Внутри функции нужно сначала вычислить матрицы L, U и матрицу перестановки Q, соответствующую выбору главного элемента по строке (т.е. перестановке столбцов): $AQ = LU$.

3. После этого в функции нужно решить системы с треугольными матрицами с помощью прямой и обратной подстановок.

4. Функция должна возвращать матрицы и вектор решения: L, U, Q, x.

5. Программа должна вызывать реализованную фукнцию для какой-то матрицы и правой части и выводить норму разницы между полученным решением, и решением, которое возвращает готовая библиотечная функция, например, `numpy.linalg.solve`.

2.3 Метод Холецкого

При написании программы нельзя использовать готовые функции для матричного умножения и обращения матриц кроме случаев, когда это явно описано.

Напишите программу для решения линейной системы

$$Ax = b, A = A^T > 0$$

методом Холецкого. Требования к программе:

1. Программа должна содержать функцию, принимающую на вход матрицу $A = A^T$ и правую часть b.

2. Внутри функции нужно вычислить матрицу : $A = CC^T$

3. После этого нужно решить 2 системы с треугольными матрицами с помощью прямой и обратной подстановок.

4. Функция должна возвращать матрицу C и вектор решения x.

5. Программа должна создавать случайную матрицу $A = A^T > 0$ заданного размера и правую часть, вызывать функцию для этих данных, и выводить норму разницы между полученным и решением из стандартной функции, например, `numpy.linalg.solve`.

   Для проверки правильности разложения, можно использовать функцию `numpy.linalg.cholesky`

2.4 QR-разложение

При написании программы нельзя использовать готовые функции для матричного умножения и обращения матриц кроме случаев, когда это явно описано.

Напишите программу для решения системы $Ax = b$ методом наименьших квадратов с помощью QR-разложения Требования к программе:

1. Программа должна содержать функцию, принимающую на вход матрицу A и вектор правой части b.

2. Функция должна сначала вычислять QR-разложение матрицы $A \in \mathbb{C}^{m \times n}$, $m \geq n$ с помощью модифицированного алгоритма Грамма-Шмидта.

3. После вычисления QR разложения нужно решить систему методом наименьших квадратов. **Для умножения на матрицу Q^T можно использовать готовую функцию**. Для решения системы с треугольной матрицей R нужно реализовать метод обратной подстановки.

4. Функция должна возвращать матрицы Q, R и вектор решения x.

5. Программа должна вызывать реализованную функцию для

   (a) квадратной невырожденной матрицы

   (b) прямоугольной матрицы с $m > n$ с линейно независимыми столбцами

   и выводить норму разницы между полученными решениями, и решениями из стандартных функций (например, `numpy.linalg.solve`, `numpy.linalg.lstsq`)