

### 3 Итерационные методы решения линейных систем

#### 3.1 Метод Ричардсона

Напишите программу для решения линейной системы с действительной матрицей

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

итерационным методом Ричардсона.

Требования к программе:

1. Программа должна содержать функцию, которая принимает на вход матрицу  $A$ , правую часть  $b$ , итерационный параметр  $\tau$  максимальное число итераций и параметр  $tol$ , задающий критерий остановки для нормы невязки. Функция должна возвращать приближенное решение  $y$ , и одномерный массив со значениями нормы невязки на каждой итерации. Остановка должна происходить при норме невязки меньше заданного порога  $tol$ .
2. Программа должна создавать матрицу  $A = A^T$  произвольного размера  $n > 100$ , правую часть  $b$ .
3. Программа должна вычислять оценку собственных чисел с помощью кругов Гершгорина, и вычислять точные собственные числа с помощью готовой функции (например, `numpy.linalg.eigvals`)
4. Программа должна вычислять приближенное решение с помощью реализованной функции при 3-х различных значениях итерационного параметра:
  - (a) Произвольное значения из допустимой области
  - (b) Оптимальное значение, вычисленное по оценкам с.ч.
  - (c) Оптимальное значение, вычисленное по точным с.ч.
5. Программа должна выводить норму разницы между 3-мя приближенными решениями и точным решением, вычисленным с помощью готовой функции, например, `numpy.linalg.solve`

6. Программа должна выводить время работы готовой функции (с помощью которой вычислялось точное решение), и время работы написанной функции.

Для ускорения можно использовать декоратор `@jit` из пакета `numba`.

7. Программа должна строить графики зависимости логарифма нормы невязки от номера итерации для 3-х значений итерационного параметра (на одном рисунке).
8. Автор программы должен уметь объяснить полученные результаты на основе изученной теории.

## 3.2 Метод Зейделя

**В итерационном методе нельзя использовать обращение матриц и матричное умножение: нужно реализовать метод с помощью циклов.**

Напишите программу для решения линейной системы с действительной матрицей

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

методом Зейделя.

Требования к программе:

1. Программа должна содержать функцию, которая принимает на вход матрицу  $A$ , правую часть  $b$ , максимальное число итераций и параметр  $tol$  для ограничения на норму невязки для критерия остановки итераций. Программа должна возвращать приближенное решение, вычисленное методом Зейделя, и массив со значениями невязки на каждой итерации.
2. Программа должна создавать симметричную положительно определенную матрицу размера  $n > 100$ , правую часть и вызывать реализованную функцию.
3. Программа должна выводить число итераций, точную ошибку (вычисленную по точному решению), а также график зависимости логарифма невязки от номера итерации.

4. Программа должна выводить время работы готовой функции (с помощью которой вычислялось точное решение), и время работы написанной функции.

Для ускорения можно использовать декоратор `@jit` из пакета `numba`.

5. Автор программы должен уметь объяснить полученные результаты на основе изученной теории.

### 3.3 Симметричный метод Зейделя

**В итерационном методе нельзя использовать обращение матриц и матричное умножение: нужно реализовать метод с помощью циклов.**

Напишите программу для решения линейной системы с действительной матрицей

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

симметричным методом Зейделя (чередуются итерации с матрицами  $L + D$  и  $U + D$ )

Требования к программе:

1. Программа должна содержать функцию, которая принимает на вход матрицу  $A$ , правую часть  $b$ , максимальное число итераций и параметр  $tol$  для ограничения на норму невязки для критерия остановки итераций. Программа должна возвращать приближенное решение, вычисленное методом Зейделя, и массив со значениями невязки на каждой итерации.
2. Программа должна создавать симметричную положительно определенную матрицу размера  $n > 100$ , правую часть и вызывать реализованную функцию.
3. Программа должна выводить число итераций, точную ошибку (вычисленную по точному решению), а также график зависимости логарифма невязки от номера итерации.
4. Программа должна выводить время работы готовой функции (с помощью которой вычислялось точное решение), и время работы написанной функции.

Для ускорения можно использовать декоратор `@jit` из пакета `numba`.

5. Автор программы должен уметь объяснить полученные результаты на основе изученной теории.

### 3.4 Метод наискорейшего спуска и метод минимальных невязок

Напишите программу для решения линейной системы с действительной матрицей

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

методом минимальных невязок и методом наискорейшего спуска.

Требования к программе:

1. Программа должна содержать функцию, которая принимает на вход матрицу  $A$ , правую часть  $b$ , максимальное число итераций, параметр  $tol$  для ограничения на норму невязки для критерия остановки итераций, флаг, который определяет с помощью какого из 2-х методов (метода наискорейшего спуска или метода минимальных невязок) нужно вычислять шаг  $\tau$ . Программа должна возвращать приближенное решение, вычисленное нужным методом, и массив со значениями невязки на каждой итерации.
2. Программа должна создавать симметричную положительно определенную матрицу размера  $n > 100$ , правую часть и вызывать реализованную функцию.
3. Программа должна выводить число итераций, точную ошибку (вычисленную по точному решению), а также график зависимости логарифма невязки от номера итерации.
4. Программа должна выводить время работы готовой функции (с помощью которой вычислялось точное решение), и время работы написанной функции.

Для ускорения можно использовать декоратор `@jit` из пакета `numba`.

5. Автор программы должен уметь объяснить полученные результаты на основе изученной теории.