

**Параллельное программирование**  
**Лабораторная работа №2. “Эпическая”**  
**Численные методы решения СЛАУ**

**Цель работы**

Реализация прямых и итерационных численных методов решения систем линейных алгебраических уравнений в среде MatLAB. Исследование применимости, сходимости методов и времени затрачиваемого на решение.

**Задание**

1. Для проверки методов решения СЛАУ реализуемых в лабораторной работе воспользоваться следующими значениями  $A$  и  $b$ . Решение  $x$  для всех методов одинаково.

$$A = \begin{bmatrix} 1 & -0.2589 & -0.3093 \\ -0.2589 & 1 & -0.2705 \\ -0.3093 & -0.2705 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad x = \begin{bmatrix} 2.2873 \\ 2.2162 \\ 2.3068 \end{bmatrix}.$$

2. Реализовать прямые методы численного решения СЛАУ: Гаусса, Гаусса-Жордана, Крамера, через обратную матрицу, через разложение Холецкого. Каждый метод оформить в виде отдельной функции возвращающей вектор найденного решения  $x$  и булева переменную  $ok$ , а принимающей в качестве входных параметров матрицу  $A$  и вектор  $b$ .

Примечание: Переменная  $ok$  принимает ложное (`false`) значение когда метод не применим или не сходится и истинное (`true`) в противоположном случае. Если метод не применим  $x$  - нулевой вектор.

3. Аналогично предыдущему заданию реализовать итерационные методы численного решения СЛАУ: Якоби, Гаусса-Зейделя и релаксации. Для итерационных методов дополнительно передаются в функцию: вектор начальных значений  $x_0$  (принять нулевым), точность  $\varepsilon = 10^{-10}$  и максимальное количество итераций  $k_{\max} = 1000$  (все параметры задаются в основном скрипте). В выходные параметры функции добавить количество итераций  $k$  за которое было найдено решение.
4. Ознакомиться с разделами документации по работе с массивами данных разных типов (Cell Arrays): “Cell Arrays”, “Access Data in a Cell Array” и функциями `zeros` и `cell`.
5. Загрузить (`load`) из файла `lab02_data_lab.mat` массив данных  $D$  в котором находится набор матриц  $A$  и векторов  $b$ . Реализовать решение СЛАУ всеми методами для каждой пары  $A$  и  $b$  из набора данных  $D$  (через цикл `for`). Полученные решения сохранить в массив данных  $X$ , где в  $i$ -ом элементе находится матрица решений полученных всеми методами для  $i$ -ой пары  $A$  и  $b$ . Количество столбцов в матрице решений соответствует количеству методов, а количество строк длине вектора  $x$ .
6. Ознакомиться с инструкциями `tic` и `toc` и методом измерения времени выполнения определенного участка кода. Определить единицы измерения времени.

7. Для каждой пары  $A$  и  $b$  и каждого метода измерить время за которое вычисляется решение  $x$ . Результат (время в секундах) сформировать в виде матрицы  $T$ , где каждая строка хранит значения времени нахождения решения каждым методом для одной пары  $A$  и  $b$ . Если метод не применим или не сходится, то время выполнения считается равным нулю (реализовать с использованием `ok` без конструкции `if...else`). Аналогично для итерационных методов сохранять в матрице  $K$  количество итераций  $k$  за которое было найдено решение.
8. Для определения времени выполнения метода с помощью статистической выборки каждый метод требуется “обернуть” в главный скрипт в цикл с числом повторений равным  $N$  (задается в начале скрипта). В цикле замерять время выполнения каждого метода, сохраняя его в вектор на каждом  $i$ -ом повторении, после чего рассчитать математическое ожидание - среднее значение `mean` времени выполнения и уже его сохранить в матрицу  $T$ . Для отладки рекомендуется взять  $N < 50$ , а для полноценной оценки  $N \geq 10000$ .
9. Если метод уже на первом повторении выдает ошибку (`ok` принимает ложное значение) - т.е. не применим или не сходится, то дальнейшие повторения по этому методу - не выполнять, время выполнения и количество итераций (для итерационных методов) принять равным нулю.
10. В командную строку после выполнения работы основной части скрипта вывести:  $A$  и  $b$ , указать их размерность (в пояснении), таблицу `table` состоящую из векторов-столбцов решений  $x$  полученных всеми методами. В заголовке таблицы указать названия методов. Использовать `fprintf` для вывода пояснений и `disp` для вывода матриц и векторов.
11. Построить на одной канве четыре графика (через `subplot`) - линейчатые диаграммы. Слева - два графика с временем выполнения прямых методов, где верхний - все прямые методы, нижний - без матричного метода. Два графика справа (итерационные методы): верхний - время выполнения метода, а нижний - количество итераций. Во всех графиках данные сгруппировать по методу.
12. Установить текстовые наименования методов вместо чисел по оси абсцисс (`set` или через структуру, полученную через `gca`). Значение устанавливаемого параметра - `XTickLabel`. Определить специфику матриц  $A$ , что указать в легенде, которую отобразить за графиком справа.

### БОНУС:

(+2 балла в итог семестра). Добавить на каждый график с линейчатой диаграммой текстовые подписи со значениями времени и количества итераций. Построить диаграмму показывающую во сколько раз для каждой матрицы изменяется (+/-) время затрачиваемое на решение СЛАУ относительно метода Гаусса. Аналогично - производительность (величина обратная затраченному времени).