

## Лабораторное задание №2

### Описание задачи:

Реализовать метод решения системы линейных алгебраических уравнений (СЛАУ) – метод Гаусса.

### Описание метода:

Путем применения элементарных преобразований над расширенной матрицей, описывающей систему (матрица коэффициентов + столбец свободных членов):

- 1) перемена мест двух строк или двух столбцов в данной матрице;
- 2) умножение строки (или столбца) на произвольное число, отличное от нуля;
- 3) прибавление к одной строке (столбцу) другой строки (столбца), умноженной на некоторое число.

привести систему к эквивалентной системе верхнетреугольного вида (прямой ход), из которой затем последовательно, начиная с последней строки, найти все переменные (обратный ход).

$$\begin{cases} a_{11}x_1 + \dots + a_{1n}x_n = b_1 & (1) \\ a_{21}x_1 + \dots + a_{2n}x_n = b_2 & (2) \\ \dots & \\ a_{m1}x_1 + \dots + a_{mn}x_n = b_m & (m) \end{cases}$$

Прямой ход:

$$(2) \rightarrow (2) - (1) \times \left(\frac{a_{21}}{a_{11}}\right) : a'_{22}x_2 + \dots + a'_{2n}x_n = b'_2$$

$$(3) \rightarrow (3) - (1) \times \left(\frac{a_{31}}{a_{11}}\right) : a'_{32}x_2 + \dots + a'_{3n}x_n = b'_3$$

...

$$(m) \rightarrow (m) - (1) \times \left(\frac{a_{m1}}{a_{11}}\right) : a'_{m2}x_2 + \dots + a'_{mn}x_n = b'_m$$

...

$$(3) \rightarrow (3) - (2) \times \left(\frac{a'_{32}}{a'_{22}}\right) : a''_{33}x_3 + \dots + a''_{3n}x_n = b''_3$$

...

$$(m) \rightarrow (m) - (m-1) \times \left(\frac{a^{(m-2)}_{m,n-1}}{a^{(m-2)}_{m-1,n-1}}\right) : a^{(m-1)}_{mm}x_m + \dots + a^{(m-1)}_{mn}x_n = b^{(m-1)}_m$$

Обратный ход:

Из последнего ненулевого уравнения выражаем базисную переменную через небазисные и подставляем в предыдущие уравнения. Повторяя эту процедуру для всех базисных переменных, получаем решение.

Программа ***sle\_solve.cpp*** последовательно решает данную задачу, сперва для матрицы размером 3x3 для проверки корректности вычислений, сравнивая полученное решение с рассчитанными вручную значениями, затем на матрице большего размера (устанавливается через task\_size).

### **Задание:**

1. Необходимо собрать проект с исходным файлом ***sle\_solve.cpp*** и запустить собранный исполняемый файл. Оценить время работы программы и корректность ее работы. Зафиксируйте это в отчете.
2. С помощью инструментария Advisor необходимо получить метрики исполнения программы в последовательном режиме: GFLOPs, количество используемых потоков, тип векторных команд, определить хотспоты, построить roofline, сохранить snapshot.
3. Оцените потенциальный прирост производительности при введении параллелизма через **Suitability** анализ.
4. За счет добавления параллелизма ускорить выполнение программы, убедиться в корректности ее работы (анализ в Inspector), построить roofline для разных решений, оценить во сколько произошло ускорение. Для параллельной версии программы так же проанализировать эффективность выполнения через VTune. Для успешной сдачи ЛР необходимо добиться хотя бы 2-х кратного ускорения выполнения программы.
5. Подготовьте отчет (разделы ниже) и продемонстрируйте работу приложений и их характеристики преподавателю.

### **Подготовьте отчет со следующими разделами:**

1. Опишите каким образом Вы проводили оптимизацию последовательно исполняемого приложения (смена последовательности выполнения циклов, разбивка циклов на подциклы, использование SoA вместо AoS и наоборот, раскручивание циклов). Оцените влияние последовательности доступа к данным и их выравнивания в памяти на векторизацию.

2. Какие значения основных метрик производительности последовательно исполняемого приложения в Intel Advisor Вы получили (производительность в GFLOPs, пропускная способность памяти, использование векторных инструкций, roofline модель). Как изменились метрики при введении векторизации? Представьте сравнение метрик производительности последовательной работы с приложением, в котором используется «ручная» оптимизация.
3. Проведите прогнозирование ускорения выполнения программы за счет введения annotations в Intel Advisor. Осуществите введение параллелизма в последовательно исполняемую программу с использованием программной модели OpenMP, проведите анализ утилизации ресурсов процессора в Intel VTune, сравните нескольких roofline между собой (параллельное исполнение, последовательное исполнение, «ручная» оптимизация) и подтверждение выполнения прогноза ускорения в Intel Advisor. Проверьте корректность выполнения параллельной программы (Intel Inspector).