

***OpenMP задачи со звездочкой.**

1. **Axisb.c** (название которое должно быть в репозитории после написания программы) – Численно решить систему линейных уравнений

$$Ax = b,$$

Либо с помощью метода Якоби (Jacobi) (**20**), либо с помощью метода Гаусса-Зайделя (Gauss-Seidel) (**30**).

<https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4%D0%AF%D0%BA%D0%BE%D0%B1%D0%B8>.

<https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4%D0%93%D0%B0%D1%83%D1%81%D1%81%D0%B0%E2%80%94%D0%97%D0%B5%D0%B9%D0%B4%D0%B5%D0%BB%D1%8F%D1%80%D0%B5%D1%88%D0%B5%D0%BD%D0%B8%D1%8F%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B%D0%BB%D0%B8%D0%BD%D0%B5%D0%B9%D0%BD%D1%8B%D1%85%D1%83%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B9>

В обоих методах обратите внимание на условия сходимости.

2. **LeastSquares.c** – запрограммировать линейную регрессию (**20**) на данных (x_i, y_i) (создайте с помощью линейной зависимости + шума). Создайте параллельный алгоритм, чтобы найти a, b в модели $f(x, a, b) = ax + b$. Генерация сэмплов (x_i, y_i) может происходить следующим образом: $y_i = ax_i + b + noise()$. Эта задача может быть переформулирована в задачу оптимизации, то есть в задачу минимизации суммы квадратов невязок $r_i = y_i - f(x_i, a, b)$:

$$\sum_{i=1}^n (y_i - f(x_i, a, b))^2 \rightarrow \min_{a, b}$$

Gradient descending method

Detailed info: https://en.wikipedia.org/wiki/Least_squares