Оценка вычислительной сложности алгоритма в методе сопряженных градиентов

- Метод сопряженных градиентов – частный случай метода сопряженных направлений.

Он всегда сходится линейно из ∀ приближения при следующих условиях на матрицу A:
  - симметричность
  - положительная определенность

```cpp
#include "GM.hpp"

std::vector<double> Conjugate_Gradient(const CSR& A, const std::vector<double>& b, const std::vector<double>& x0, long double tolerance) {
        std::vector<double> x = x0; // x0 - начальное приближение
        std::vector<double> res = A * x - b;
        std::vector<double> d = res;
        std::vector<double> r = res; // начальная невязка

        double alpha;
        double beta;

        while (modul(res) > tolerance){
            alpha = (res * res) / (d * (A * d));
            x = x - alpha * d;
            res = A * x - b;

            beta = (res * res) / (r * r);
            d = res + beta * d;
            r = res;
        }

        return x;
}
```

Пусть n – размерность задачи

- По времени от размерности задачи
На каждой итерации мы умножаем скалярно векторы : $O(n)$, CSR матрицу A на вектор : $O(n)$ [если CSR матрицу заполнили плотно, то в худшем случае сложность возрастет до $O(n^2)$] ; арифметические операции над векторами и константами : $O(n)$

⇓

Сложность : $C_1 \cdot O(n)$, где C_1 - число операций

- По памяти от размерности задачи
Хранение векторов - $O(n)$, хранение констант α и β: $O(const)$; Хранение CSR матрицы : $O(n)$

⇓

Сложность : $C_2 \cdot O(n)$, где C_2 - кол-во хранящихся векторов