

# ЛАБОРАТОРНА РОБОТА 4

## OpenMP. Розв'язування СЛАР

### Завдання.

1. Розпаралелити метод Гаусса використовуючи `omp_set_num_threads()` та `parallel`.
2. Порівняти швидкодію програми для систем різної розмірності та при різній кількості робочих процесів.

### Директива *parallel*

Паралельна область задається за допомогою директиви *parallel*

```
#pragma omp parallel [опция[, опция]...]
```

- **Private** (список) – задає список змінних, для яких робиться локальна копія для кожної нитки.

```
omp_set_num_threads(THREADS); // Число потоков указано в define, их 4
for (i = 0; i < n; i++) {
    tmp = matrix[i][i];
    for (j = n; j >= i; j--)
        matrix[i][j] /= tmp;

#pragma omp parallel for private (j, k, tmp)
    for (j = i + 1; j < n; j++) {
        tmp = matrix[j][i];
        for (k = n; k >= i; k--)
            matrix[j][k] -= tmp * matrix[i][k];
    }
}

//Обратный ход
xx[n - 1] = matrix[n - 1][n];
for (i = n - 2; i >= 0; i--) {
    xx[i] = matrix[i][n];
#pragma omp for private (j)
```

```
for (j = i + 1; j < n; j++)  
    xx[i] -= matrix[i][j] * xx[j];  
}
```

## Примітка

Метод складається з прямого і зворотного ходів.

Прямий хід полягає в зведенні розширеної матриці системи рівнянь до трикутного вигляду шляхом застосування еквівалентних перетворень. Якщо ми подивимося на вихідний код цієї функції, то зауважимо, що обчислювальна складність алгоритму -  $O(n^3)$ .

Насправді, для кожного ( $i$ -того) з  $n$  рядків ми перебираємо всі рядки ( $j$ -ті), розташовані нижче та виконуємо додавання рядків з домноженням елемента на константу. Додавання виконується на  $O(n)$ , Тому що рядок складається з  $n + 1$  елемента.

Зворотний хід полягає в обчисленні суми  $a[i][j] * x[j]$  для кожного з  $n$  рядків, починаючи з останнього. При цьому обробляються  $j$  тільки більші за  $i$  (елементи вище головної діагоналі). Асимптотична складність такого алгоритму -  $O(n * n)$ .

Таким чином, распараллеливать доцільно тільки прямий хід.