

To: Christina Taylor
 From: Eli Case
 Date: February, 17, 2023
 Subject: CAAM 420/520 – Homework 2

- Problem 1** (a) If we use the domain decomposition shown in Figure 1, we cannot parallelize the problem because of the order dependency present when using a backward finite difference stencil. With the given domain decomposition, the threads will have to execute sequentially, as thread 1 cannot execute before thread 0, thread 2 cannot execute before thread 1, etc. Thus, we will be computing the problem serially with the additional overhead of setting up the parallelization, meaning we would want to change the domain decomposition to be able to execute the problem in parallel, while preserving the order-dependency of the problem.
- (b) We can begin with the formula for the time to process the spin up region, T_{full} .

$$T_{full} = N_w(T_s + n_b T_p) \quad (1)$$

To determine when the synchronization time contributes more to the total spin-up time than the computation time, we can construct the following inequality.

$$N_w T_s > N_w n_b T_p \quad (2)$$

Reducing this inequality, we have the following.

$$\begin{aligned} T_s &> n_b T_p \\ n_b &< \frac{T_s}{T_p} \end{aligned}$$

Thus, from above, we have that synchronization contributes more time to T_{full} when the number of elements per block, n_b , is less than the ratio of synchronization time to computation time per node, $\frac{T_s}{T_p}$. While smaller blocks provide better parallelization by reducing the spin-up and spin-down time, the cost of synchronization limits how small the block sizes can be, since blocks that are too small cause more spin-up time to be spent on synchronization, rather than computation.

- Problem 2** (b) The overall number of blocks requiring computation is $N_x N_y$. We can compute the number of blocks in the spin-up and spin-down regions, `blocks_spin` as follows.

$$\text{blocks_spin} = 2(N_T - 1)! \quad (3)$$

Similarly, the number of blocks fully parallel region, `block_parallel`, is as shown below.

$$\text{block_parallel} = N_x N_y - 2(N_T - 1)! \quad (4)$$

If we assume every block takes the same amount of time to process and synchronize, we can write the ratio of the time spend in the spin-up and spin down region to the fully parallel region as follows.

$$\frac{\text{blocks_spin}}{\text{block_parallel}} = \frac{2(N_T - 1)!}{N_x N_y - 2(N_T - 1)!} \quad (5)$$

- (c) Below in Figure 1, we have the computed strongly scaled parallel speed-up plotted across ratios from 1 to 3 for blocks to threads.

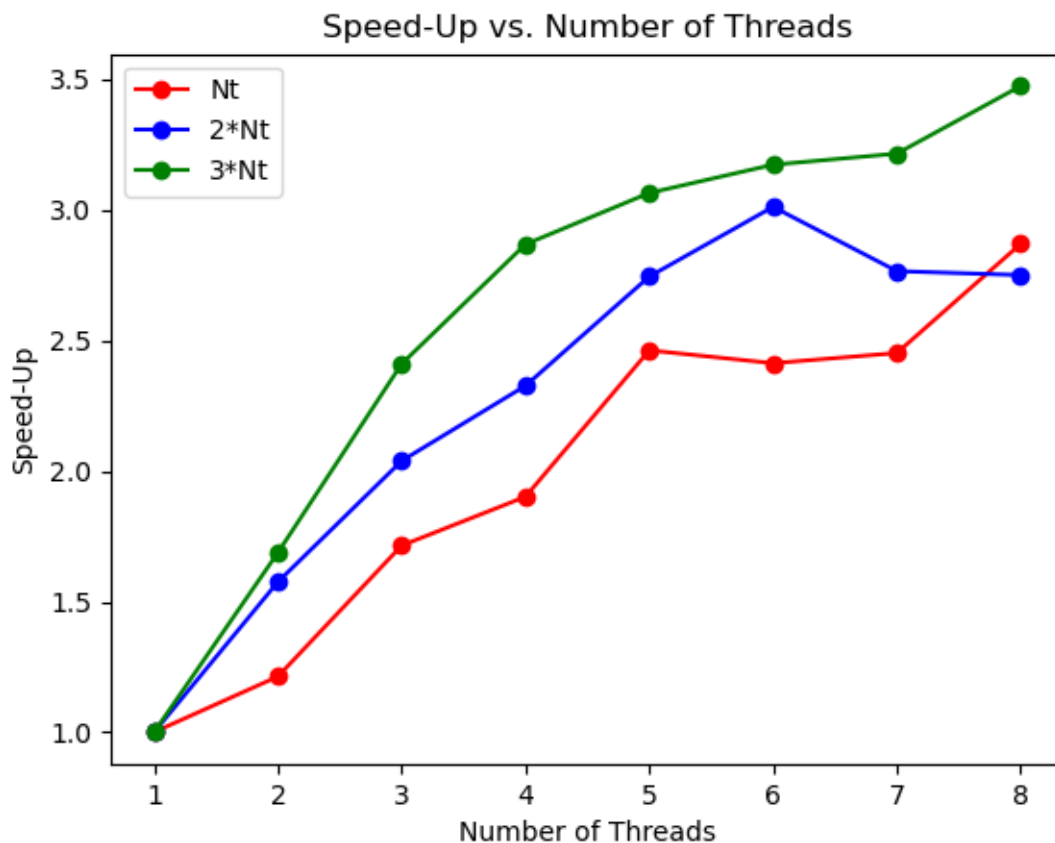


Figure 1. Strong Scaling: Speed-up vs. Number of Threads