To: Christina Taylor
From: Anwar Khaddaj
Date: February 15, 2023
Subject: CAAM 420/520 – Homework 2
I collaborated with Arshia Singhal and Jonathan Cangelosi on this homework.

Problem 1 (a) We understood 2 versions of this problem:

i. If each grey block is one block instead of several small blocks, then we would have the issue of order dependency in some parts of the grey blocks especially that this is being parallelized so all the grey blocks are processed at the same time. Thus, some parts of grey block 1 depends on grey block 0 (3-point stencil of backward FD), and so on.

ii. If each grey block is made up of several blocks like the ones in lecture notes then, the blocks seem to have the same size. Thus, the effects of the spin-up and spin-down phases are high so the system might have a small (relatively small) fully parallelized portion and it spends more time working with the phases where not all threads are active. Therefore, we should have smaller blocks (smaller in size) especially for the spin-up and spin-down so that we minimize their effects. Another issue is that depending on the stencil, this might not be achievable in higher dimensions.

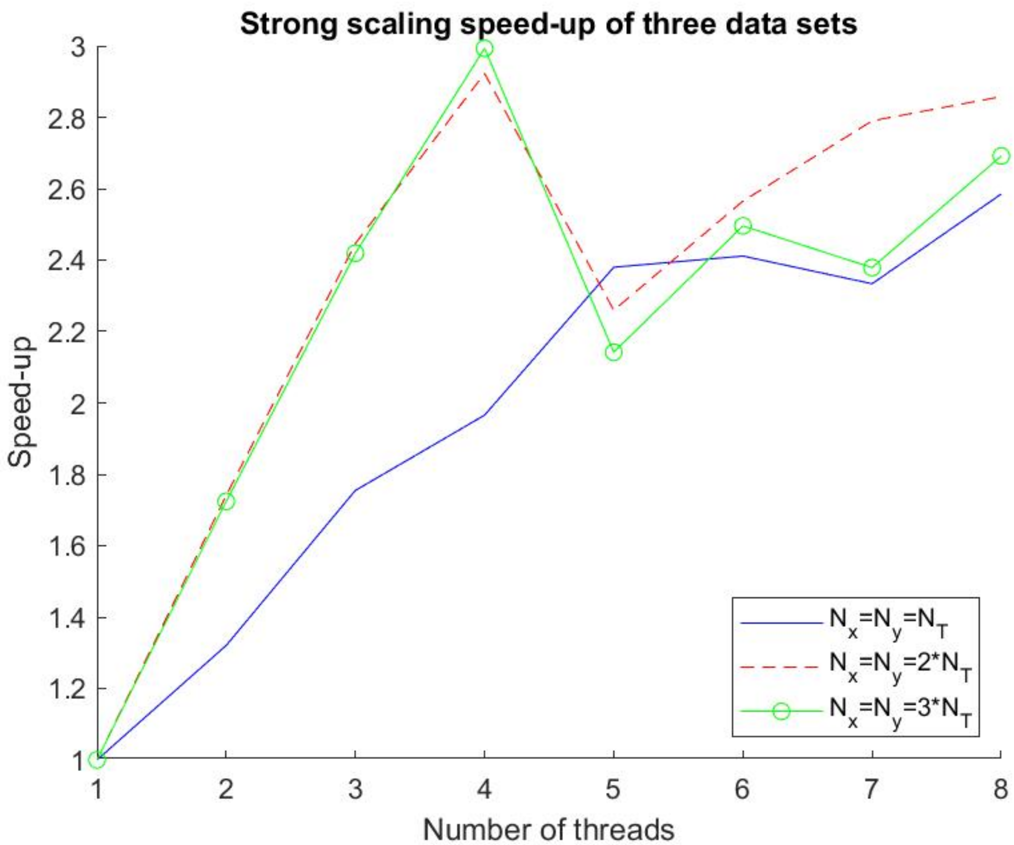
(b) As we increase the number of blocks (i.e. have more smaller size blocks), we decrease n_b thus, $Ts > n_b T_p$ and T_s contributes more time. Thus, if we have smaller size blocks, although we achieve better parallelization by minimizing the spin-up and spin-down time, we encounter large costs of synchronization so this impacts our choice of block sizes by discouraging us to go with the smallest possible block size for example. However, if we have larger size blocks, we achieve better synchronization (in terms of cost) at the cost of parallelization. A suggestion would be to have variable block sizes (larger sizes for the fully-spun phase and smaller ones for the spin-up and spin-down phases). If the block sizes ought to be constant across the entire system, then we can find a middle-ground size that satisfies both purposes.

Problem 2 for CAAM420.

Problem 3 (a) Submitted as code.

(b) Total number of blocks: $N_x * N_y$.
Number of blocks in the spin-up phase$= 1 + ... + (N_T - 1) = \frac{N_T*(N_T-1)}{2}$. Similarly for the spin-down phase.
Thus, number of blocks in the spin-up and spin-down$=N_T * (N_T - 1)$. Therefore, the fraction of the program spent in the spin-up and spin-down is $\frac{N_T*(N_T-1)}{N_x*N_y}$ while the fraction of the program spent in the fully-parallelized region is $1 - \frac{N_T*(N_T-1)}{N_x*N_y}$.

(c) Submitted as MATLAB code. The plot is the following:


Strong scaling speed-up of three data sets
Speed-up
Number of threads
1
1.2
1.4
1.6
1.8
2
2.2
2.4
2.6
2.8
3
1
2
3
4
5
6
7
8
N_x=N_y=N_T
N_x=N_y=2*N_T
N_x=N_y=3*N_T