1 Matrix Transposition

The sequential transposition of a matrix $A$ of size $N \times N$ works as follows.

$$
\text{for } i \leftarrow 1 \text{ to } N \text{ do} \\
\quad \text{for } j \leftarrow i \text{ to } N \text{ do} \\
\quad \quad \text{tmp} \leftarrow A(i,j); \ A(i,j) \leftarrow A(j,i); \ A(j,i) \leftarrow \text{tmp}
$$

a) In order to compute $A^T$ in parallel, one can choose between function, data, and competitive parallelism. Depending on the underlying hardware (shared or distributed memory) these approaches have different strong and weak aspects. Give an implementation idea for each approach and discuss your results.

b) Compute for a data parallel approach with column-wise data decomposition the speed-up depending on $N$ and $p$ (the number of processes) only! Exchanging two local elements $A(i,j)$ and $A(j,i)$ takes $T_{EX}$ time, transmitting one element from process $p_i$ to process $p_j$ takes $T_{COM} = 1.5 \cdot T_{EX}$ time. For simplification you can further assume, that one process transmits all elements of its domain (i.e. this includes also the redundant transmission from $p_i$ to $p_i$ itself). What can you observe for the parallel efficiency?

2 Dependency Analysis

For the successful parallel execution of processes a dependency analysis might become necessary. Here, Bernstein’s conditions help to identify instruction-level parallelism as to be observed in loops, i.e. each iteration of a loop must be independent from all other iterations, thus no read–write dependencies exist. Examine the following code fragment and decide whether instruction-level parallelism might occur or not.

$$
\text{for } i \leftarrow 1 \text{ to } N \text{ do} \\
\quad a(i) \leftarrow 2 \cdot a(i+m) + b(i)
$$

For which values of $m$ ($-1 \leq m \leq 2$) does parallel execution of the loop iterations become possible?