Write in CUDA a matrix-vector multiplication routine \((y = A x)\) where the matrix \(A\) corresponds to the five-point stencil approximating the Laplacian operator in 2D, as discussed in Lecture #9 (see slides #17 and 19).

Try to optimize the routine and report the performance for various matrix sizes. Justify correctness. What would be a theoretical performance peak and how close to it is your implementation?

Bonus: Suggest further optimizations for computing \(A^2 x\).