Flop counting

In this tutorial you will implement LU-decomposition for *banded* matrices. You will compare the FLOP count to that of LU decomposition of matrices without this special structure. For simplicity, we will not consider pivoting (row permutations).
Consider a *banded* matrix:

$$A \in \mathbb{R}^{n\times n} \text{ with } A_{ij} = 0 \text{ for } j > i + b \text{ or } i > j + b$$

We say that A has *band width* b. On each row it has at most $2b + 1$ nonzero elements.

- Write down a 4×4 matrix A with band width 1. Perform LU-decomposition without pivoting by hand. Which steps can you skip because of the zeros in the matrix?
- Write a pseudo-code for the LU decomposition of a banded matrix with band width b. Make sure it does not perform unnecessary flops.
- Implement your pseudo-code and test it on a few banded matrices (like the one from part (**a**) or random banded matrices you create. Check the result against the LU-decomposition function in the Solving systems of (non)linear equations/Codes folder on Blackboard.
- Count the flops in your pseudo-code, or find an upper bound for it that is correct to leading order in n and b.
- Run your function on matrices with bandwidth $b = 3$ and $n = 8, 16, \ldots, 512$ and store the wall times. Then run it for matrices of size $n = 100$ and $b = 2, 3, \ldots, 50$ and store the wall times. Plot the wall times on a logarithmic scale and compare the leading order in n and b to your flop count.

Discussion
How much faster is the decomposition for banded matrices as compared to general, dense matrices? Is there a “quick and dirty” way to estimate the number of flops required (meaning without first carefully counting all of them)?