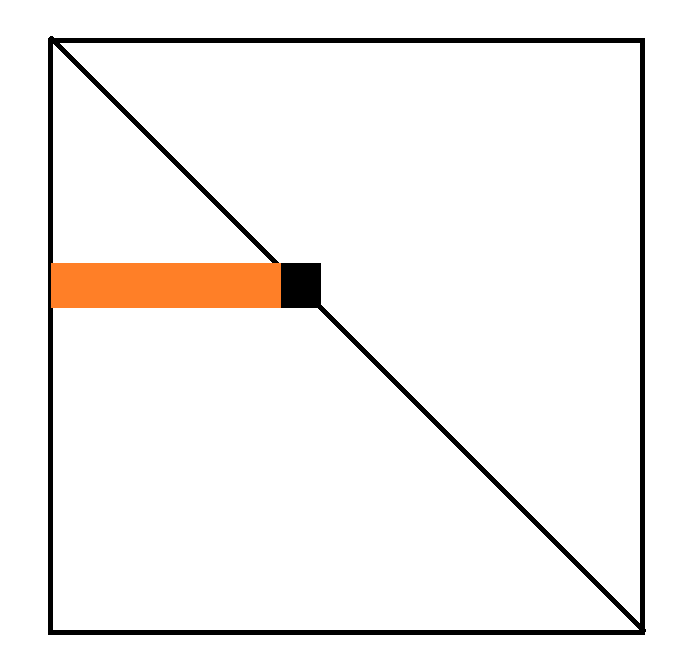
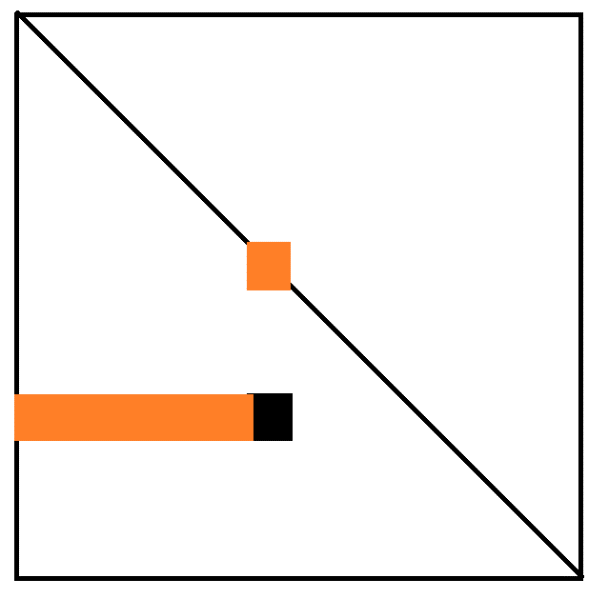
Choleksy Algorithm Notes/Inquiry

1. To optimize the algorithm, each CPU will be connected to a GPU.
2. Key observation:
   1. Diagonal Calculations:
      1. To calculate the diagonal (except the base case A11), the factorization for the row that belongs to the diagonal must be computed first before the diagonal can be computed.
   2. Not Diagonal Calculations:
      1. When calculating elements not on the diagonal, one of the values it requires is the computed diagonal value directly on top of the element.
      2. It also requires all of the previous computed row values.

In this image, in order to calculate the Diagonal value labeled in black, we need to have the previously computed values labeled in orange.

In this image, to calculate an element not on the diagonal, it requires the diagonal value directly above it to be calculated and the values of the row to be calculated.

Now the question is how do we optimize this. A typical algorithm, is to do this column by column. We go through all columns one at a time. As an example, the first column is only depending on the diagonal value L11. Thus, in a sense L11 is a base case and the column with that value relies on that value to become base cases for the subsequent columns.

Questions for Optimization:

1. What exactly are GPU kernels?
2. Any tutorials for GPU kernels on Julia?
3. The paper specified here uses each kernel for each element in each column:
   1. Is that desirable?
   2. What is the typical strategy
   3. <https://lume.ufrgs.br/bitstream/handle/10183/151001/001009773.pdf>
4. In the original paper, I am having a very hard time understanding the idea of PLASMA and the MAGMA BLAS libraries.
5. The basic question now is how to use Kernels
6. A non-tile algorithm

Advantage of Julia