Prof. Dr. Peter Bastian, Santiago Ospina De Los Ríos
IWR, Universität Heidelberg

Submission date: 2023-11-13

Exercise 1 *LU decomposition and Computational Intensity*

In this exercise we want to analyze the computational intensity of the LU decomposition for dense matrices and band matrices. Let $A \in \mathrm{R}^{n\times n}$ be a regular matrix. In the following you can assume that the LU decomposition can be done without permuting the matrix.

1. What is the computational intensity of the LU decomposition[2] of A if you use standard Gauß elimination without blocking?
2. Now we want to analyze the LU decomposition with blocking[3], similar to the matrix multiplication example. Make sure you know what is going on and that you could explain the idea with a quick sketch.
3. What is the computational intensity for the blocked version?
4. As a last step we want to consider band matrices[4] with band width k. Calculate the computational intensity for this case.

(5 Points)

Exercise 2 *Matrix Transpose*

In the `hasc-code` repository you can find the matrix transposition program presented in the lecture.

1. The file `transpose.cc` contains matrix transposition with four different implementations:
   - Consecutive write – Strided read
   - Strided write – Consecutive read
   - Blocked consecutive write – Strided read
   - Blocked strided write – Consecutive read

   Make a small parameter study on the block and matrix sizes for these different strategies. Measure the runtime and report your results in a table or plot. What are the best block sizes for your computer and how they relate to your hardware?
2. Have a look at the vectorised versions of matrix transposition and make sure to understand what's going[5]:
   - `transpose_avx.cc`
     - AVX transpose with 4×4 – Vector Class Library
     - AVX transpose with 4×4 – Intel Intrinsics
     - AVX transpose with 4×4 – Intel Intrinsics – Non-temporal stores
   - `transpose_neon.cc`
     - NEON transpose with 4×4 – Consecutive write
     - NEON transpose with 4×4 – Consecutive write – TLB Blocking

[2] `https://en.wikipedia.org/wiki/LU_decomposition`
[3] Don't confuse this with block LU decomposition
[4] `https://en.wikipedia.org/wiki/Band_matrix`
[5] Specially with all those intrinsics and the `blend4` function

For one of these architectures, make a small parameter study on the matrix sizes and reason about your results with respect to your hardware. Additionally, what is memory alignment and what happens if the memory is not properly aligned in this experiment?

3. **Bonus:** Try to implement the matrix transpositions in a faster way. You can try whatever you want. Explain your reasoning.

4. **Bonus:** Compare the implementation of the library MKL or Eigen against other implementations in this exercise sheet.

   - `transpose_library.cc`
     - Library transpose – Eigen
     - Library transpose – MKL

(5 Points)