Institut of Numerical Mathematics
Dr. Andreas F. Borchert and Dr. Michael C. Lehn
Constantin Greif

11 January 2019
Quiz 5

High Performance Computing I (WS 2018/2019)

Deadline: 18 January 2019, 2pm

For a regular $m \times n$ matrix A, the LU-factorization has the form

$$PA = LU$$

where P is a permutation matrix, L is lower triangular with unit diagonal elements (lower trapezoidal if $m > n$), and U is upper triangular (upper trapezoidal if $m < n$). An implementation can overwrite A compactly with L and U and we therefore denote such an operation by

$$A \leftarrow P^T(L \backslash U) \quad \text{(lu-variant)}.$$

Assuming an algorithm *lu_unblk* is known we can define algorithm *lu_blk* as follows:

- if $\min\{m,n\} < \text{bs}_{\min} := 2^k$ (for fixed $k \geq 0$)

$$A \;\leftarrow\; P\,(L \backslash U) \quad \text{lu\_unblk}$$

- else choose $\text{bs} = \max\left\{2^\ell \;:\; 2^\ell < \min\{m,n\}\right\}$ and proceed for

$$A = \left(\begin{array}{c|c} A_{0,0} & A_{0,\text{bs}} \\ \hline A_{\text{bs},0} & A_{\text{bs},\text{bs}} \end{array}\right)$$

as follows:

$$\begin{array}{rcll}
\left(\begin{array}{c} A_{0,0} \\ \hline A_{\text{bs},0} \end{array}\right) & \leftarrow & P^T \left(\begin{array}{c} L_{0,0} \backslash U_{0,0} \\ \hline L_{\text{bs},0} \end{array}\right) & \text{(lu_blk)} \\
\left(\begin{array}{c} A_{0,\text{bs}} \\ \hline A_{\text{bs},\text{bs}} \end{array}\right) & \leftarrow & P \left(\begin{array}{c} A_{0,\text{bs}} \\ \hline A_{\text{bs},\text{bs}} \end{array}\right) & \text{(swap)} \\
A_{0,\text{bs}} & \leftarrow & L_{0,0}^{-1} A_{0,\text{bs}} & \text{(triangular solver)} \\
A_{\text{bs},\text{bs}} & \leftarrow & A_{\text{bs},\text{bs}} - A_{\text{bs},0} A_{0,\text{bs}} & \text{(matrix product)} \\
A_{\text{bs},\text{bs}} & \leftarrow & \tilde{P}^T \left(L_{\text{bs},\text{bs}} \backslash U_{\text{bs},\text{bs}}\right) & \text{(lu_blk)} \\
A_{\text{bs},0} & \leftarrow & \tilde{P} \;\; A_{\text{bs},0} & \text{(swap)}
\end{array}$$

Write a function *lu_blk_var2* that implements this algorithm. You can use the test cases from session 21. But note that your implementation must not call any MKL functions. Only use the corresponding ulmBLAS functions. The file

```
/home/numerik/pub/hpc/ws18/lu_quiz05_stub.hpp
```

(accessible on *theon*) contains a stub for your code.

Please submit your code "lu.hpp" as follows:

```
theon$ submit hpc quiz05 lu.hpp
```