Numerical High Performance Algorithms
WS 2018

Assignment 1: High Performance LU Factorization

General remarks:

- The main objective of all your codes is to maximize performance (minimize runtime) while providing accurate results.
- You have to write your own code for factorizing a matrix A in Octave or Matlab. For solving the resulting triangular systems you may use the \-operator.
- **Hardware target system**: You can use your own hardware system. Determine and report the efficiency which you achieved in your experimental evaluations. If you do not have any hardware available for doing this homework, please contact me.
- **Evaluation of accuracy**:
  - For the computed solution of a linear system:
    - Compute the *relative forward error*, i.e. $\frac{\|x-x'\|_1}{\|x'\|_1}$, where x is the computed solution and x' is the known exact solution. Determine b such that the exact solution x' is a vector of all ones: $x' = [1\ 1\ 1 \ldots 1]^T$.
    - Compute the *relative residual norm* of the computed solution vector x: $\|r\|_1 := \frac{\|Ax-b\|_1}{\|b\|_1}$. *Do not count the work and the time needed for the accuracy evaluation in your performance evaluation*!
  - For the LU factorization:
    - Compute the *relative factorization error* $\|PA - LU\|_1 / \|A\|_1$, where LU is the computed factorization of the given matrix A.
- **Evaluation of runtime performance**: Measure the runtime of your routine. Make sure that you measure ONLY the time needed for the operation under consideration (without main routine, input data generation, accuracy evaluation, etc.). Explain carefully in your report how you measured the runtime.
- Summarize the results of your experimental evaluations graphically for problem sizes n=100:100:N, where N as large as possible in reasonable time (N depends on your code optimization and on the machine you are using!).

Blocked Right-Looking LU Factorization with Partial Pivoting

Write routines which perform a *blocked* and an *unblocked* right-looking LU factorization with partial pivoting of a given matrix $A \in \mathbb{R}^{nxn}$. Evaluate accuracy, runtime performance and efficiency of your codes. In particular, compare runtimes and efficiency of blocked and unblocked LU factorization. For the experimental evaluations, use randomly generated (nonsingular) A.

Hint: For detailed information about blocked LU factorization, please refer to the paper [Cole], to Section 3 of the paper [Toledo] or to http://www.netlib.org/utk/papers/outofcore/node3.html

Due date (*submission of report and code in Moodle*): **22.10.2018, 18:00**