Technische Universität München
Institut für Informatik
Prof. Dr. Thomas Huckle
Dipl.-Inf. Christoph Riesinger

WiSe 2015/16

# Parallel Numerics

Exercise 5: Parallel LU Decomposition & Collective Operations

## 1 LU Decomposition

To calculate the solution of a linear equation system

$$Ax = b$$

with a non singular matrix $A = (a_{i,j}) \in \mathbb{R}^{n \times n}$ and $b \in \mathbb{R}^n$ the LU decomposition algorithm can be applied. The algorithm decomposes the matrix $A$ in a lower triangular matrix $L$ and an upper triangular matrix $U$: $A = LU$ (cp. lecture notes).

A recursive procedure for calculating the upper triangular matrix $U =: A^{(n)}$ from the matrix $A$ can be given as follows:

1. Define $A^{(1)} = (a_{i,j}^{(1)}) := A$
2. Calculate for $k = 1, \ldots, n-1$ the values $l_{i,k}$, $i = k+1, \ldots, n$ and the matrices $A^{(k+1)} = (a_{i,j}^{(k+1)})$ iteratively with

$$l_{i,k} := a_{i,k}^{(k)} / a_{k,k}^{(k)}$$

$$a_{i,j}^{(k+1)} := \begin{cases} a_{i,j}^{(k)} - l_{i,k} a_{k,j}^{(k)} & \text{for } i \in \{k+1, \ldots, n\},\ j \in \{k, \ldots, n\} \\ a_{i,j}^{(k)} & \text{otherwise} \end{cases}$$

A serial implementation of this iterative algorithm could have the following form ("$kij$-loop-arrangement"):

```
for k = 1 to n - 1
    i = k + 1 to n
        l_{i,k} := a_{i,k} / a_{k,k}
        for j = k to n
            a_{i,j} := a_{i,j} - l_{i,k} a_{k,j}
```

a) Define the term *in situ* storage scheme.

b) Define the term *Forward Elimination* and the term *Backward Elimination* mathematically.

c) Given is

$$M = \begin{pmatrix} 2 & 2 & 1 \\ 4 & 2 & 3 \\ 2 & 2 & 2 \end{pmatrix}$$

Compute the LU decomposition $M = L \cdot U$.

d) Compute the LU decomposition using a pivot search (move the entry with the largest absolute value to the pivot position).

# 2 Collective Operations

a) Define the MPI term *collective operation*.

b) Are there any non-blocking collective operations?

c) Give examples for this three different types of collective operations:

| | |
|---|---|
| Collective Computations | ... |
| Data movement | ... |
| Synchronisation | ... |

d) The `MPI_Reduce` command supports lots of predefined computations. List and explain them. What is the difference to `MPI_Allreduce`?

# 3 Parallel LU Decomposition

a) Give the data dependency graph for the first two equations of the backward elimination. Is there a parallel algorithm to compute the backward elimination?

b) Divide the matrix into blocks of rows. Scatter the blocks among the nodes of a parallel machine. Give a parallel algorithm. What collective operations are used? How many nodes are idle troughout the computation?

c) Divide the matrix into blocks of columns. Scatter the blocks among the nodes of a parallel machine. Give a parallel algorithm. What collective operations are used? How many nodes are idle throughout the computation?

d) Assign the columns of the matrix alternating to the different machines. Reformulate a parallel algorithm in words (or pseudocode). What about the load balancing and load difference, respectively?