

Parallel Numerics

Exercise 5: Parallel Gaussian Elimination & Collective Operations

1) Gaussian Elimination

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 Gauss Elimination 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 $k = 1, \dots, n - 1$ the values $l_{i,k}$, $i = k + 1, \dots, n$ and the matrices $A^{(k+1)} = (a_{i,j}^{(k+1)})$ iteratively with

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

A serial implementation of this iterative algorithm could have the following form (“*kij*-loop-arrangement”):

```
for k = 1 to n - 1
  for i = k + 1 to n
    li,k := ai,k / ak,k
    for j = k to n
      ai,j := ai,j - li,k ak,j
```

- i) Define the term *in situ* storage scheme.

- ii) Define the term *Forward Elimination* and the term *Backward Elimination* mathematically.
- iii) Given is

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

Compute the Gaussian Elimination $M = L \cdot U$.

- iv) Compute the Gaussian Elimination using a pivot search (move the entry with the largest absolute value to the pivot position).

2) Collective Operations

- i) Define the MPI term *collective operation*.
- ii) Are there any non-blocking collective operations?
- iii) Give examples for this three different types of collective operations:

Collective Computations	...
Data movement	...
Synchronisation	...

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

3) Parallel Gaussian Elimination

- i) Give the data dependency graph for the first two equations of the backward elimination. Is there a parallel algorithm to compute the backward elimination?
- ii) 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 throughout the computation?
- iii) 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?
- iv) 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?