PaStiX: Distributed Interface

Supervisors: Pierre Ramet
and Mathieu Faverge

Alycia Lisito

Summary

01

PaStiX

PaStiX = Parallel Sparse Linear Algebra Solver

- Linear Algebra Solver
  - Solves $Ax = b$
- Sparse
  - Matrix with a lot of zero elements
- Parallel
  - Several schedulers:
    - Sequential
    - Static
    - Dynamic
    - StarPU
    - Parsec
  - MPI

Inria

How does PaStiX work ?

4 steps

- Analyse
  - Ordering: Scotch, Metis
    - Computes the permutation P
  - Symbolic Factorisation
    - Computes the graph of A
  - Blend
    - Computes the blocks Partition of A
- Numerical Factorisation
  - Computes PAP^T
  - Stores PAP^T in blocks
- Solve
  - Solves $PAP^T Px = Pb$
- Refinement
  - Refines the solution x

Analyse

- Computes the permutation P
- Computes the graph of A
- Computes the blocks Partition of A

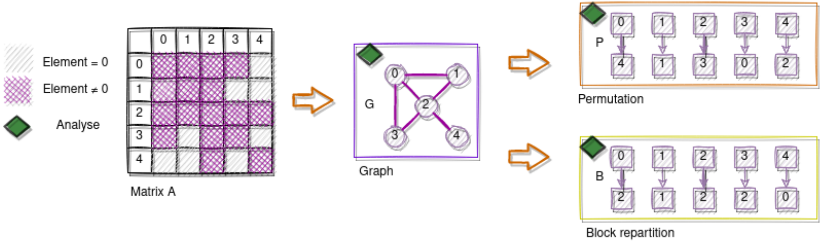

Figure: Example of the analyse step for a matrix A

Factorisation

- Computes the permutation PAP^T
- Stores PAP^T in the blocks

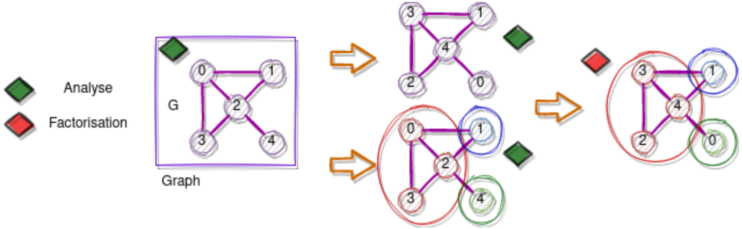

Figure: Example of the factorisation step for a matrix A: the graphs

Inria

Factorisation

- Computes the permutation PAP^T
- Stores PAP^T in the blocks

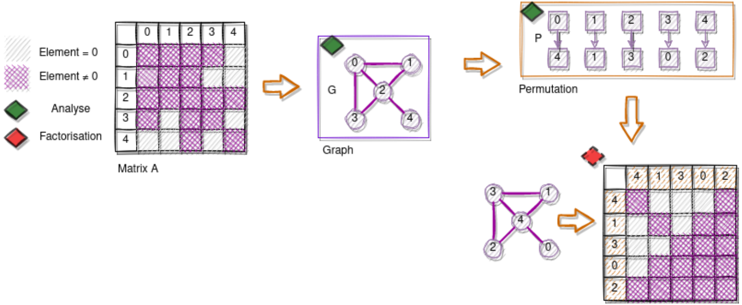

Figure: Example of the factorisation step for a matrix A: permutation

Factorisation

- Computes the permutation PAP^T
- Stores PAP^T in the blocks

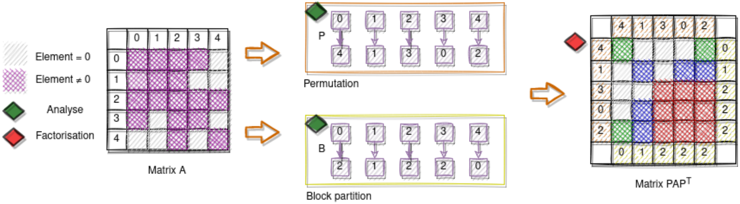

Figure: Example of the factorisation step for a matrix A

02

Degree of Freedom

Single DoF: temperature

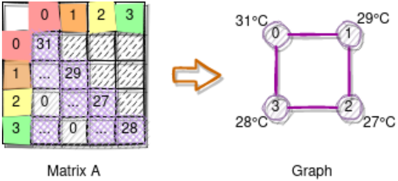

Figure: Graph of a matrix A with Single DoF

Multiple Constant DoF: temperature and pressure

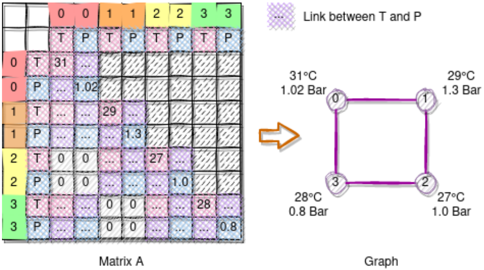

Figure: Graph of a matrix A with Multiple Constant DoF

Multiple Variadic DoF: temperature, pressure and volume

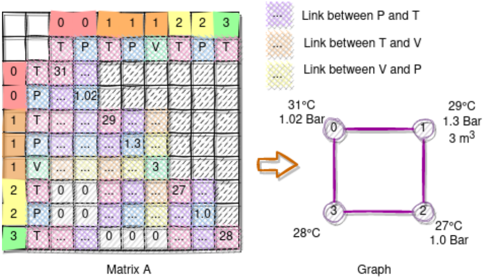

Figure: Graph of a matrix A with Multiple Variadic DoF

03

From *CSC* to *Block CSC*

CSC format

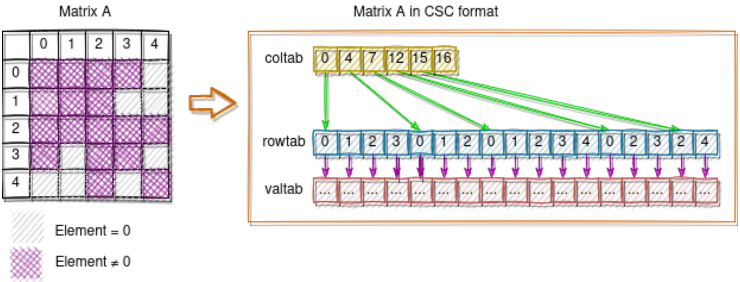

Figure: A in CSC format

BCSC format

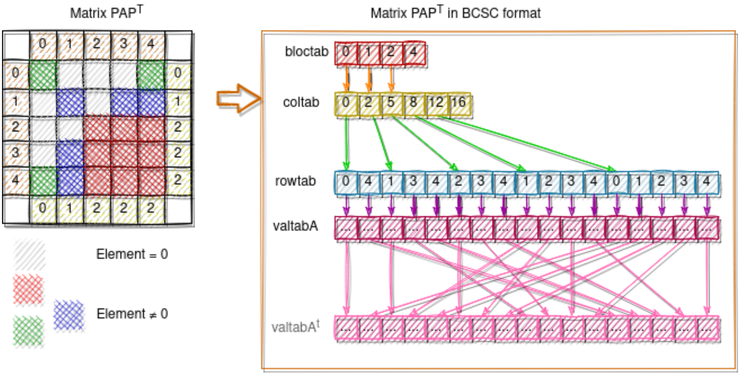

Figure: PAP^T in *BCSC* format

Shared memory implementation

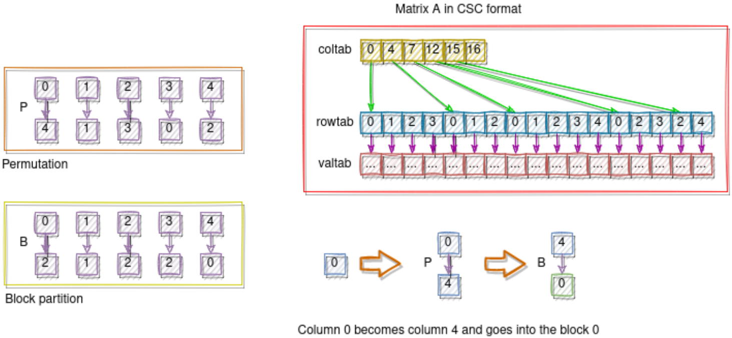

Figure: *CSC* to *BCSC* in shared memory

04

MPI implementation

Matrix distributed in memory

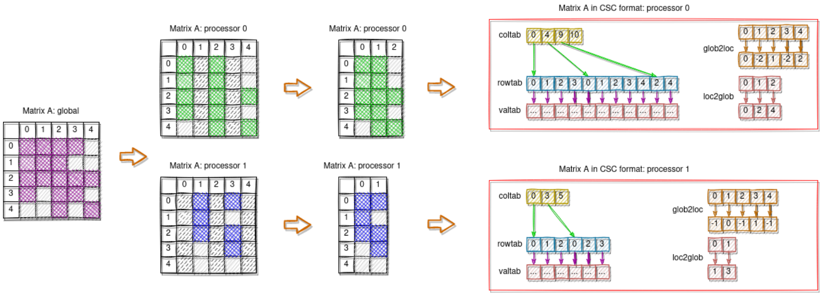

Figure: Matrix Partition in distributed memory

Block distributed in memory

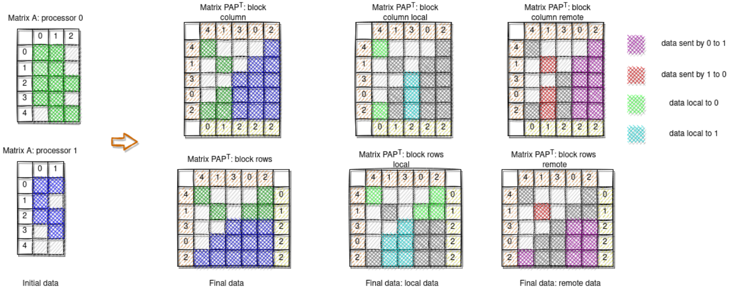

Figure: Block Partition in distributed memory

Distributed memory implementation

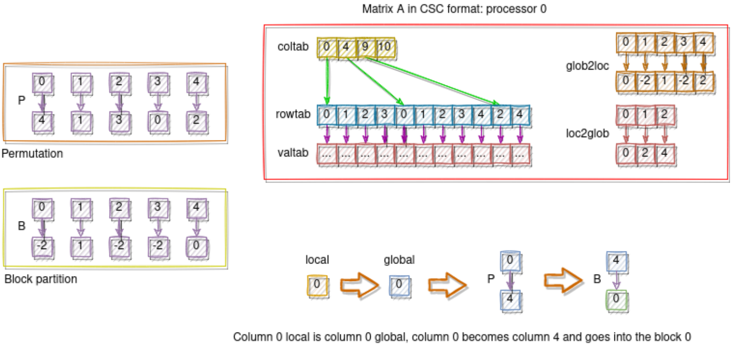

Figure: CSC to BCSC in distributed memory: processor 0

Distributed memory implementation

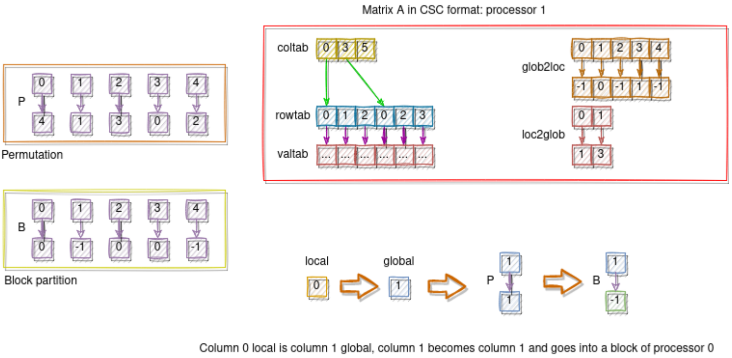

Figure: CSC to BCSC in distributed memory: processor 1

Exchanging the data

Data buffers

- Sending indexes: $nbr_{proc} - 1$
- Sending values: $nbr_{proc} - 1$
- Receiving indexes: 1
- Receiving values: 1

Count the data

- If *DoF* equals to 1: 1 value per 2 indexes.
- If *DoF constant* equal to d: d per 2 indexes.
- If *DoF variadic*: number of values per 2 indexes depends on the *DoF* of the indexes.

05

Conclusion

Next steps

- Reduce the amount of data exchanged
- Distributed Solve and Refinement:
  - Exchange the data for the vector
  - TRSM part of the solve
- Variadic degree of freedom