PaStiX: Distributed Interface

Supervisors: Pierre Ramet and Mathieu Faverge

Alycia Lisito
Summary

01. PaStiX
02. Degree of Freedom
03. From CSC to Block CSC
04. MPI implementation
05. Conclusion
PaStiX
What is 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
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 $PAP^TPx = Pb$

• Refinement
  > Refines the solution $x$
The Analyse step

**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 $A$
The Factorisation step

Factorisation

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

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

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

**Figure:** Example of the factorisation step for a matrix $A$: permutation
The Factorisation step

Factorisation

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

**Figure**: Example of the factorisation step for a matrix $A$
Degree of Freedom
Single 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 Degree of Freedom

Multiple Variadic DoF: temperature, pressure and volume

Figure: Graph of a matrix A with Multiple Variadic DoF
From CSC to Block CSC
CSC format

Figure: A in CSC format
BCSC format

**Figure:** $PAP^T$ in BCSC format
From $A_{CSC}$ to $PAP^T_{BCSC}$

Shared memory implementation

**Figure:** CSC to BCSC in shared memory
MPI implementation
Matrix distributed

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: \( n_{\text{br}}_{\text{proc}} - 1 \)
- Sending values: \( n_{\text{br}}_{\text{proc}} - 1 \)
- Receiving indexes: 1
- Receiving values: 1

Count the data

- If \( \text{DoF} \) equals to 1: 1 value per 2 indexes.
- If \( \text{DoF constant} \) equal to \( d \): \( d \) per 2 indexes.
- If \( \text{DoF variadic} \): number of values per 2 indexes depends on the \( \text{DoF} \) of the indexes.
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