Pentadsolver, a scalable batch-pentadiagonal solver library for ADI applications

Gábor Dániel Balogh, István Reguly

Pázmány Péter Catholic University
Faculty of Information Tecnology and Bionics

GPU Day 05. 16. 2023.

fides et ratio

Preview

1. What is a pentadiagonal system and where to find them?
2. Pentadsolver library implementation considerations
3. Performance

Previously

- We worked on a distributed batch-tridiagonal solver library
- In a CFD context, pentadiagonal systems have the potential to offer more flexibility and more accuracy.
- This work is a follow up on that library.

Pentadiagonal equations

- A traditional system of linear equations, with non-zeros on the main diagonal and on 2 lower and upper subdiagonal
- Gauss-Jordan elimination special case – Thomas algorithm
- Inherently sequential algorithm . . .
    - O(N) complexity

$$
\left[\begin{array}{ccccccccccccccccc}
d_0 & u_0 & w_0 \\
l_1 & d_1 & u_1 & w_1 \\
s_2 & l_2 & d_2 & u_2 & w_2 \\
 & s_3 & l_3 & d_3 & u_3 & w_3 \\
 & & s_4 & l_4 & d_4 & u_4 & w_4 \\
 & & & s_5 & l_5 & d_5 & u_5 & w_5 \\
\hline
 & & & & s_6 & l_6 & d_6 & u_6 & w_6 \\
 & & & & & s_7 & l_7 & d_7 & u_7 & w_7 \\
 & & & & & & s_8 & l_8 & d_8 & u_8 & w_8 \\
 & & & & & & & s_9 & l_9 & d_9 & u_9 & w_9 \\
 & & & & & & & & s_{10} & l_{10} & d_{10} & u_{10} & w_{10} \\
 & & & & & & & & & s_{11} & l_{11} & d_{11} & u_{11} & w_{11} \\
\hline
 & & & & & & & & & & s_{12} & l_{12} & d_{12} & u_{12} & w_{12} \\
 & & & & & & & & & & & s_{13} & l_{13} & d_{13} & u_{13} & w_{13} \\
 & & & & & & & & & & & & s_{14} & l_{14} & d_{14} & u_{14} & w_{14} \\
 & & & & & & & & & & & & & s_{15} & l_{15} & d_{15} & u_{15} \\
 & & & & & & & & & & & & & & s_{16} & l_{16} & d_{16}
\end{array}\right]
\left[\begin{array}{c}
x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ \hline x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ \hline x_{12} \\ x_{13} \\ x_{14} \\ x_{15} \\ x_{16}
\end{array}\right]
=
\left[\begin{array}{c}
b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ \hline b_6 \\ b_7 \\ b_8 \\ b_9 \\ b_{10} \\ b_{11} \\ \hline b_{12} \\ b_{13} \\ b_{14} \\ b_{15} \\ b_{16}
\end{array}\right]
$$

Algorithm `thomas`(s, l, d, u, w, b)

for $i = 0, 1, 2, ..., N - 1$ **do**
    eliminate lower subdiagonals s, l
end for
for $i = N - 2, ..., 1, 0$ **do**
    eliminate upper subdiagonals u, w
end for
return b

Solving pentadiagonal equations in parallel

- Parallel Cyclic Reduction for Pentadiagonal systems - PCR-Penta
  - Eliminate $x_{i\pm1}$, $x_{i\pm2}$,$x_{i\pm4}$ ... from equation i
  - O(N logN) steps, but only logN sequential steps
- PCR with blocks
  - Create a tridiagonal like structure from 2×2 blocks
  - Eliminate the variables of $i \pm 1$, $i \pm 2$,$i \pm 4$ ... block from block i
- O(N logN) steps, but only logN sequential steps

$$
\left[\begin{array}{ccccccccccccccccc}
1 & & u_0^* & w_0^* & & & & & & & & & & & & & \\
 & 1 & u_1^* & w_1^* & & & & & & & & & & & & & \\
s_2^* & l_2^* & 1 & & u_2^* & w_2^* & & & & & & & & & & & \\
s_3^* & l_3^* & & 1 & u_3^* & w_3^* & & & & & & & & & & & \\
 & & s_4^* & l_4^* & 1 & & u_4^* & w_4^* & & & & & & & & & \\
 & & s_5^* & l_5^* & & 1 & u_5^* & w_5^* & & & & & & & & & \\
 & & & & s_6^* & l_6^* & 1 & & u_6^* & w_6^* & & & & & & & \\
 & & & & s_7^* & l_7^* & & 1 & u_7^* & w_7^* & & & & & & & \\
 & & & & & & s_8^* & l_8^* & 1 & & u_8^* & w_8^* & & & & & \\
 & & & & & & s_9^* & l_9^* & & 1 & u_9^* & w_9^* & & & & & \\
 & & & & & & & & s_{10}^* & l_{10}^* & 1 & & u_{10}^* & w_{10}^* & & & \\
 & & & & & & & & s_{11}^* & l_{11}^* & & 1 & u_{11}^* & w_{11}^* & & & \\
 & & & & & & & & & & s_{12}^* & l_{12}^* & 1 & & u_{12}^* & w_{12}^* & \\
 & & & & & & & & & & s_{13}^* & l_{13}^* & & 1 & u_{13}^* & w_{13}^* & \\
 & & & & & & & & & & & & s_{14}^* & l_{14}^* & 1 & & u_{14}^* \\
 & & & & & & & & & & & & s_{15}^* & l_{15}^* & & 1 & u_{15}^* \\
 & & & & & & & & & & & & & & s_{16}^* & l_{16}^* & 1
\end{array}\right]
\left[\begin{array}{c}
x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \\ x_{13} \\ x_{14} \\ x_{15} \\ x_{16}
\end{array}\right]
=
\left[\begin{array}{c}
b_0^* \\ b_1^* \\ b_2^* \\ b_3^* \\ b_4^* \\ b_5^* \\ b_6^* \\ b_7^* \\ b_8^* \\ b_9^* \\ b_{10}^* \\ b_{11}^* \\ b_{12}^* \\ b_{13}^* \\ b_{14}^* \\ b_{15}^* \\ b_{16}^*
\end{array}\right]
$$

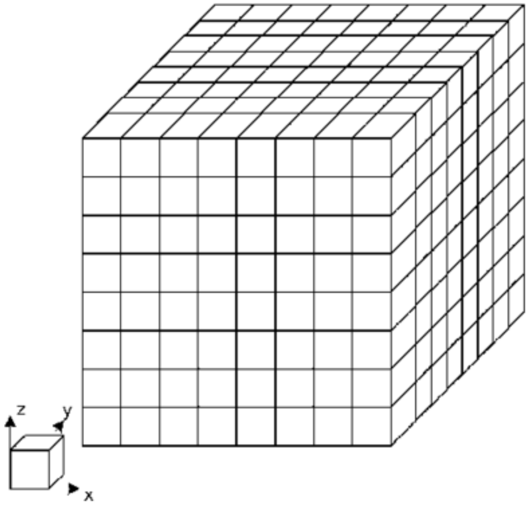
Solving pentadiagonal equations in parallel

- Hybrid
  - Doing Forward on blocks in parallel
  - Solving "reduced" system
  - Backsubstitution in parallel
  - O(N)

$$
\left[\begin{array}{ccccccccccccccccc}
1 & & & & & & u_0^* & w_0^* & & & & & & & & & \\
 & 1 & & & & & u_1^* & w_1^* & & & & & & & & & \\
s_2^* & l_2^* & 1 & u_2^* & w_2^* & & & & & & & & & & & & \\
s_3^* & l_3^* & & 1 & u_3^* & w_3^* & & & & & & & & & & & \\
s_4^* & l_4^* & & & 1 & & u_4^* & w_4^* & & & & & & & & & \\
s_5^* & l_5^* & & & & 1 & u_5^* & w_5^* & & & & & & & & & \\
\hline
 & & & & s_6^* & l_6^* & 1 & & & & & & u_6^* & w_6^* & & & \\
 & & & & s_7^* & l_7^* & & 1 & & & & & u_7^* & w_7^* & & & \\
 & & & & & & s_8^* & l_8^* & 1 & u_8^* & w_8^* & & & & & & \\
 & & & & & & s_9^* & l_9^* & & 1 & u_9^* & w_9^* & & & & & \\
 & & & & s_{10}^* & l_{10}^* & & & & & 1 & & u_{10}^* & w_{10}^* & & & \\
 & & & & s_{11}^* & l_{11}^* & & & & & & 1 & u_{11}^* & w_{11}^* & & & \\
\hline
 & & & & & & & & & & s_{12}^* & l_{12}^* & 1 & & & & \\
 & & & & & & & & & & s_{13}^* & l_{13}^* & & 1 & & & \\
 & & & & & & & & & & & & s_{14}^* & l_{14}^* & 1 & u_{14}^* & w_{14}^* \\
 & & & & & & & & & & s_{15}^* & l_{15}^* & & & & 1 & \\
 & & & & & & & & & & s_{16}^* & l_{16}^* & & & & & 1
\end{array}\right]
\left[\begin{array}{c}
x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ \hline x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ \hline x_{12} \\ x_{13} \\ x_{14} \\ x_{15} \\ x_{16}
\end{array}\right]
=
\left[\begin{array}{c}
b_0^* \\ b_1^* \\ b_2^* \\ b_3^* \\ b_4^* \\ b_5^* \\ \hline b_6^* \\ b_7^* \\ b_8^* \\ b_9^* \\ b_{10}^* \\ b_{11}^* \\ \hline b_{12}^* \\ b_{13}^* \\ b_{14}^* \\ b_{15}^* \\ b_{16}^*
\end{array}\right]
$$

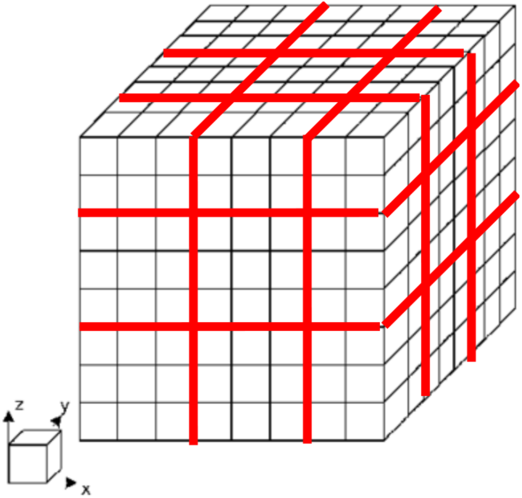
Batching & Higher dimensions

- Solving a single 1D system is easy
- Higher dimensional problem: Alternating Direction Implicit method
    - E.g. flow solvers based on implicit high-order finite-difference schemes
    - Solve along x direction, then y direction, then z
    - Coefficients & unknowns laid out on 3D mesh



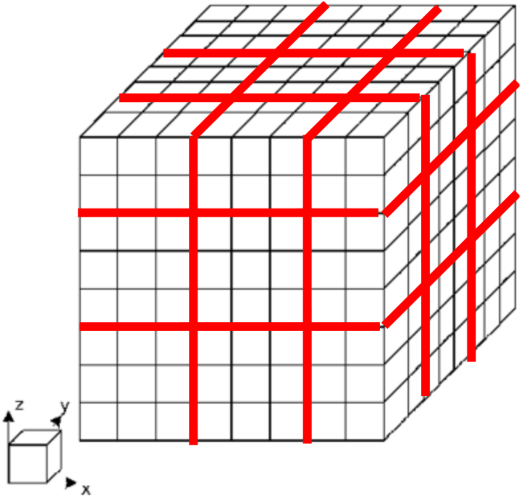
Pentadsolver

- Grid is decomposed along all the axes, processes only hold a block of any given pentadiagonal system
- Within each partition:
  Thomas + PCR/Jacobi Hybrid



Pentadsolver

- Grid is decomposed along all the axes, processes only hold a block of any given pentadiagonal system
- Within each partition:
  Thomas + PCR/Jacobi Hybrid
- Distributed reduced system solve options:
  - Allgather, solve, backsubstitute
    - One large comm step
  - PCR-X
    - Multiple smaller messages to increasingly distant processes
  - Approximate solution with Jacobi iteration
    - Multiple smaller messages to neighbors



Pentadsolver - Tomas Hybrid

$$
\left[\begin{array}{ccccccccccccccccc}
d_0 & u_0 & w_0 & & & & & & & & & & & & & & \\
l_1 & d_1 & u_1 & w_1 & & & & & & & & & & & & & \\
s_2 & l_2 & d_2 & u_2 & w_2 & & & & & & & & & & & & \\
 & s_3 & l_3 & d_3 & u_3 & w_3 & & & & & & & & & & & \\
 & & s_4 & l_4 & d_4 & u_4 & w_4 & & & & & & & & & & \\
 & & & s_5 & l_5 & d_5 & u_5 & w_5 & & & & & & & & & \\
\hline
 & & & & s_6 & l_6 & d_6 & u_6 & w_6 & & & & & & & & \\
 & & & & & s_7 & l_7 & d_7 & u_7 & w_7 & & & & & & & \\
 & & & & & & s_8 & l_8 & d_8 & u_8 & w_8 & & & & & & \\
 & & & & & & & s_9 & l_9 & d_9 & u_9 & w_9 & & & & & \\
 & & & & & & & & s_{10} & l_{10} & d_{10} & u_{10} & w_{10} & & & & \\
 & & & & & & & & & s_{11} & l_{11} & d_{11} & u_{11} & w_{11} & & & \\
\hline
 & & & & & & & & & & s_{12} & l_{12} & d_{12} & u_{12} & w_{12} & & \\
 & & & & & & & & & & & s_{13} & l_{13} & d_{13} & u_{13} & w_{13} & \\
 & & & & & & & & & & & & s_{14} & l_{14} & d_{14} & u_{14} & w_{14} \\
 & & & & & & & & & & & & & s_{15} & l_{15} & d_{15} & u_{15} \\
 & & & & & & & & & & & & & & s_{16} & l_{16} & d_{16}
\end{array}\right]
\left[\begin{array}{c} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ \hline x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ \hline x_{12} \\ x_{13} \\ x_{14} \\ x_{15} \\ x_{16} \end{array}\right]
=
\left[\begin{array}{c} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ \hline b_6 \\ b_7 \\ b_8 \\ b_9 \\ b_{10} \\ b_{11} \\ \hline b_{12} \\ b_{13} \\ b_{14} \\ b_{15} \\ b_{16} \end{array}\right]
$$

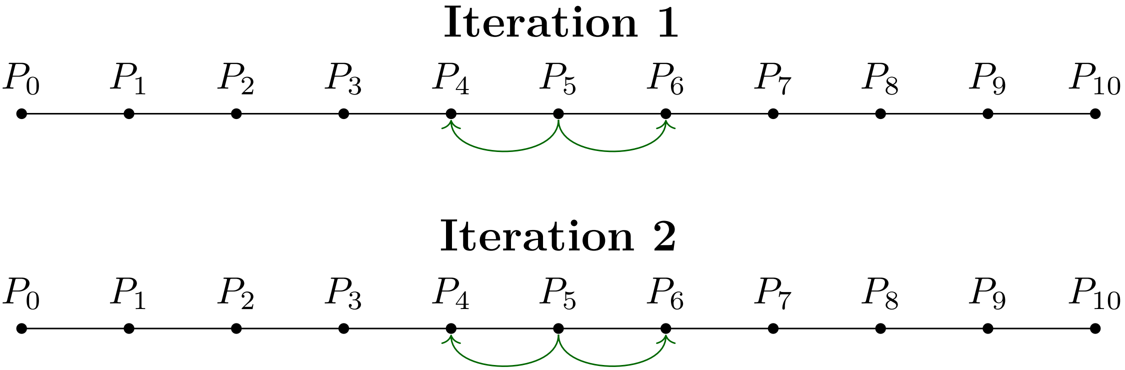
Pentadsolver - Thomas Hybrid

$$
\left[\begin{array}{ccccccccccccccccc}
\mathbf{1} & & & & & & \boldsymbol{u_0^*} & \boldsymbol{w_0^*} & & & & & & & & & \\
 & \mathbf{1} & & & & & \boldsymbol{u_1^*} & \boldsymbol{w_1^*} & & & & & & & & & \\
s_2^* & l_2^* & 1 & u_2^* & w_2^* & & & & & & & & & & & & \\
s_3^* & l_3^* & & 1 & u_3^* & w_3^* & & & & & & & & & & & \\
\boldsymbol{s_4^*} & \boldsymbol{l_4^*} & & & \mathbf{1} & & \boldsymbol{u_4^*} & \boldsymbol{w_4^*} & & & & & & & & & \\
\boldsymbol{s_5^*} & \boldsymbol{l_5^*} & & & & \mathbf{1} & \boldsymbol{u_5^*} & \boldsymbol{w_5^*} & & & & & & & & & \\
\hline
 & & & & \boldsymbol{s_6^*} & \boldsymbol{l_6^*} & \mathbf{1} & & & & & & \boldsymbol{u_6^*} & \boldsymbol{w_6^*} & & & \\
 & & & & \boldsymbol{s_7^*} & \boldsymbol{l_7^*} & & \mathbf{1} & & & & & \boldsymbol{u_7^*} & \boldsymbol{w_7^*} & & & \\
 & & & & & & s_8^* & l_8^* & 1 & u_8^* & w_8^* & & & & & & \\
 & & & & & & s_9^* & l_9^* & & 1 & u_9^* & w_9^* & & & & & \\
 & & & & \boldsymbol{s_{10}^*} & \boldsymbol{l_{10}^*} & & & & & \mathbf{1} & & \boldsymbol{u_{10}^*} & \boldsymbol{w_{10}^*} & & & \\
 & & & & \boldsymbol{s_{11}^*} & \boldsymbol{l_{11}^*} & & & & & & \mathbf{1} & \boldsymbol{u_{11}^*} & \boldsymbol{w_{11}^*} & & & \\
\hline
 & & & & & & & & & & \boldsymbol{s_{12}^*} & \boldsymbol{l_{12}^*} & \mathbf{1} & & & & \\
 & & & & & & & & & & \boldsymbol{s_{13}^*} & \boldsymbol{l_{13}^*} & & \mathbf{1} & & & \\
 & & & & & & & & & & & & s_{14}^* & l_{14}^* & 1 & u_{14}^* & w_{14}^* \\
 & & & & & & & & & & \boldsymbol{s_{15}^*} & \boldsymbol{l_{15}^*} & & & & \mathbf{1} & \\
 & & & & & & & & & & \boldsymbol{s_{16}^*} & \boldsymbol{l_{16}^*} & & & & & \mathbf{1}
\end{array}\right]
\left[\begin{array}{c}
\boldsymbol{x_0} \\ \boldsymbol{x_1} \\ x_2 \\ x_3 \\ \boldsymbol{x_4} \\ \boldsymbol{x_5} \\ \hline \boldsymbol{x_6} \\ \boldsymbol{x_7} \\ x_8 \\ x_9 \\ \boldsymbol{x_{10}} \\ \boldsymbol{x_{11}} \\ \hline \boldsymbol{x_{12}} \\ \boldsymbol{x_{13}} \\ x_{14} \\ \boldsymbol{x_{15}} \\ \boldsymbol{x_{16}}
\end{array}\right]
=
\left[\begin{array}{c}
\boldsymbol{b_0^*} \\ \boldsymbol{b_1^*} \\ b_2^* \\ b_3^* \\ \boldsymbol{b_4^*} \\ \boldsymbol{b_5^*} \\ \hline \boldsymbol{b_6^*} \\ \boldsymbol{b_7^*} \\ b_8^* \\ b_9^* \\ \boldsymbol{b_{10}^*} \\ \boldsymbol{b_{11}^*} \\ \hline \boldsymbol{b_{12}^*} \\ \boldsymbol{b_{13}^*} \\ b_{14}^* \\ \boldsymbol{b_{15}^*} \\ \boldsymbol{b_{16}^*}
\end{array}\right]
$$

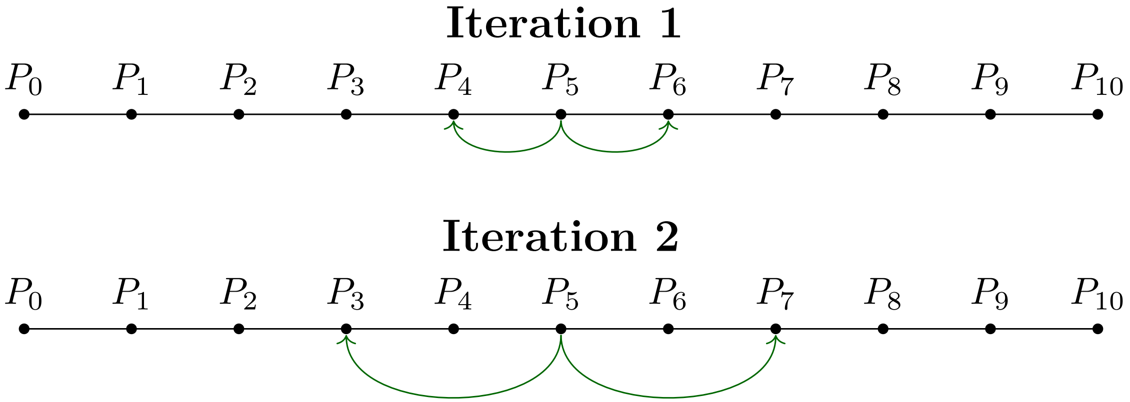
Pentadsolver - Thomas Hybrid

$$
\left[\begin{array}{ccccccccccccccccc}
\mathbf{1} & & & & & & \boldsymbol{u_0^*} & \boldsymbol{w_0^*} & & & & & & & & & \\
 & \mathbf{1} & & & & & \boldsymbol{u_1^*} & \boldsymbol{w_1^*} & & & & & & & & & \\
s_2^* & l_2^* & 1 & u_2^* & w_2^* & & & & & & & & & & & & \\
s_3^* & l_3^* & & 1 & u_3^* & w_3^* & & & & & & & & & & & \\
\boldsymbol{s_4^*} & \boldsymbol{l_4^*} & & & \mathbf{1} & & \boldsymbol{u_4^*} & \boldsymbol{w_4^*} & & & & & & & & & \\
\boldsymbol{s_5^*} & \boldsymbol{l_5^*} & & & & \mathbf{1} & \boldsymbol{u_5^*} & \boldsymbol{w_5^*} & & & & & & & & & \\
\hline
 & & & & \boldsymbol{s_6^*} & \boldsymbol{l_6^*} & \mathbf{1} & & & & & & \boldsymbol{u_6^*} & \boldsymbol{w_6^*} & & & \\
 & & & & \boldsymbol{s_7^*} & \boldsymbol{l_7^*} & & \mathbf{1} & & & & & \boldsymbol{u_7^*} & \boldsymbol{w_7^*} & & & \\
 & & & & & & s_8^* & l_8^* & 1 & u_8^* & w_8^* & & & & & & \\
 & & & & & & s_9^* & l_9^* & & 1 & u_9^* & w_9^* & & & & & \\
 & & & & & & \boldsymbol{s_{10}^*} & \boldsymbol{l_{10}^*} & & & \mathbf{1} & & \boldsymbol{u_{10}^*} & \boldsymbol{w_{10}^*} & & & \\
 & & & & & & \boldsymbol{s_{11}^*} & \boldsymbol{l_{11}^*} & & & & \mathbf{1} & \boldsymbol{u_{11}^*} & \boldsymbol{w_{11}^*} & & & \\
\hline
 & & & & & & & & & & \boldsymbol{s_{12}^*} & \boldsymbol{l_{12}^*} & \mathbf{1} & & & & \\
 & & & & & & & & & & \boldsymbol{s_{13}^*} & \boldsymbol{l_{13}^*} & & \mathbf{1} & & & \\
 & & & & & & & & & & & & s_{14}^* & l_{14}^* & 1 & u_{14}^* & w_{14}^* \\
 & & & & & & & & & & & & \boldsymbol{s_{15}^*} & \boldsymbol{l_{15}^*} & & \mathbf{1} & \\
 & & & & & & & & & & & & \boldsymbol{s_{16}^*} & \boldsymbol{l_{16}^*} & & & \mathbf{1}
\end{array}\right]
\left[\begin{array}{c}
\boldsymbol{x_0} \\ \boldsymbol{x_1} \\ x_2 \\ x_3 \\ \boldsymbol{x_4} \\ \boldsymbol{x_5} \\ \hline \boldsymbol{x_6} \\ \boldsymbol{x_7} \\ x_8 \\ x_9 \\ \boldsymbol{x_{10}} \\ \boldsymbol{x_{11}} \\ \hline \boldsymbol{x_{12}} \\ \boldsymbol{x_{13}} \\ x_{14} \\ \boldsymbol{x_{15}} \\ \boldsymbol{x_{16}}
\end{array}\right]
=
\left[\begin{array}{c}
\boldsymbol{b_0^*} \\ \boldsymbol{b_1^*} \\ b_2^* \\ b_3^* \\ \boldsymbol{b_4^*} \\ \boldsymbol{b_5^*} \\ \hline \boldsymbol{b_6^*} \\ \boldsymbol{b_7^*} \\ b_8^* \\ b_9^* \\ \boldsymbol{b_{10}^*} \\ \boldsymbol{b_{11}^*} \\ \hline \boldsymbol{b_{12}^*} \\ \boldsymbol{b_{13}^*} \\ b_{14}^* \\ \boldsymbol{b_{15}^*} \\ \boldsymbol{b_{16}^*}
\end{array}\right]
$$

Communication in the reduced solve: Jacobi iterations



Communication in the reduced solve: PCR with blocks

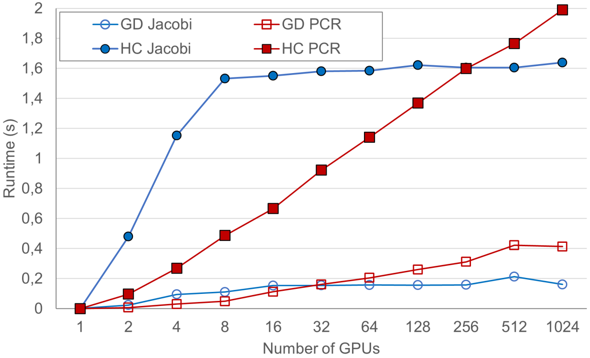


Test setup

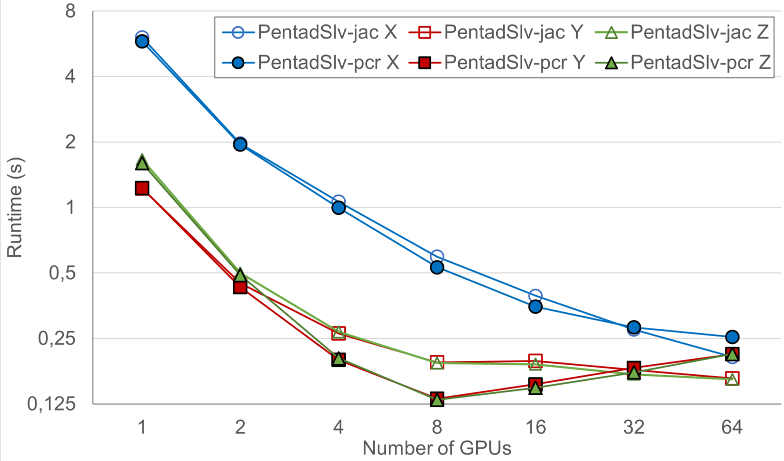
- LUMI
- AMD Trento CPU, 64 cores
- 4x AMD MI250X per node (8 GPUs per node)
- 10 iterations for Jacobi solver
- All MPI ranks in the solve direction

Pentadsolver - Scaling - Reduced solve

- All other parts of the solver trivially scales
- **H**ost **C**opy vs **G**PU **D**irect
- Fix 10 Jacobi iterations
  - Performance change on GPU and node borders
- PCR with blocks shows the cost of each additional iteration as well

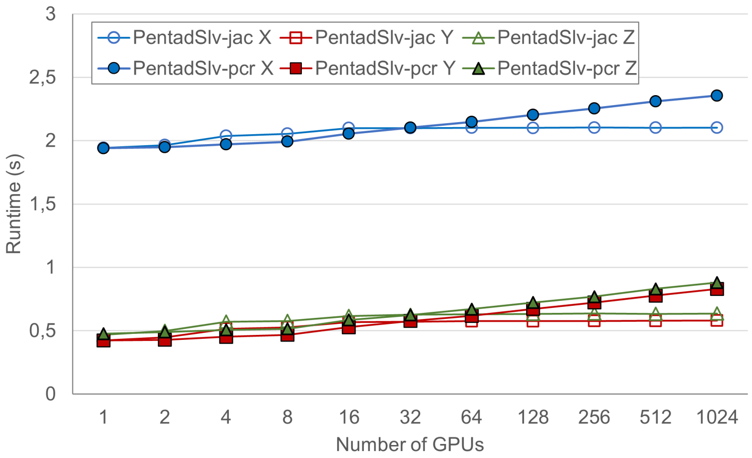


Pentadsolver - Strong Scaling $1024 \times 512 \times 512$



- 80% Scaling efficiency until 8 GPUs ($128 \times 512 \times 512$)
- After 8 GPUs the problem is just too small

Pentadsolver - Weak Scaling 512^3



- Jacobi: 92% (X) and 73% (Y, Z) scaling efficiency to 1024 GPUs
- PCR: 98% (X), 80% (Y, Z) scaling efficiency to 8, 82% (X), 51% (Y,Z) to 1024 GPUs

Conclusions

- Parallelization of batch pentadiagonal system solves: across systems
- Can be used in distributed memory systems
- Scaling behavior determined by the reduced solve
  - Close communication is much cheaper
  - Small cost of each iteration of the reduced solve adds up

Supported by the ÚNKP-22-4 New National Excellence Program of the Ministry for Culture and Innovation from the source of the National Research, Development and Innovation Fund.

NEMZETI KUTATÁSI, FEJLESZTÉSI ÉS INNOVÁCIÓS HIVATAL

Új Nemzeti Kiválóság Program