A Distributed Memory Tri-diagonal Solver Optimised for CPU and GPU Architectures

Semih Akkurt and Sylvain Laizet

Department of Aeronautics
Imperial College London

CIUK 2023
7 December 2023, Manchester UK

Overview

- Various discretisations for solving PDE's result in tridiagonal systems.
- This study focuses on Compact Finite Difference Schemes.
  - Space-implicit coupling in compact schemes results in tridiagonal systems.
- In Xcompact3D solver, you can have around 50 batches of tridiagonal systems where each batch may contain around $\sim 10^{6/7}$ individual systems of size $\sim 10^{3/4}$ with a total DOF up to around $\sim 10^{12}$.
- Challenge on supercomputers.

High-order Compact Finite Difference Schemes

- Example from Xcompact3D based on 6th order compact FD schemes.
- We solve $\sim$ 50 batches of such systems per time step.

$$\alpha f'_{i-1} + f'_i + \alpha f'_{i+1} = a\frac{f_{i+1} - f_{i-1}}{2\Delta x} + b\frac{f_{i+2} - f_{i-2}}{4\Delta x}$$

$$\begin{bmatrix} b_0 & c_0 & & & & & \\ a_1 & b_1 & c_1 & & & & \\ & a_2 & b_2 & c_2 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & & a_n & b_n \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_{n-1} \\ u_n \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ \vdots \\ d_{n-1} \\ d_n \end{bmatrix}$$

Thomas Algorithm with 1D/2D Decomposition

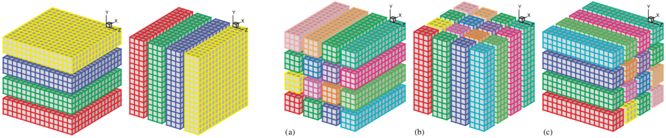

Figure: 1D/2D (Slab/Pencil) Domain Decomposition

- Currently, Xcompact3D does not use distributed solvers.
- Thus, Xcompact3D require 12 pairs of data transfers per iteration.
- These large scale data transfers can be costly.
- The best case behaviour can be modelled theoretically.

512^3 Domain

# GPUs	Comp. BW	Comm. BW	Comp. Time	Comm. Time	Total Time
1	332 GiB	0 GiB	0.300 s	0 s	0.300 s
2	166 GiB	3 GiB	0.150 s	0.015 s	0.165 s
4	83 GiB	2.25 GiB	0.075 s	0.004 s	0.079 s
8	42 GiB	1.31 GiB	0.038 s	0.060 s	0.098 s

Parallelisation Strategy - Distributed Tridiagonal Solvers

- Examples algorithms are PDD (Sun 1995) and SPIKE (Polizzi and Sameh 2007)
- Our strategy combines Hybrid Thomas-PCR (Laszlo 2016) and PDD (Sun 1995) algorithms
- 1/2/3 D domain decomposition, no all-to-all communication between sub-domains.
- A distributed solver is implemented along the direction that is split between ranks

$$
\left[\begin{array}{cccccc|cccccc}
b_0 & c_0 & & & & & & & & & & a_0 \\
a_1 & b_1 & c_1 & & & & & & & & & \\
 & a_2 & b_2 & c_2 & & & & & & & & \\
 & & \ddots & \ddots & \ddots & & & & & & & \\
 & & & a_{m-1} & b_{m-1} & c_{m-1} & & & & & & \\
 & & & & a_m & b_m & c_m & & & & & \\
\hline
 & & & & & a_0 & b_0 & c_0 & & & & \\
 & & & & & & a_1 & b_1 & c_1 & & & \\
 & & & & & & & a_2 & b_2 & c_2 & & \\
 & & & & & & & & \ddots & \ddots & \ddots & \\
 & & & & & & & & & a_{m-1} & b_{m-1} & c_{m-1} \\
c_m & & & & & & & & & & a_m & b_m
\end{array}\right] \tag{1}
$$

Parallelisation Strategy - Distributed Tridiagonal Solvers

$$
\left[\begin{array}{cccccc|cccccc}
1 & c_0 & & & & & & & & & & a_0^* \\
a_1^* & 1 & c_1 & & & & & & & & & \\
a_2^* & & 1 & c_2 & & & & & & & & \\
\vdots & & & \ddots & \ddots & & & & & & & \\
a_{m-1}^* & & & & 1 & c_{m-1} & & & & & & \\
a_m^* & & & & & 1 & c_m & & & & & \\
\hline
 & & & & & a_0^* & 1 & c_0 & & & & \\
 & & & & & & a_1^* & 1 & c_1 & & & \\
 & & & & & & a_2^* & & 1 & c_2 & & \\
 & & & & & & \vdots & & & \ddots & \ddots & \\
 & & & & & & a_{m-1}^* & & & & 1 & c_{m-1} \\
c_m & & & & & & a_m^* & & & & & 1
\end{array}\right] \tag{2}
$$

$$
\left[
\begin{array}{cccccc|cccccc}
1 & & & & & c_0^* & & & & & & a_0^* \\
a_1^* & 1 & & & & c_1^* & & & & & & \\
a_2^* & & 1 & & & c_2^* & & & & & & \\
\vdots & & & \ddots & & \vdots & & & & & & \\
a_{m-1}^* & & & & 1 & c_{m-1}^* & & & & & & \\
a_m^* & & & & & 1 & c_m^* & & & & & \\
\hline
 & & & & & a_0^* & 1 & & & & & c_0^* \\
 & & & & & & a_1^* & 1 & & & & c_1^* \\
 & & & & & & a_2^* & & 1 & & & c_2^* \\
 & & & & & & \vdots & & & \ddots & & \vdots \\
 & & & & & & a_{m-1}^* & & & & 1 & c_{m-1}^* \\
c_m^* & & & & & & a_m^* & & & & & 1
\end{array}
\right]
\tag{3}
$$

Parallelisation Strategy - Distributed Tridiagonal Solvers

$$
\left[\begin{array}{cccccc|cccccc}
1 & & & & & \cancel{c_0^*} & & & & & & a_0^* \\
a_1^* & 1 & & & & c_1^* & & & & & & \\
a_2^* & & 1 & & & c_2^* & & & & & & \\
\vdots & & & \ddots & & \vdots & & & & & & \\
a_{m-1}^* & & & & 1 & c_{m-1}^* & & & & & & \\
\cancel{a_m^*} & & & & & 1 & c_m^* & & & & & \\
\hline
 & & & & & a_0^* & 1 & & & & & \cancel{c_0^*} \\
 & & & & & & a_1^* & 1 & & & & c_1^* \\
 & & & & & & a_2^* & & 1 & & & c_2^* \\
 & & & & & & \vdots & & & \ddots & & \vdots \\
 & & & & & & a_{m-1}^* & & & & 1 & c_{m-1}^* \\
c_m^* & & & & & & \cancel{a_m^*} & & & & & 1
\end{array}\right] \tag{3}
$$

$$
\left[\begin{array}{cccccc|cccccc}
1 & & & & & \cancel{c_0^*} & & & & & & a_0^* \\
a_1^* & 1 & & & & c_1^* & & & & & & \\
a_2^* & & 1 & & & c_2^* & & & & & & \\
\vdots & & & \ddots & & \vdots & & & & & & \\
a_{m-1}^* & & & & 1 & c_{m-1}^* & & & & & & \\
\cancel{a_m^*} & & & & & 1 & c_m^* & & & & & \\
\hline
& & & & & a_0^* & 1 & & & & & \cancel{c_0^*} \\
& & & & & & a_1^* & 1 & & & & c_1^* \\
& & & & & & a_2^* & & 1 & & & c_2^* \\
& & & & & & \vdots & & & \ddots & & \vdots \\
& & & & & & a_{m-1}^* & & & & 1 & c_{m-1}^* \\
c_m^* & & & & & & \cancel{a_m^*} & & & & & 1
\end{array}\right] \qquad (3)
$$

Parallelisation Strategy - Distributed Tridiagonal Solvers

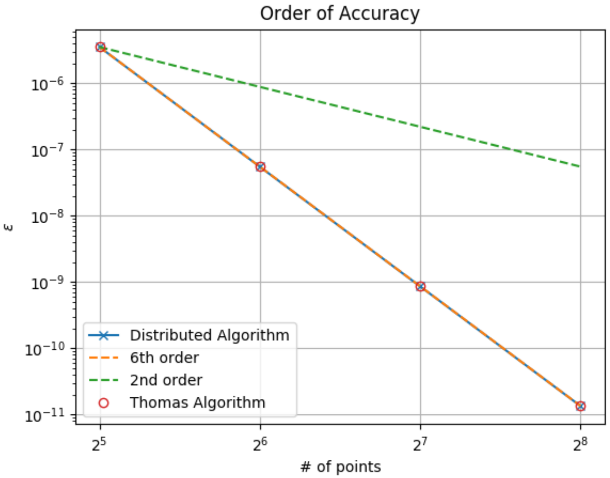

Figure: Order of Accuracy comparison.

Strong Scaling - Distributed Tridiagonal Solvers

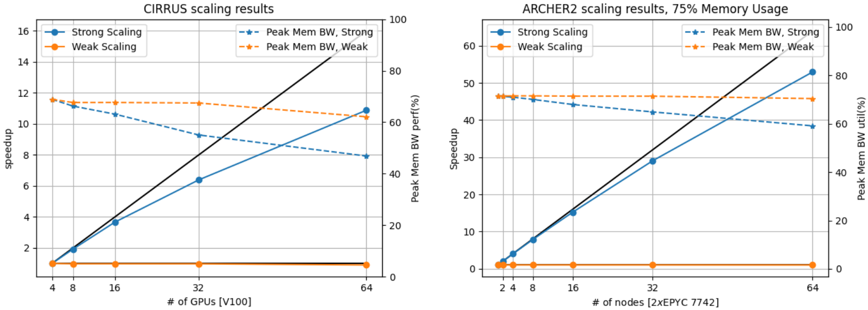

Figure: Strong Scaling on a GPU and CPU cluster.

Conclusion

- Number of points per rank is important for accuracy
- Global communications are eliminated for the tridiagonal solvers.
  - Few layers halo-data communication between previous and next ranks
- Constant communication overhead regardless of the number of ranks

Thank you!