Portable HPC Toolbox for Simulation and Inversion of Waves

Tilman Steinweg (KIT),Thomas Bohlen (KIT), Mark Wienöbst (KIT), Eliakim Schünemann (TEEC), Rüdiger Zehn (TEEC), Gerald Eisenberg-Klein (TEEC), Thomas Soddemann (SCAI), Thomas Brandes (SCAI), Moritz von Looz (U Berlin), Maria Predari (U Berlin), Charilaos Tzovas (U Berlin), Henning Meyerhenke (U Berlin)

17.10.2019

Simulation & Inversion

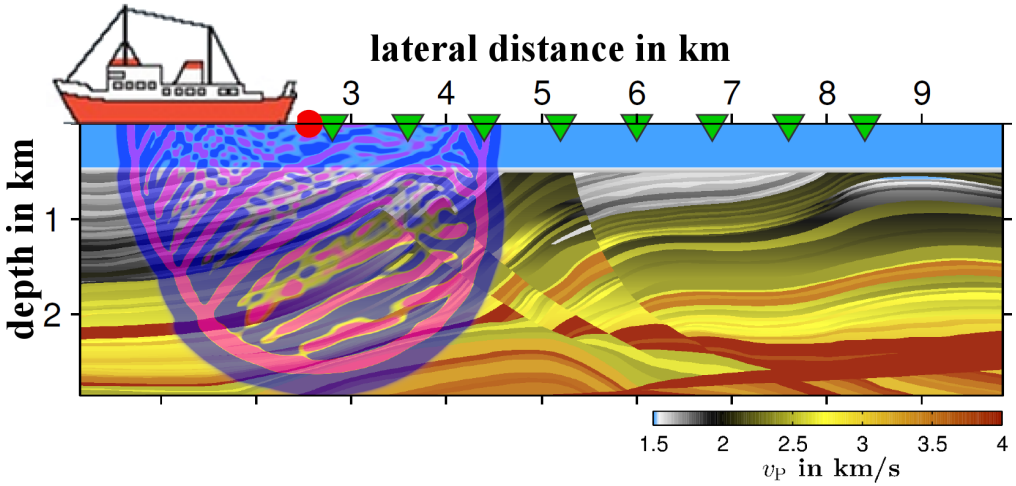
HPC Library LAMA

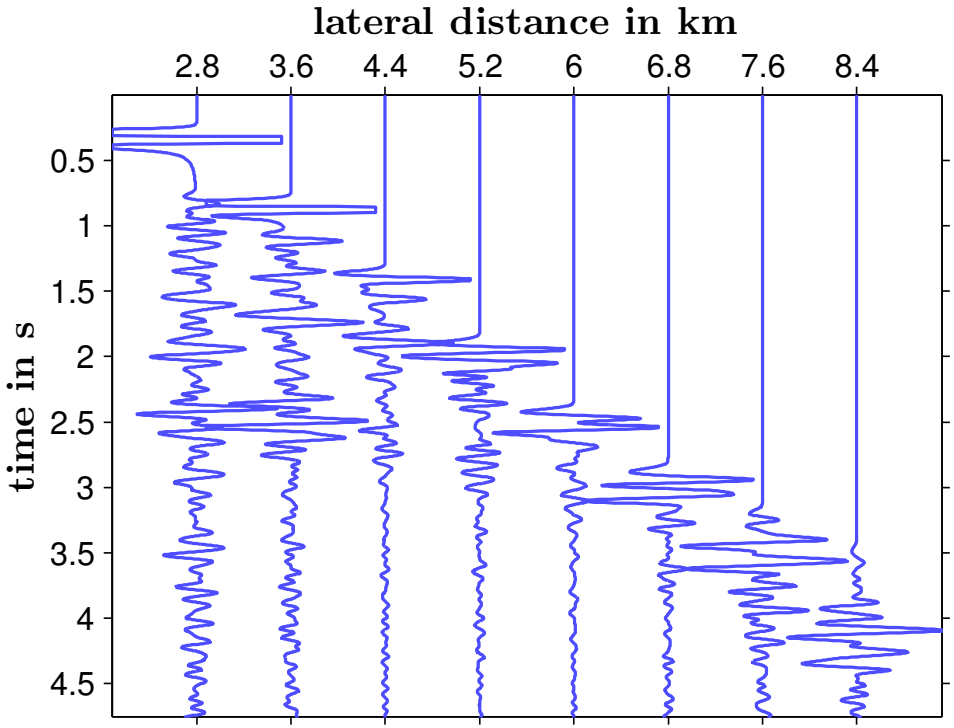
Load Balancing

Applications

Agenda

- Simulation and Inversion of Seismic Waves
- HPC Toolbox
  - FD Method
  - Discontinuous Grid
  - Partitioning
- Benchmark CPU/GPU
- Conclusion
- Outlook: Application of FWI



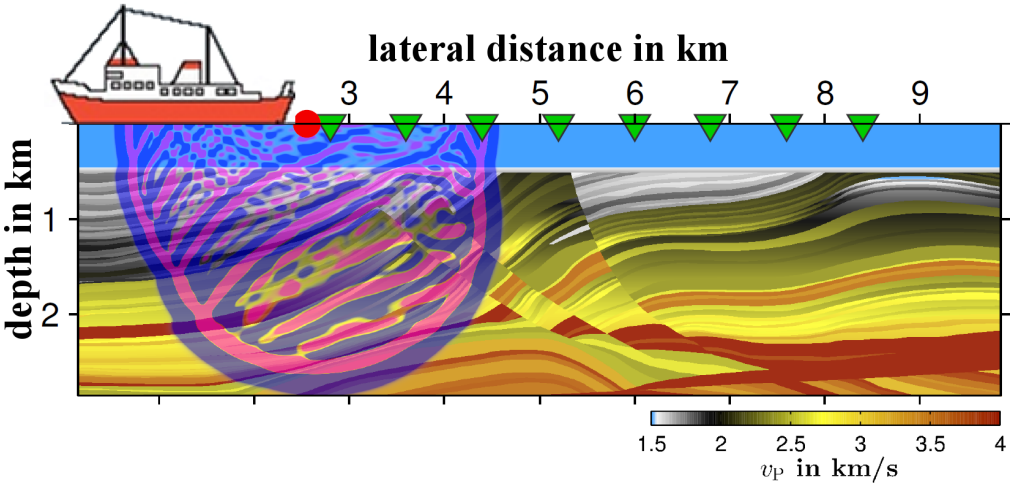


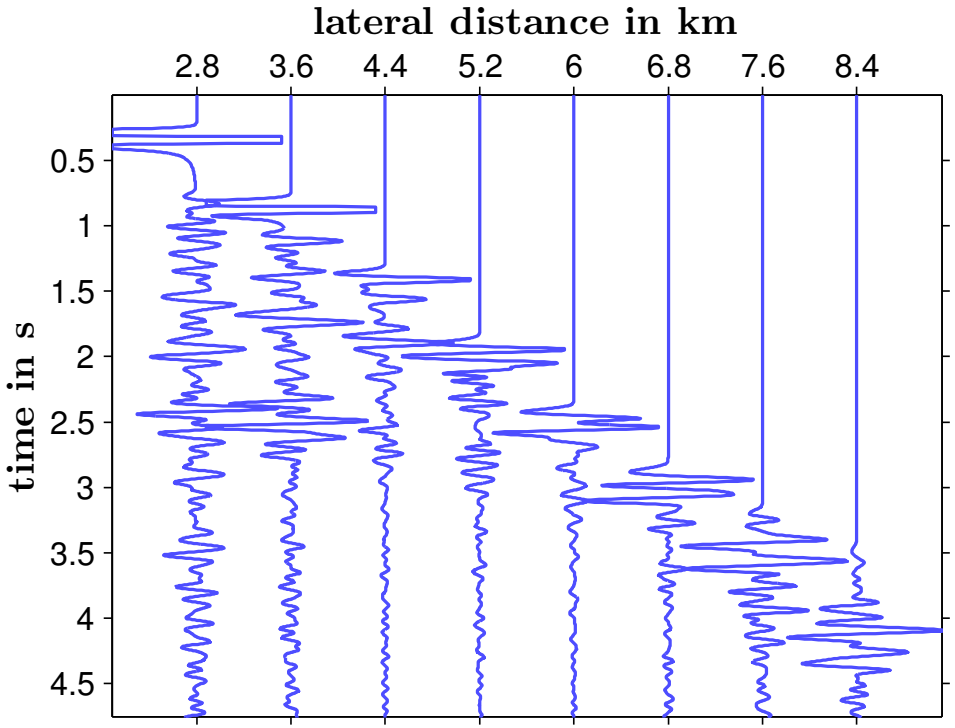
Simulation of Seismic Waves

Modelparameter:

- medium velocities v_p / v_s
- density
- attenuation

Wave equation (PDE) solved with finite difference method.





Full-Waveform Inversion

- inverse problem of seismic wave propagation
- find a model which explains seismic data
- full information of the recorded seismogram

C++ HPC Toolbox

HPC Tools

1. matrix-vector finite-difference solver (2D/3D acoustic / elastic / viscoelastic)
2. HPC library LAMA (mpi / OpenMP)
3. applications on heterogeneous architectures (CPU / GPU / Accelerator)
4. load balancing / graph partitioning (Geographer / ParMETIS)

C++ HPC Toolbox

HPC Tools

1. matrix-vector finite-difference solver (2D/3D acoustic / elastic / viscoelastic)
2. HPC library LAMA (mpi / OpenMP)
3. applications on heterogeneous architectures (CPU / GPU / Accelerator)
4. load balancing / graph partitioning (Geographer / ParMETIS)

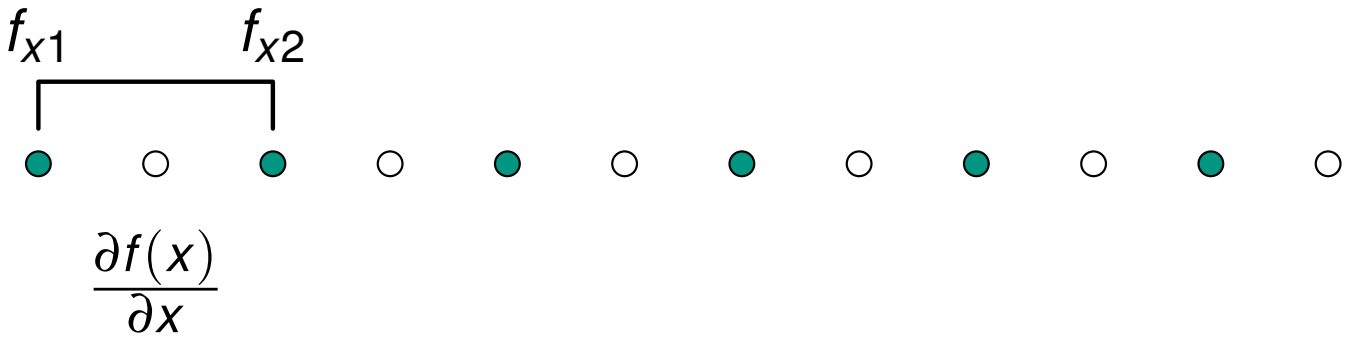
Focus in this talk

- matrix-vector formalism in FD-solver
- discontinuous grid
- partitioning

FD method

Conventional 1D FD operator for 2nd order derivatives on a staggered grid:

$$\left.\frac{\partial f(x)}{\partial x}\right|_{\frac{x_1+x_2}{2}} = \frac{f_{x_2} - f_{x_1}}{\Delta x}$$



Matrix-Vector Formalism

The operator for every grid point can be expressed as a matrix-vector product:

$$\begin{pmatrix} \frac{\partial f(x)}{\partial x}\Big|_{\frac{x_1+x_2}{2}} \\ \frac{\partial f(x)}{\partial x}\Big|_{\frac{x_2+x_3}{2}} \\ \dots \end{pmatrix} = \frac{1}{\Delta x} \cdot \begin{pmatrix} -1 & 1 & 0 & \dots \\ 0 & -1 & 1 & \dots \\ \dots & \dots & \dots & \dots \end{pmatrix} \cdot \begin{pmatrix} f_{x_1} \\ f_{x_2} \\ f_{x_3} \\ \dots \end{pmatrix}$$

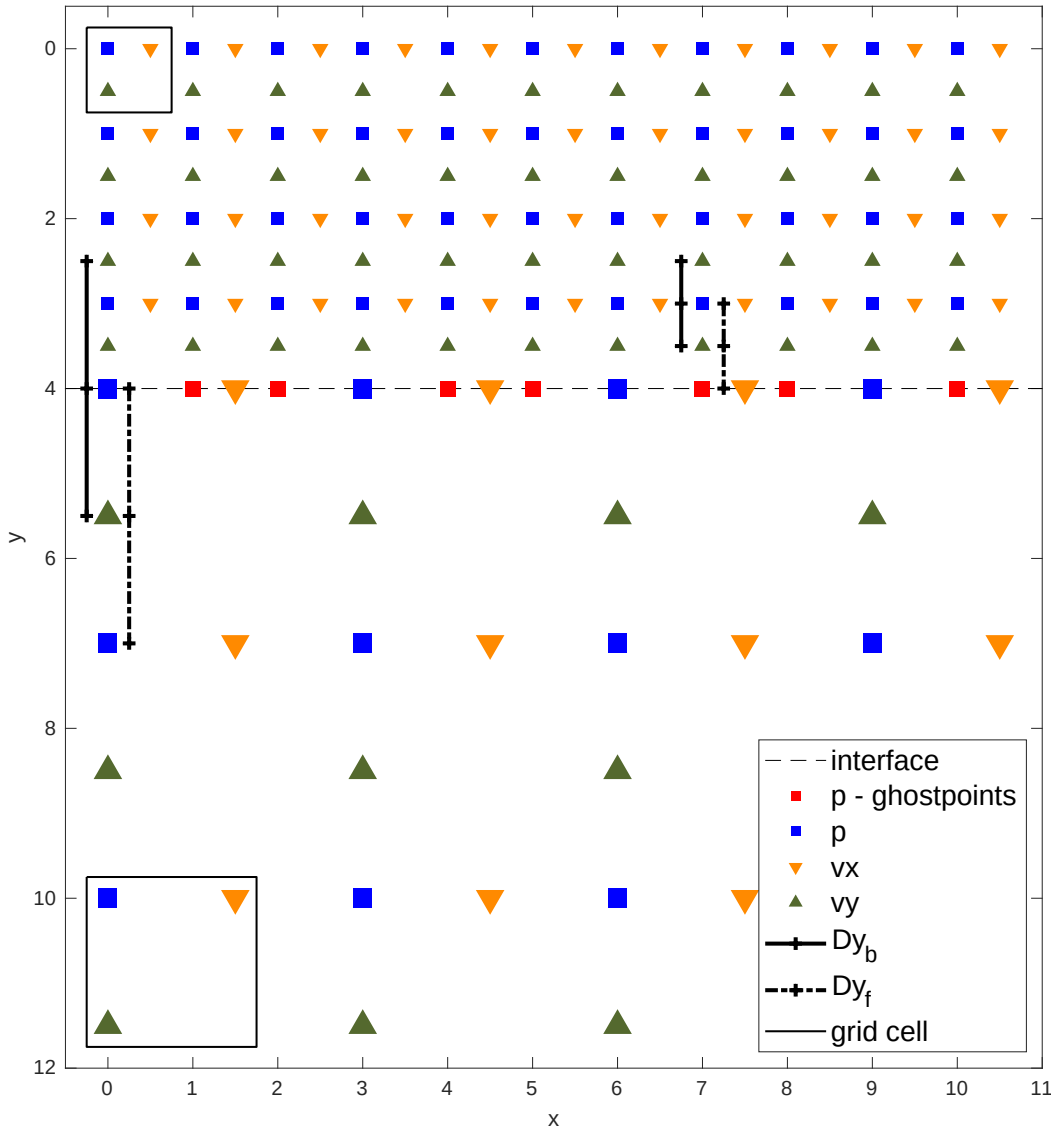
$$\frac{\partial \vec{f}}{\partial x} = \frac{1}{\Delta x} \cdot \underline{D} \cdot \vec{f}$$

- derivative matrix $\underline{D}$ has a strong sparsity rate
- FD coefficients in $\underline{D}$ can be adjusted for higher spatial order ($n = 2, 4, 6 \dots$)
- matrix with n non-zero elements per row

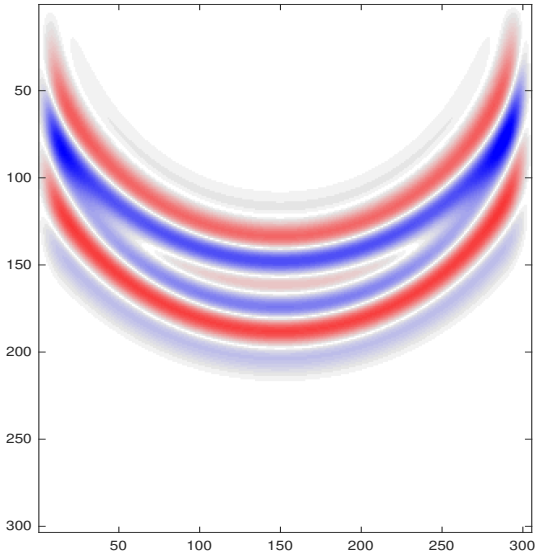
Discontinuous Grid

2D Acoustic example:

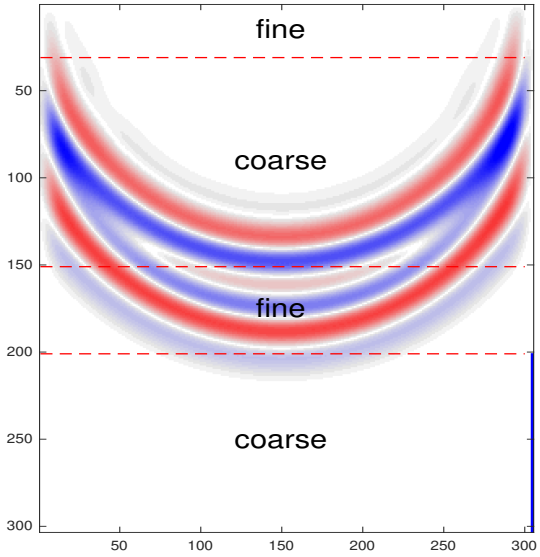
- connected regular grids
- 1D / 2D interpolation at the interfaces
- grid spacing factor: 3



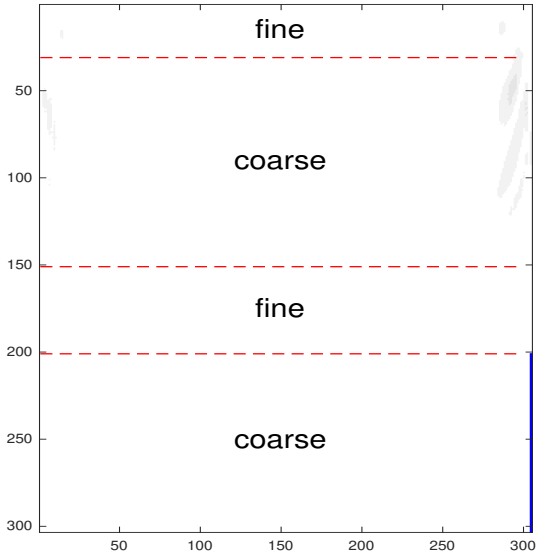
Discontinuous Grid



regular grid



discontinuous grid



difference x 2

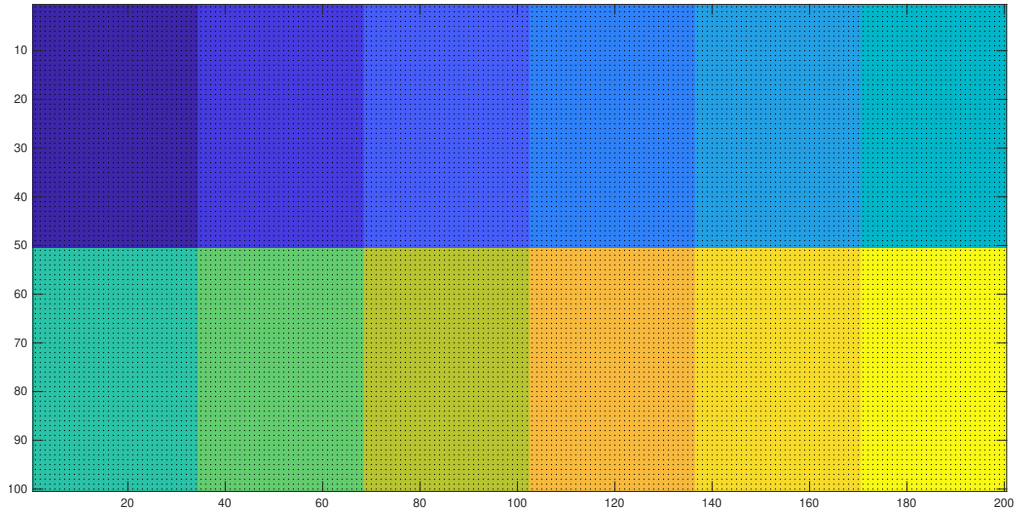
Setup

- homogeneous model
- 4 grid layers

Partitioning - Regular Grid

Grid partition

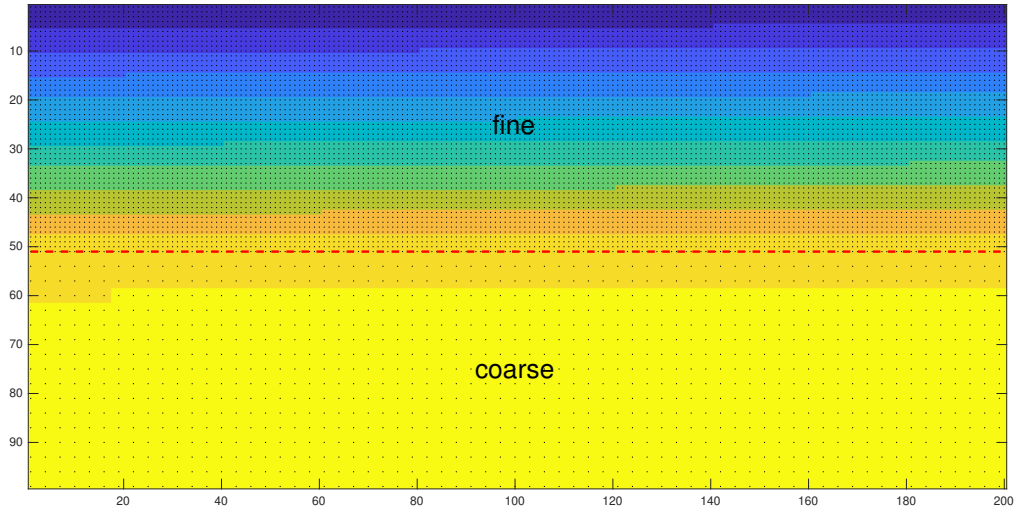
- regular process grid: NPx * NPy



Partitioning - Discontinuous Grid

Block partition

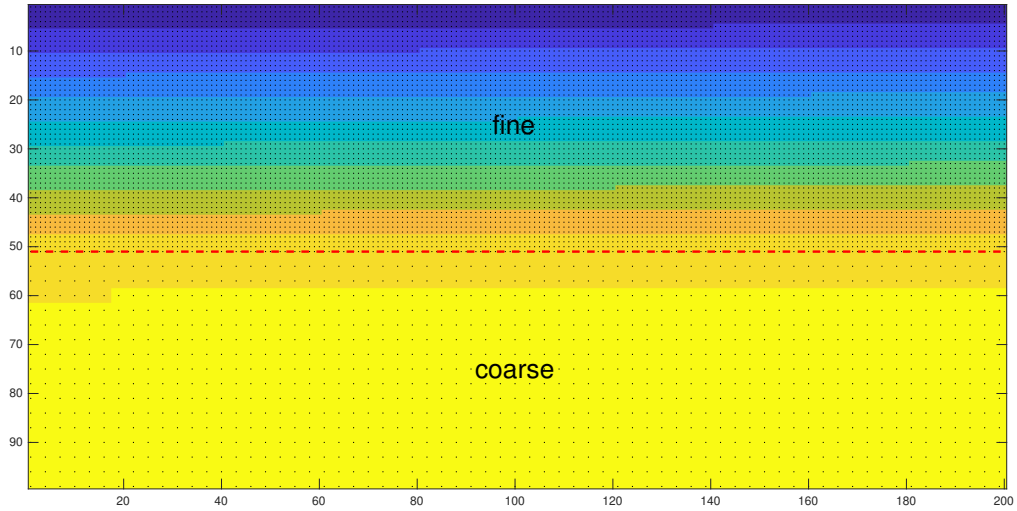
- equal sized blocks of model vector



Partitioning - Discontinuous Grid

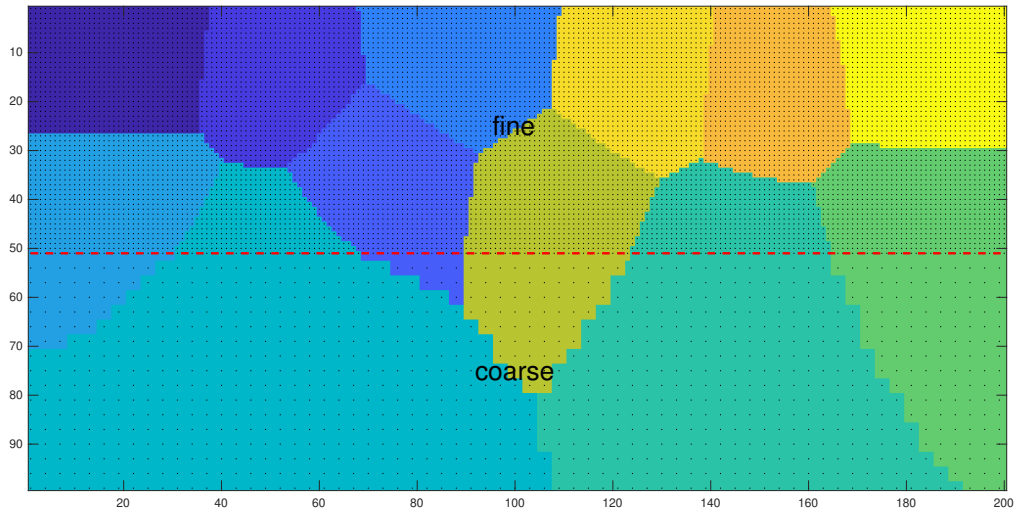
Block partition

- equal sized blocks of model vector

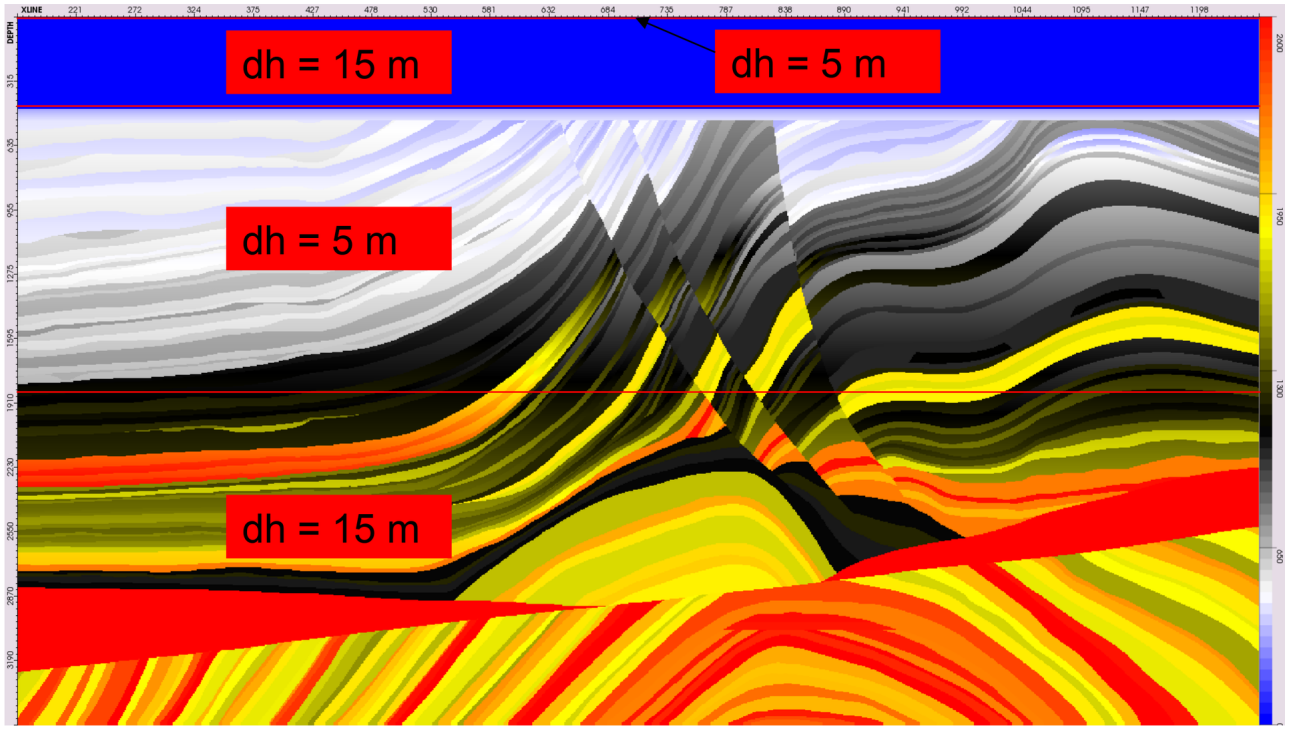


Graph partition (Geographer)

- start partition: block partition
- algorithm: k-means
- weights (imbalance)
  - absorbing boundary
  - variable FD-order
  - interpolation ...



3D Elastic Benchmark



FD-Order=2
FD-Order=6

FD-Order=12

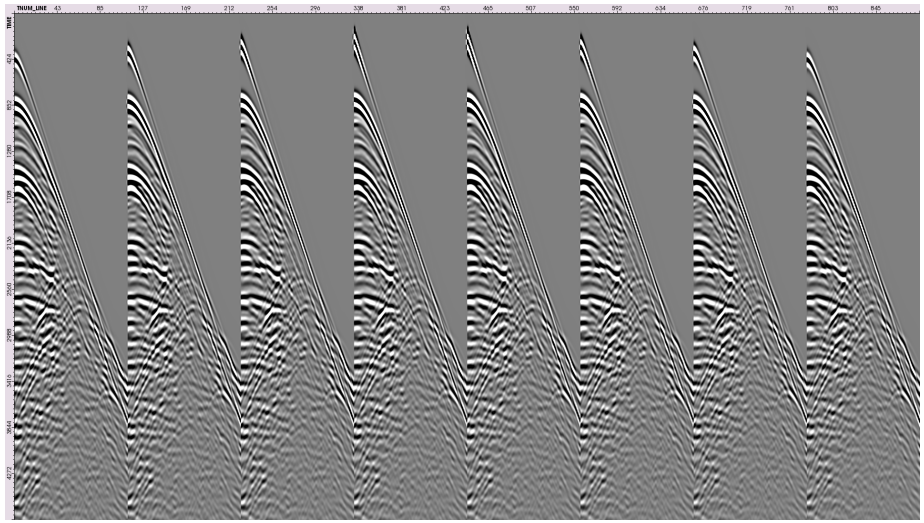
FD-Order=10

Setup

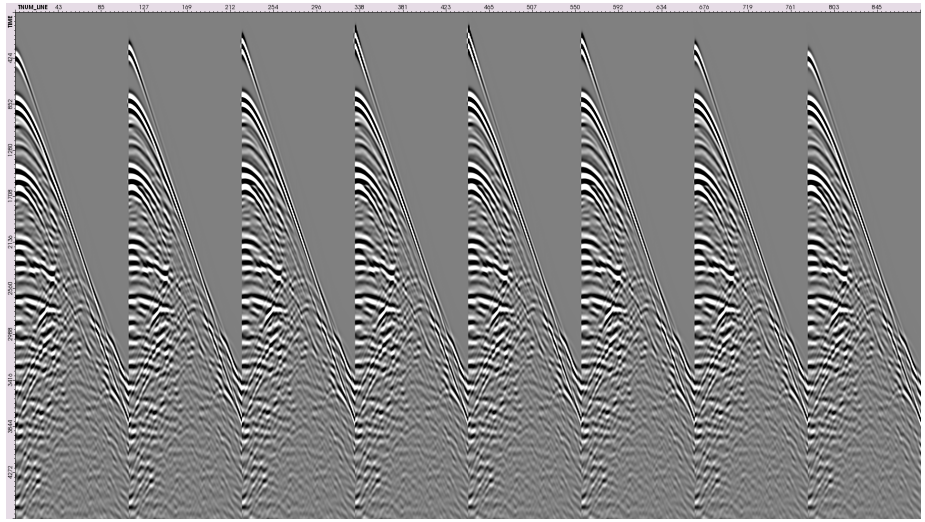
- FDorder regular/discontinuous: 12th order / 2nd-12th order
- timesteps regular/discontinuous: 11.1×10^3 / 8.6×10^3 (78%)
- model size regular/discontinuous: 155.7×10^6 / 66.8×10^6 GP (43%)

Benchmark Results

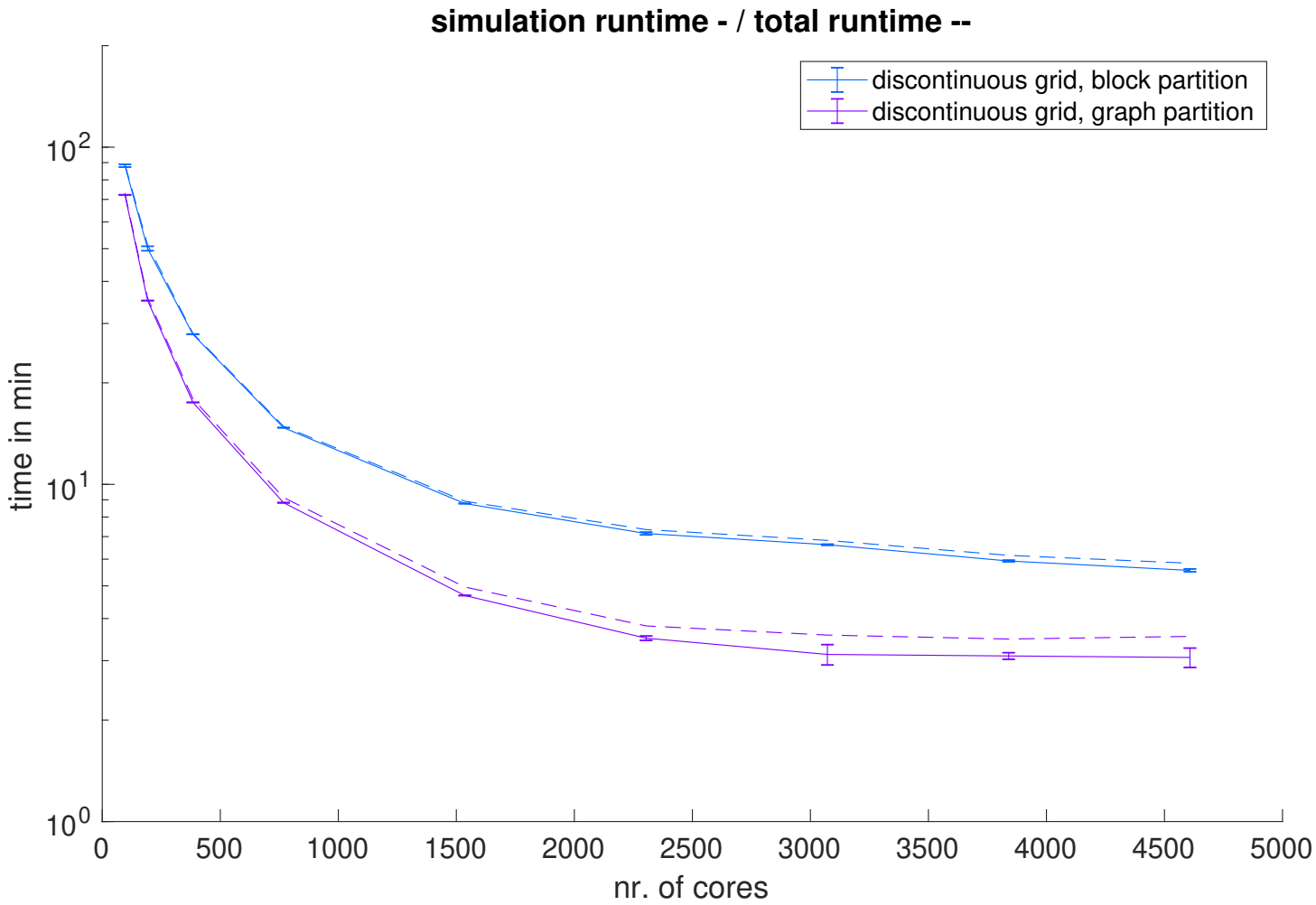
Regular Grid



Discontinuous Grid

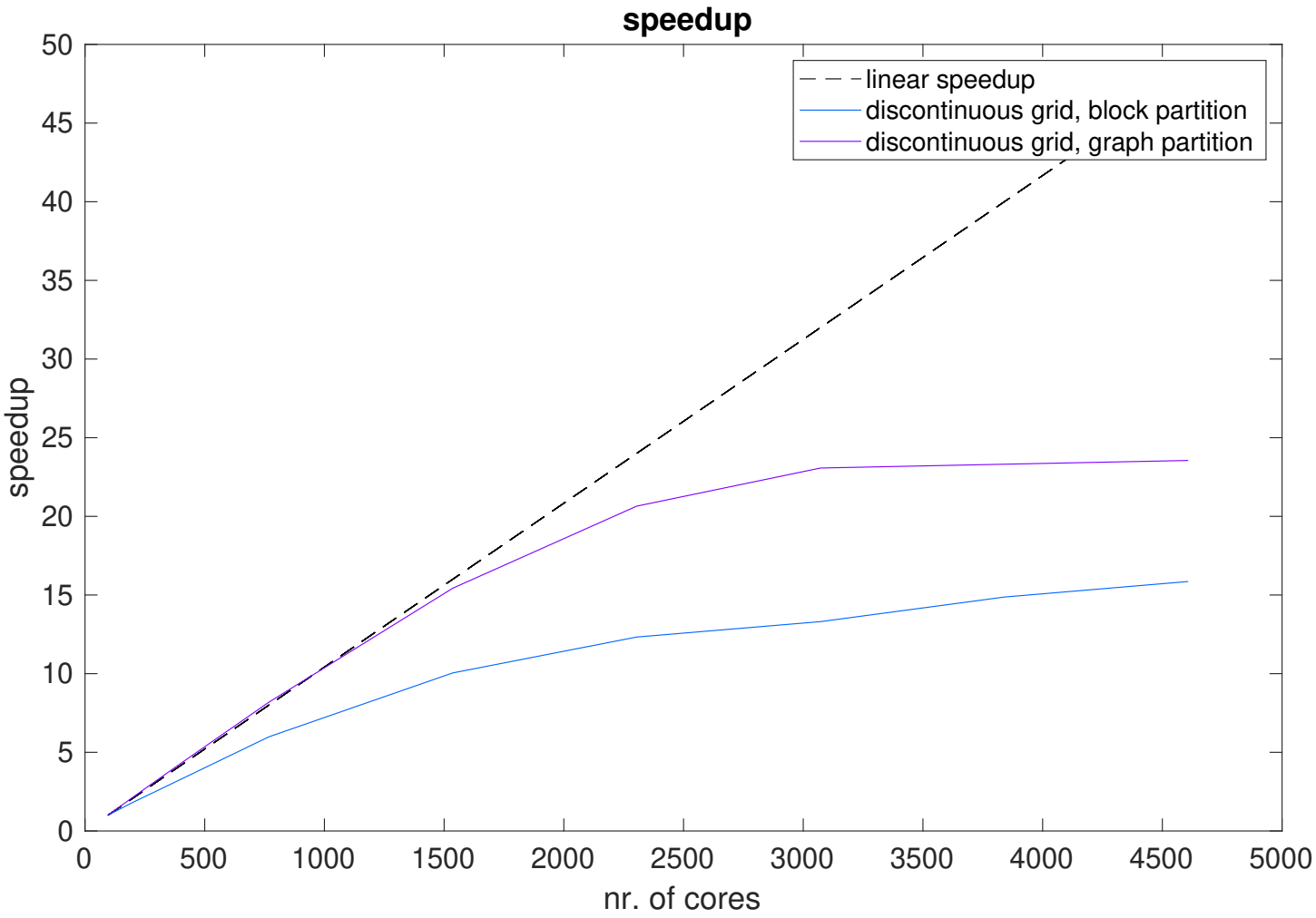


3D Benchmark CPU - Strong Scaling



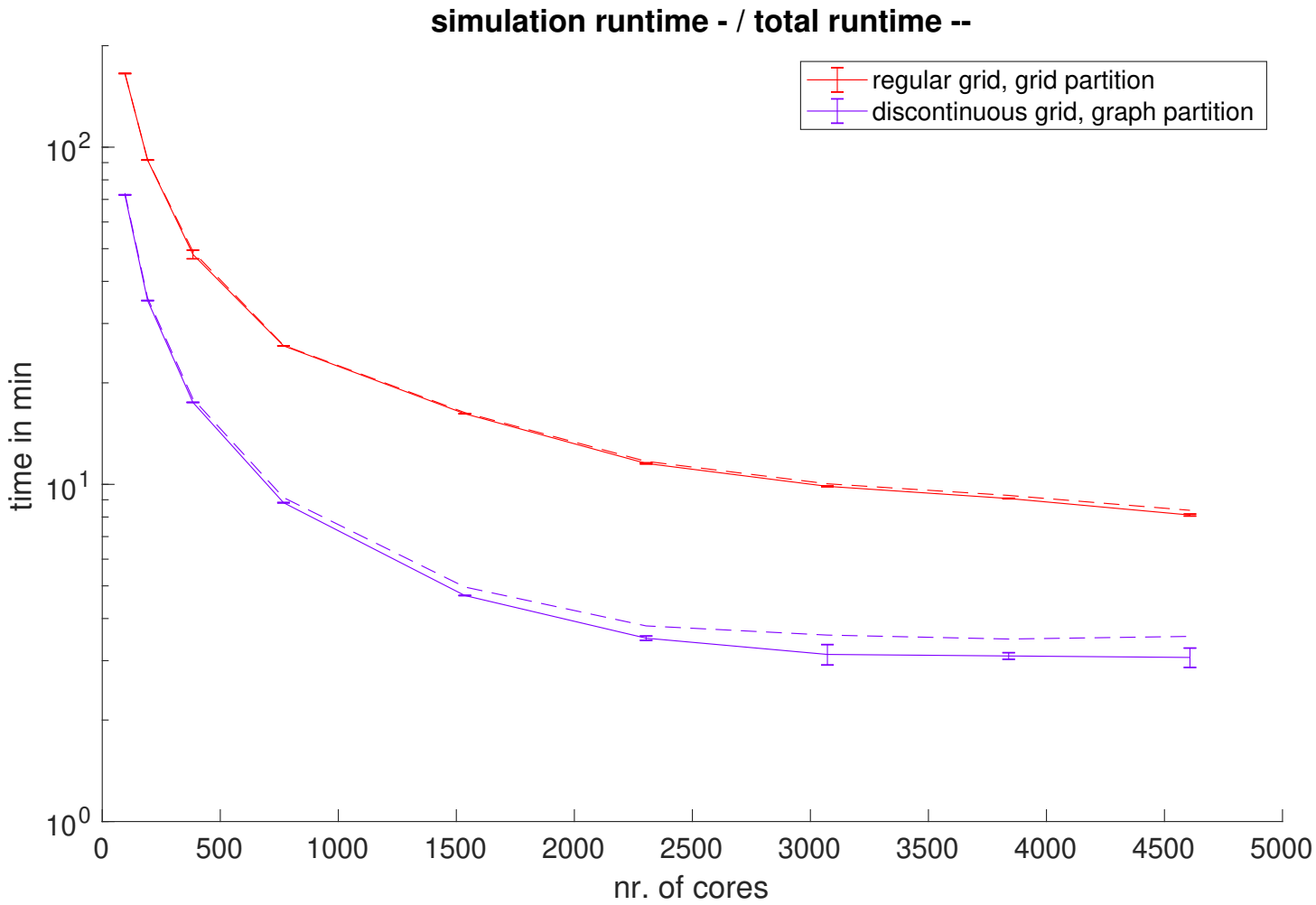
JUWELS @ Jülich Supercomputing Centre
CPU: Dual Intel Xeon Platinum (2x24 cores)

3D Benchmark CPU - Strong Scaling



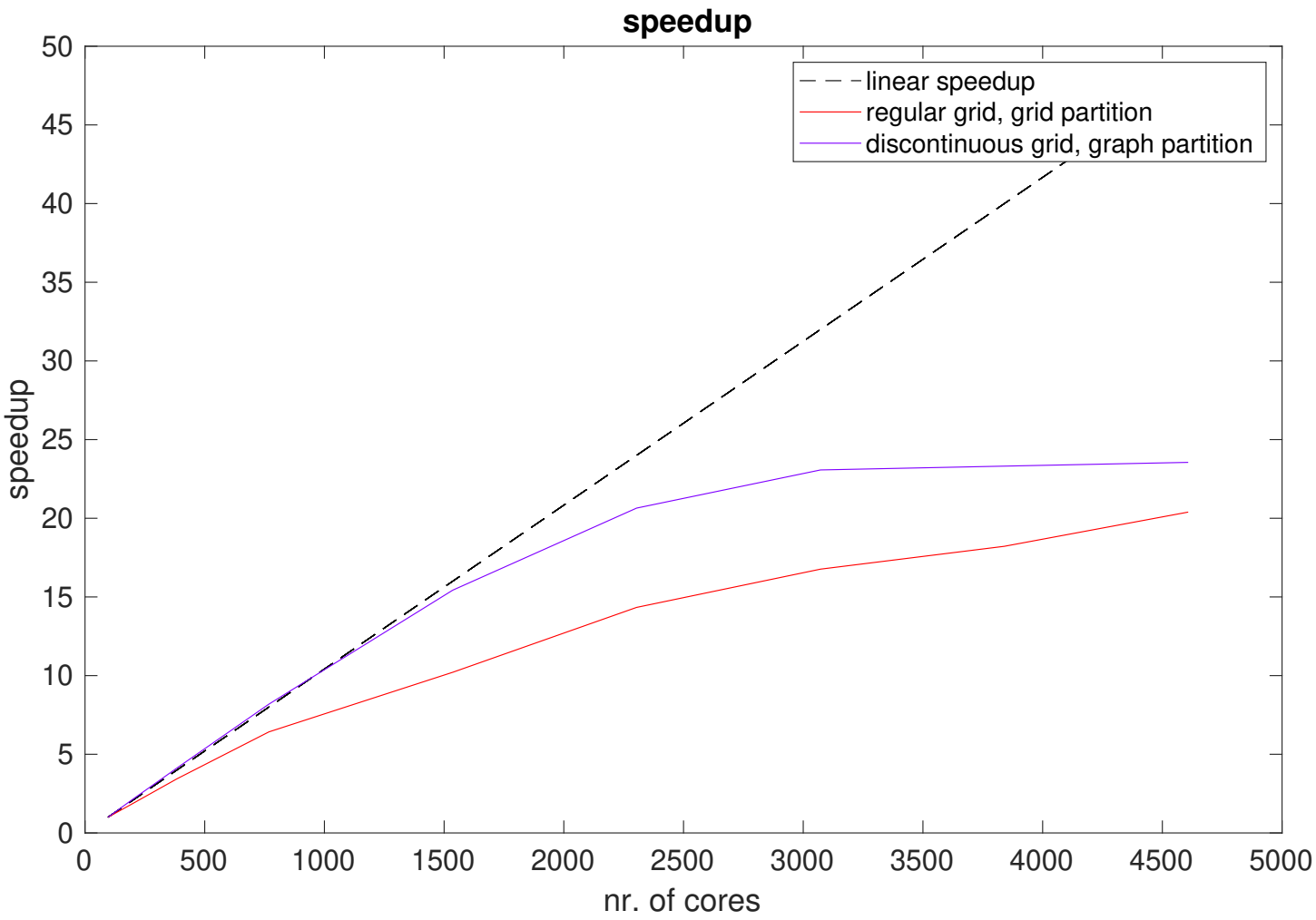
JUWELS @ Jülich Supercomputing Centre
Speedup relative to 2 Nodes (96 cores)

3D Benchmark CPU - Strong Scaling



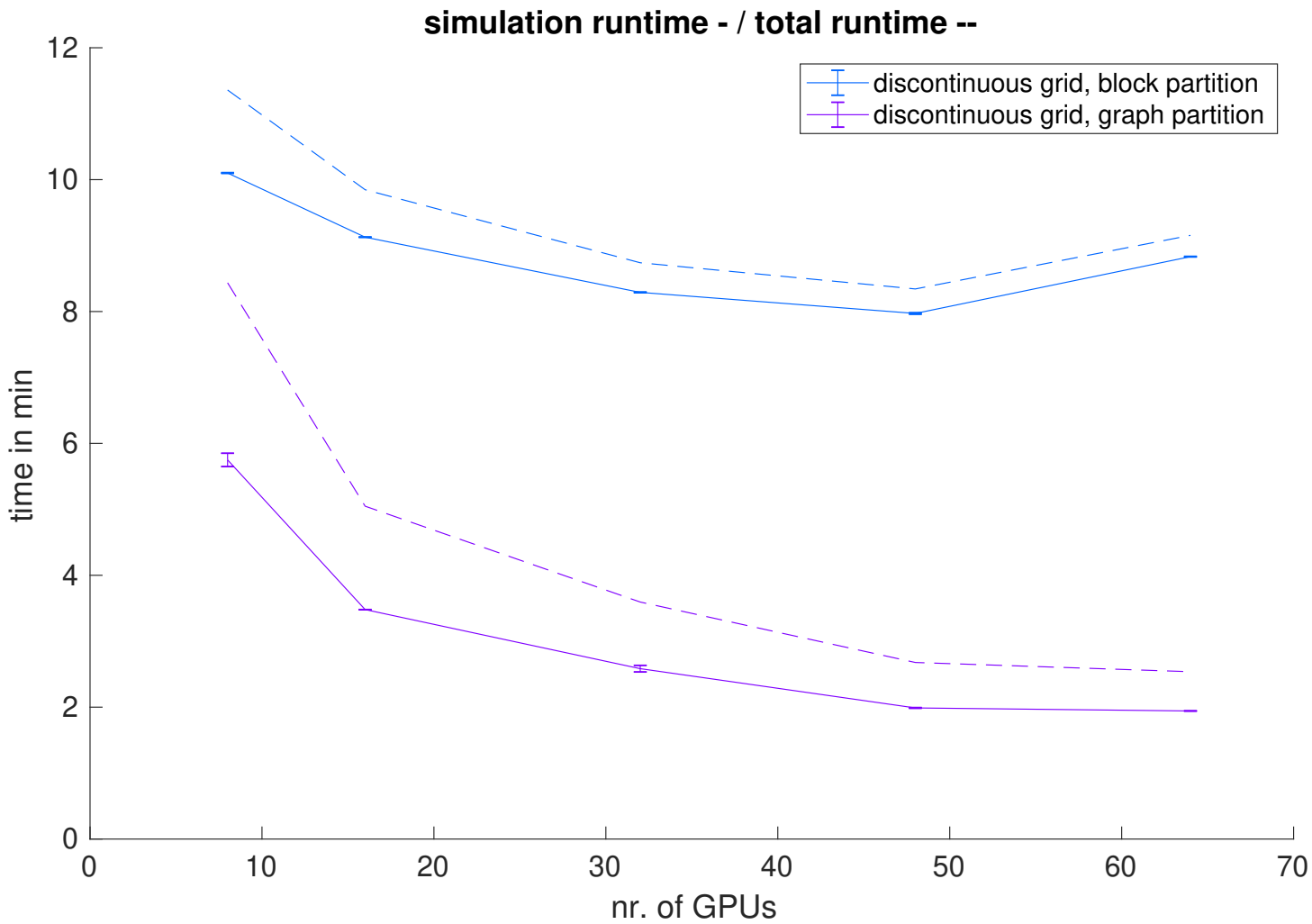
Speedup discontinuous grid:
2.5 - 3.3

3D Benchmark CPU - Strong Scaling



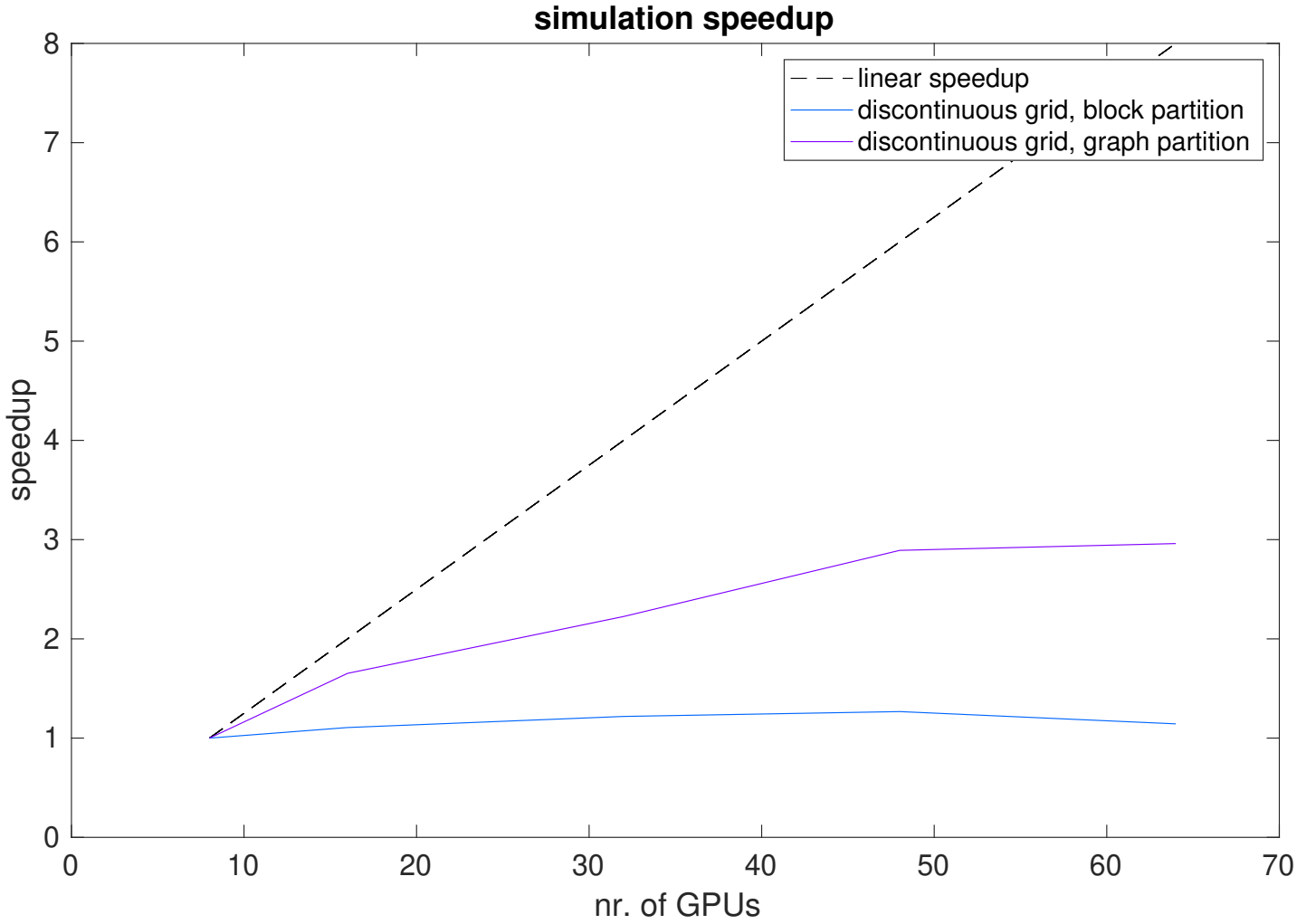
JUWELS @ Jülich Supercomputing Centre
Speedup relative to 2 Nodes (96 cores)

3D Benchmark GPU - Strong Scaling



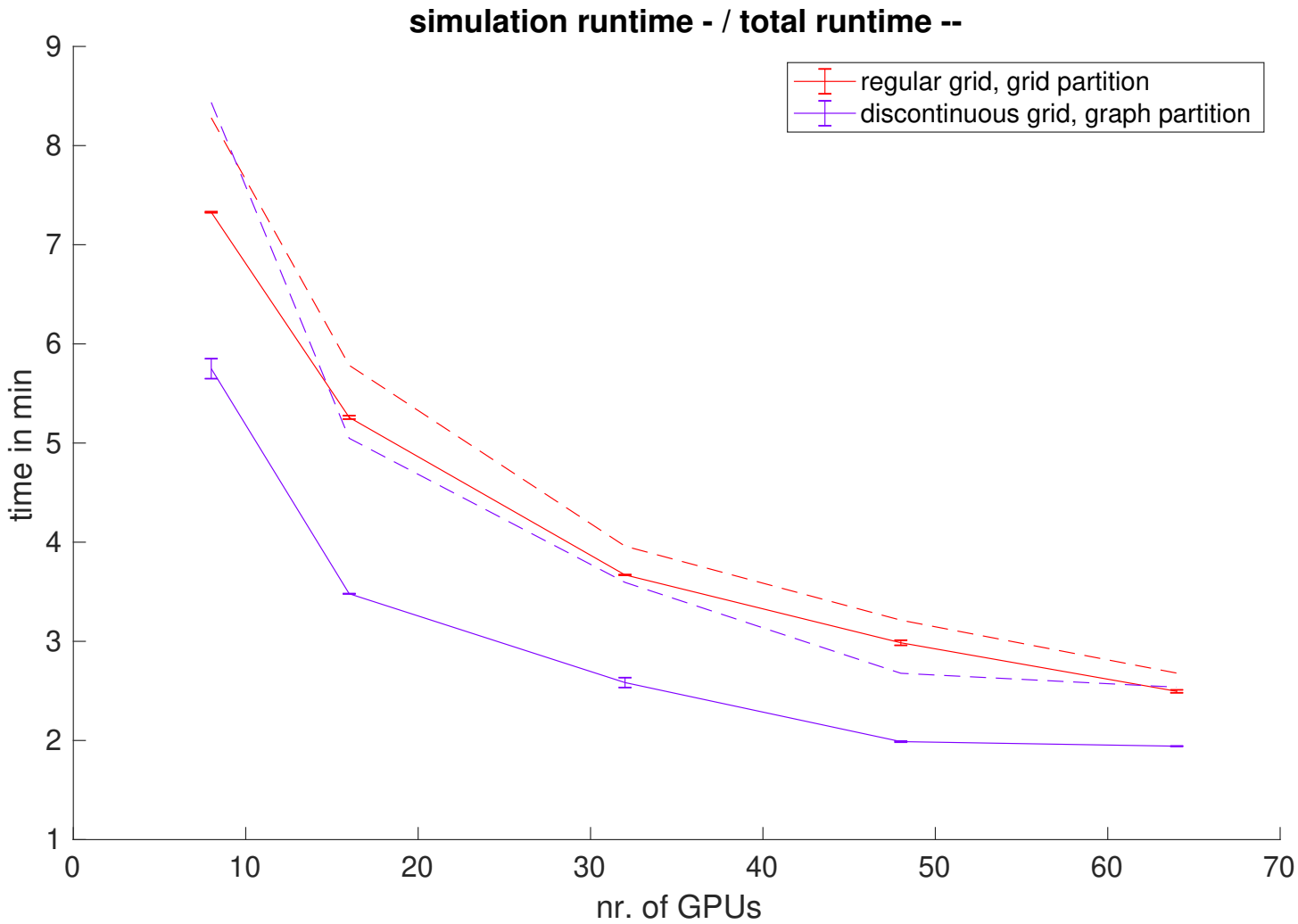
JUWELS @ Jülich Supercomputing Centre
GPU: Nvidia V100 GPU

3D Benchmark GPU - Strong Scaling



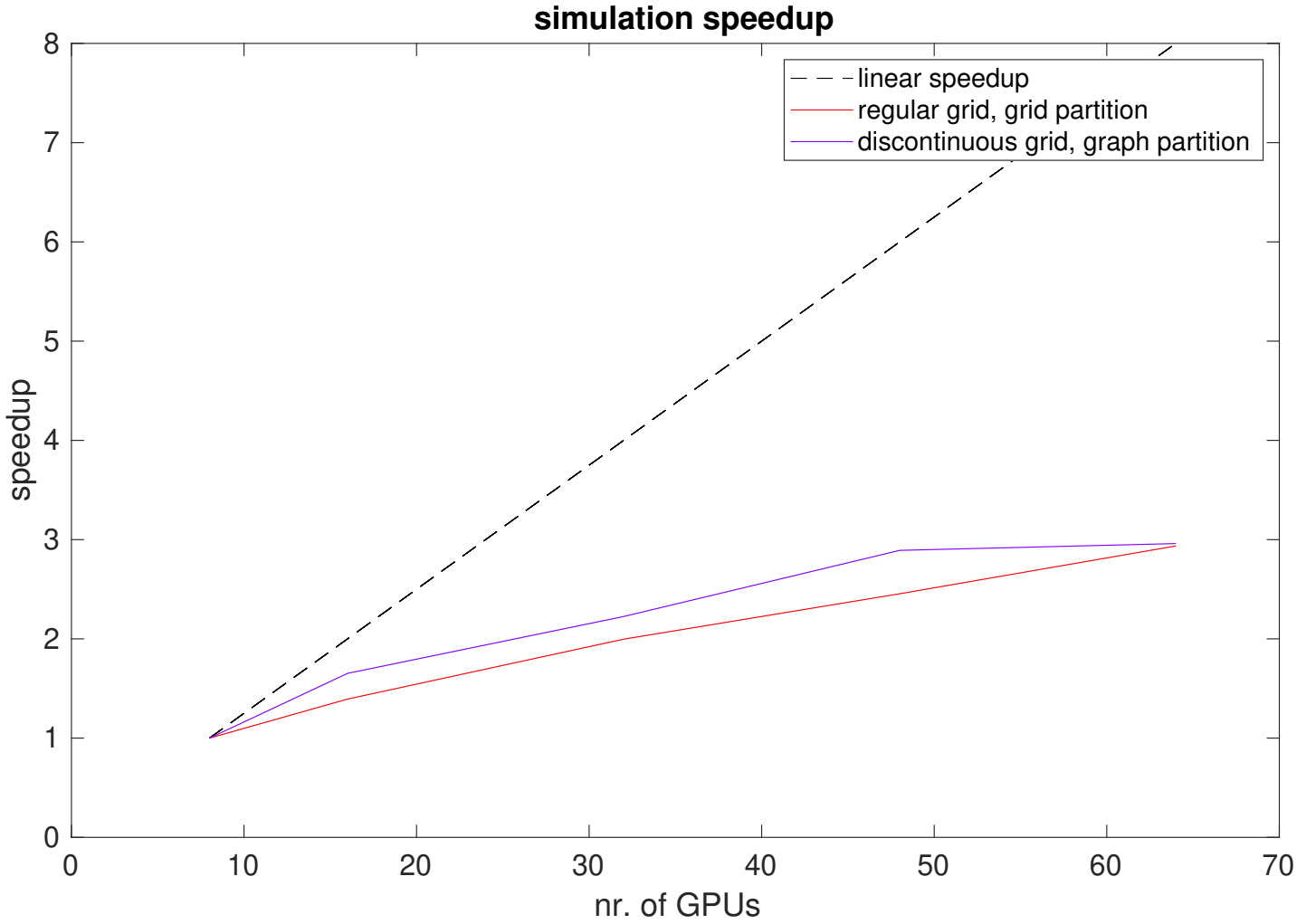
JUWELS @ Jülich Supercomputing Centre
Speedup relative to 2 Nodes (8 GPUs)

3D Benchmark GPU - Strong Scaling



Speedup discontinuous grid:
1.3 - 1.5

3D Benchmark GPU - Strong Scaling



JUWELS @ Jülich Supercomputing Centre
Speedup relative to 2 Nodes (8 GPUs)

Conclusion

- HPC ToolBox
  - 2D/3D acoustic / elastic /viscoelastic forward solver
  - discontinuous grid
  - graph partitioning
  - task parallelisation for heterogeneous architectures

Conclusion

- HPC ToolBox
  - 2D/3D acoustic / elastic /viscoelastic forward solver
  - discontinuous grid
  - graph partitioning
  - task parallelisation for heterogeneous architectures

Outlook:

- Full Waveform Inversion application on 3D "real" dataset

Outlook: seismic fieldwork

- Target: Ettlinger Line: trench built during the war of spanish succesion (18th century)

Acknowledgement

We greatfully acknowledge financial support from

Project WAVE

Federal Ministry of Education and Research

We gratefully acknowledge the Gauss Centre for Supercomputing e.V. (www.gauss-centre.eu) for funding this project by providing computing time through the John von Neumann Institute for Computing (NIC) on the GCS Supercomputer JUWELS at Jülich Supercomputing Centre (JSC).

Thank you for your attention

Tilman.Steinweg@kit.edu

http://www.gpi.kit.edu/