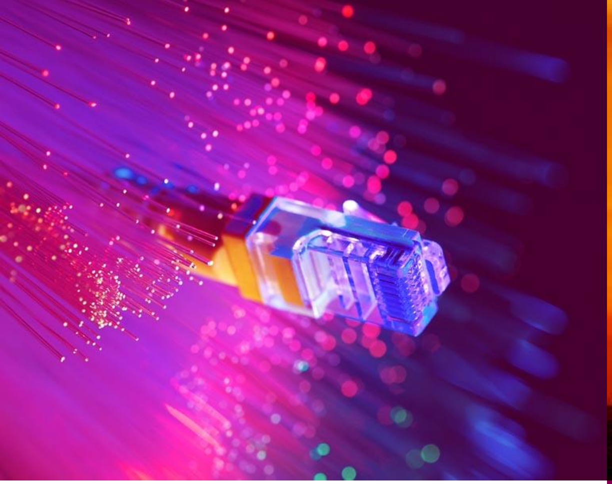


Elmer/Ice advanced workshop 2017

Grenoble, France

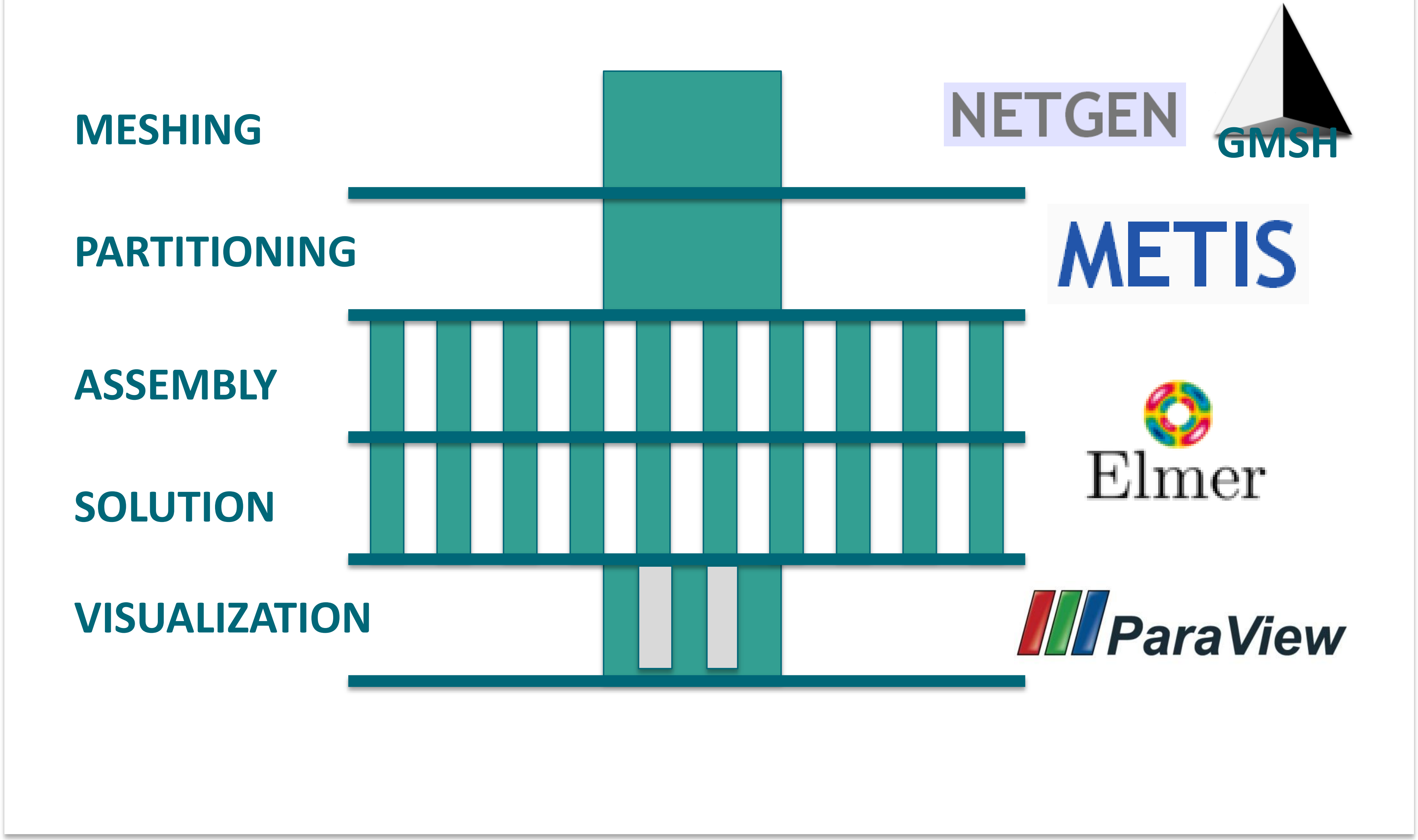
CSC – Finnish research, education, culture and public administration ICT knowledge center

Parallel strategies in Elmer/Ice

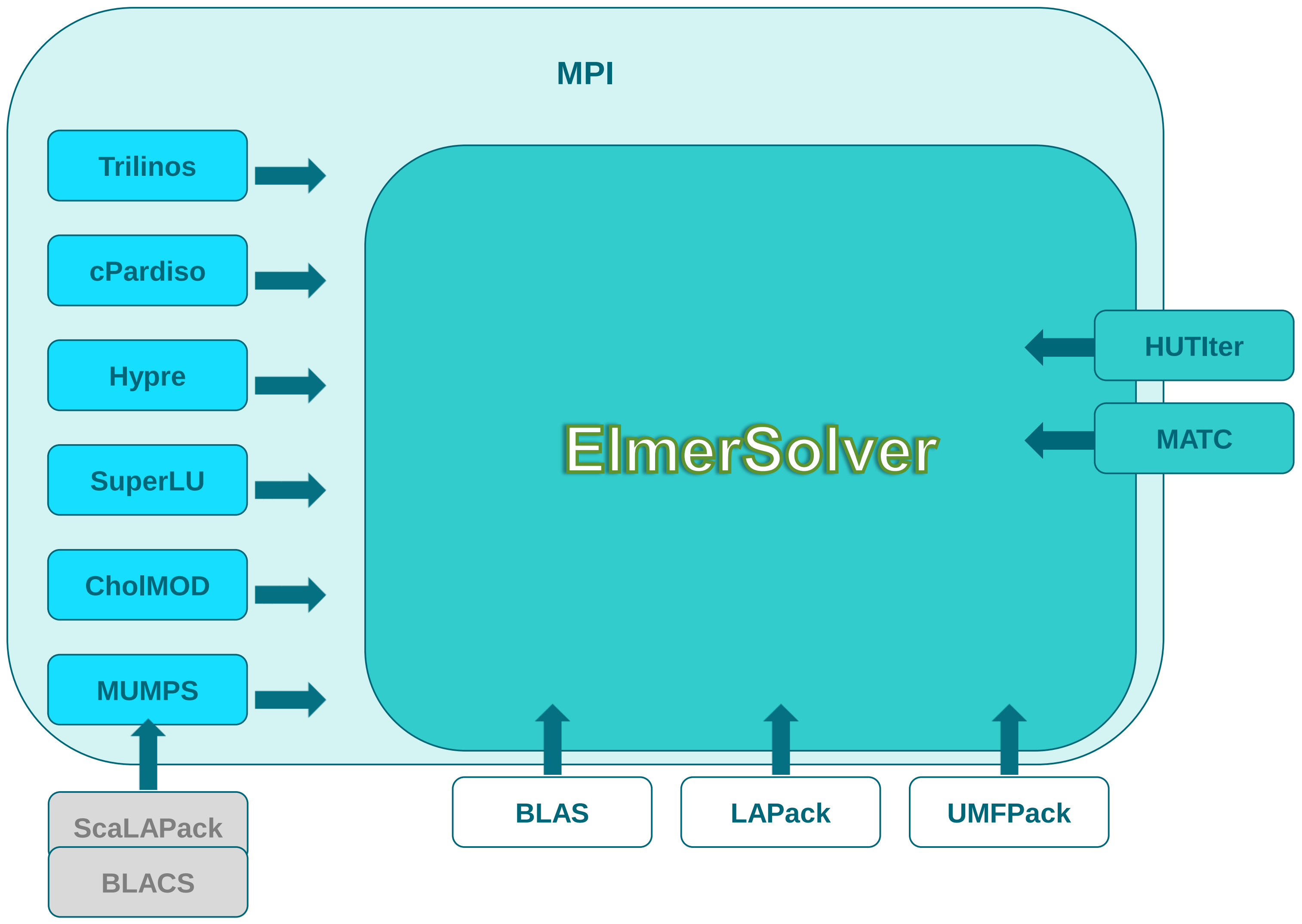
Including pre- and postprocessing in parallel

Thomas Zwinger

Parallel concept of Elmer

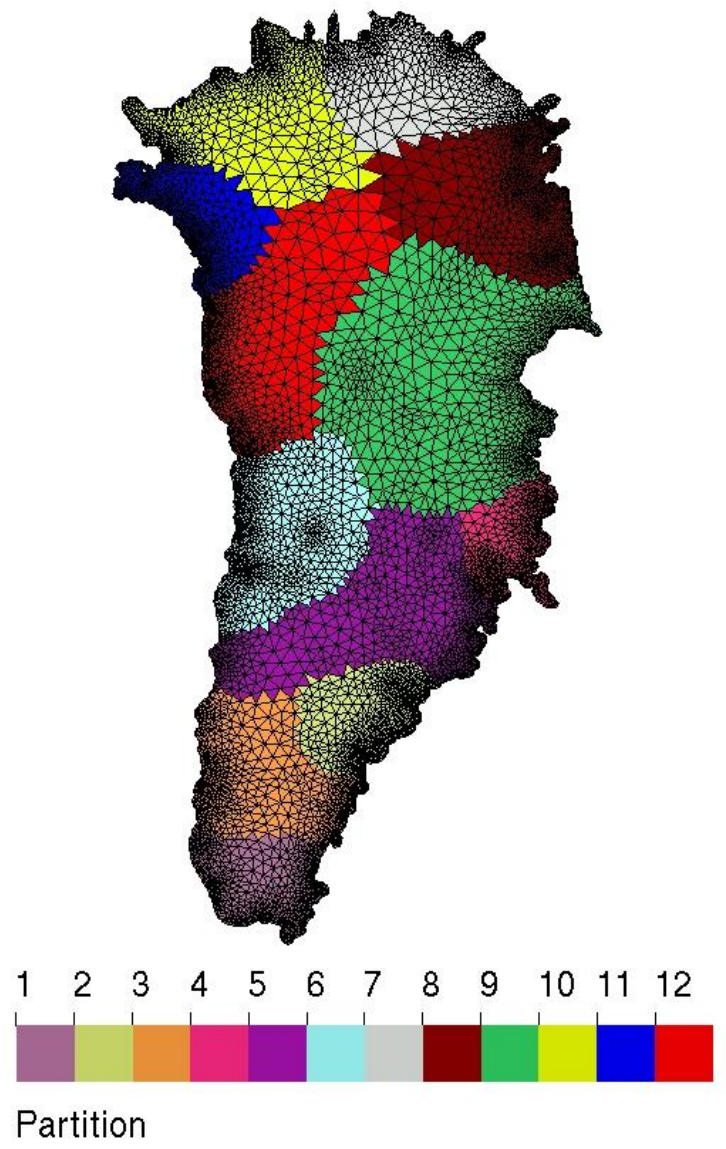


Parallel concept of Elmer

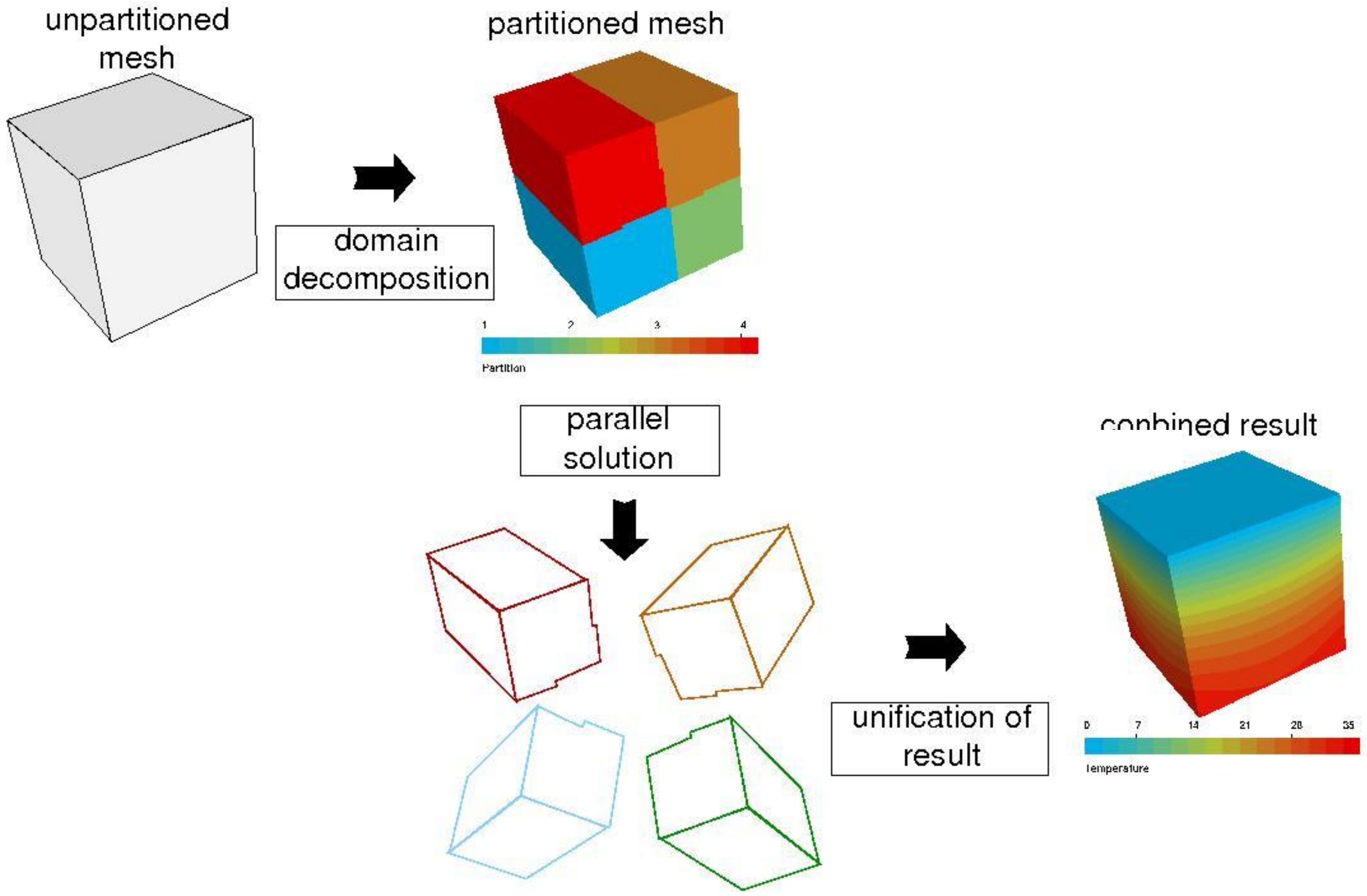


Parallel Concept of Elmer

- Domain decomposition
- Additional pre-processing step (splitting)
- Every domain is running its ”own” ElmerSolver
- Parallel process communication: Message Passing Interface (MPI)

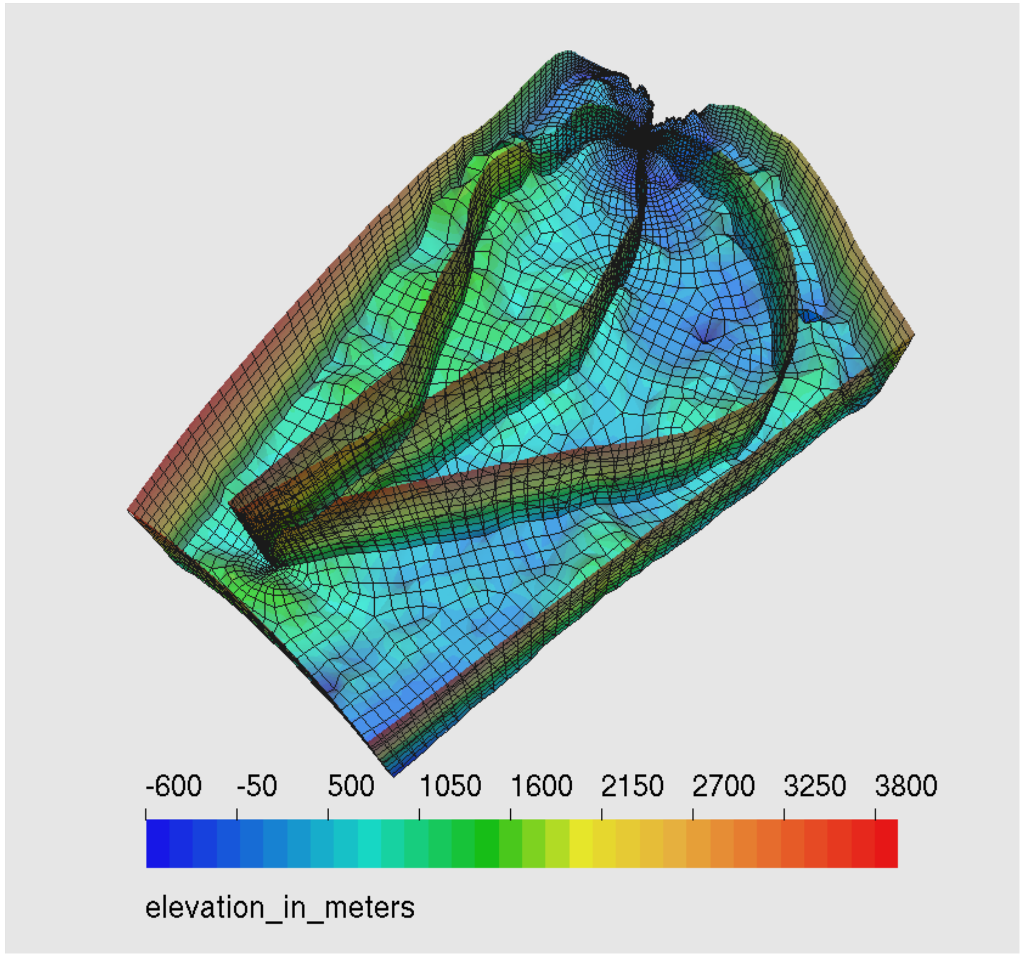


Parallel Concept of Elmer



Pre-processing in Elmer/Ice (focus parallel runs)

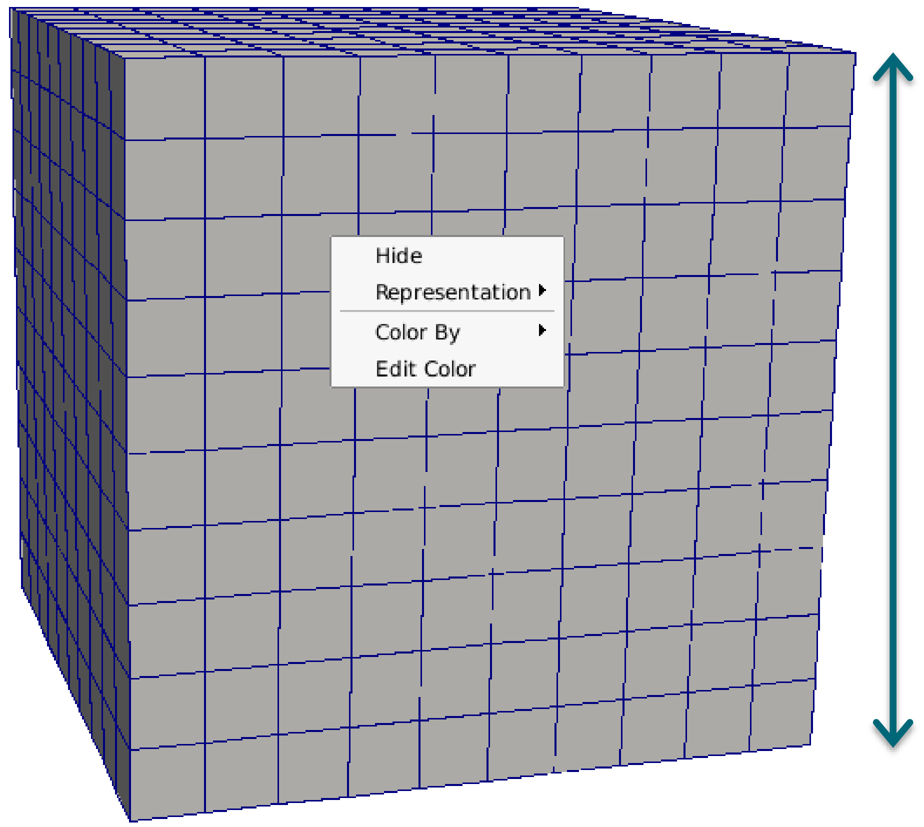
- Ice sheet = flat geometry
- Meshing tools (e.g., Gmsh) have difficulties with aspect ratios deviating from unity
- Creating footprint mesh in 2D
- Extruding it to 3D



Internal Extrusion

- Implemented as an internal strategy in Elmer (2013)
    - Juha, Peter & Rupert
- First partition a 2D mesh, then extrude into 3D
- Implemented also for partitioned meshes
    - Extruded lines belong to the same partition by construction!
- Deterministic, i.e. element and node numbering determined by the 2D mesh
    - Complexity: O(N)
- Neecessary, if using parallel runs with StructuredMeshMapper (see later)

Internal extrusion



```
Extruded Mesh Levels = 11
```

By default z in [0,1]

Else give:

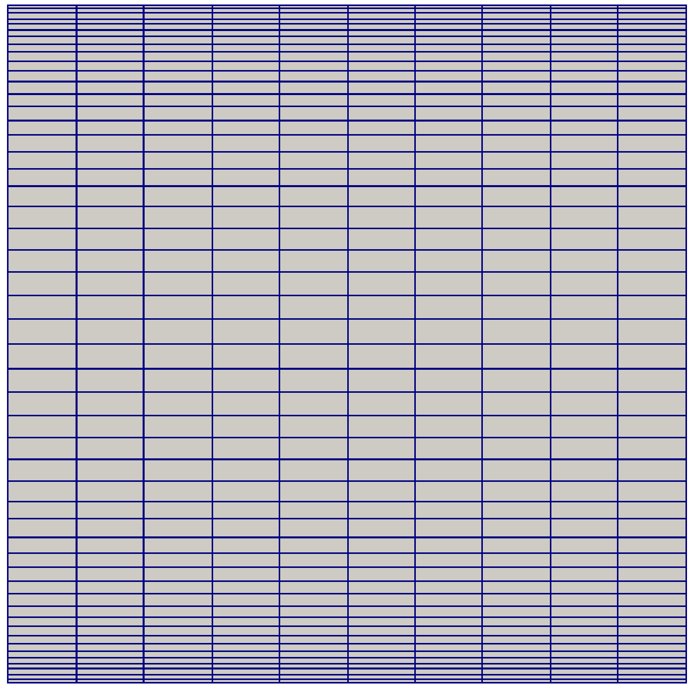
```
Extruded Max Coordinate = Real 10000.0
Extruded Min Coordinate = Real -100.0
```

Internal extrusion

```
Extruded Mesh Levels = 11
Extruded Mesh Density = Variable Coordinate 1
  Real MATC "0.2+sin(pi*tx)"
```

Any functional dependence is ok as long as it is positive!

The optimal division is found iteratively using Gauss-Seidel type of iteration and large variations make the iterations converge slowly.



Internal Extrusion

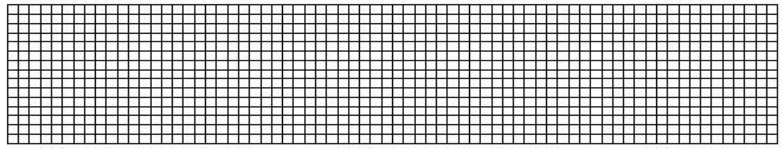
- **StructuredMesh Mapper** to impose geometry to a topological prism (3D)
- Define functions at bottom:

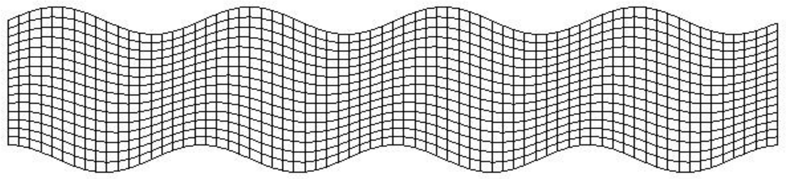
```
Bottom Surface = Variable
  "Coordinate 1"
    Real MATC "0.1*cos(5*tx)"
```

- And surface:

```
Bottom Surface = Variable
  "Coordinate 1"
```

```
Solver 2
  Equation = "MapCoordinate"
  Procedure = "StructuredMeshMapper"
"StructuredMeshMapper"
  Active Coordinate = Integer 2
End
```





ElmerSolver parallel

- Same executable: `ElmerSolver`
- Depending on platform/MPI: `mpirun -np` *`N`*

```
$ mpirun -np 6 ElmerSolver
```

- Needs information for different processes, which SIF to load: `ELMERSOLVER_STARTINFO`
- User defined functions/routines usually do not need special rewriting for MPI

Improvements on block preconditioners

Massive parallel Stokes problem

- Solving Stokes equations:

$$\nabla \cdot \tau(\mathbf{u}) - \nabla p + \rho \mathbf{g} = \mathbf{0}$$

$$\nabla \cdot \mathbf{u} = 0$$

- No connection between pressure and density – saddle point problem:

$$\begin{pmatrix} A & B^{\mathrm{T}} \\ B & 0 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{u} \\ p \end{pmatrix} = \begin{pmatrix} g \\ 0 \end{pmatrix}$$

Massive parallel Stokes problem

- Needs stabilization to avoid null-space:

$$\begin{pmatrix} A & B^{\mathrm{T}} \\ B & C \end{pmatrix} \cdot \begin{pmatrix} \mathbf{u} \\ p \end{pmatrix} = \begin{pmatrix} g \\ 0 \end{pmatrix}$$

- Yet, this matrix usually has a bad condition number:
  - Direct parallel solver (expensive): MUMPS, cPardiso (N^2 –algorithm)
  - Good pre-conditioner + iterative solvers ($N\ log(N)$ –algorithm)

$$\begin{pmatrix} A & B^{\mathrm{T}} \\ B & C \end{pmatrix} \rightarrow \mathbf{P} \cdot \begin{pmatrix} A & B^{\mathrm{T}} \\ B & C \end{pmatrix}$$

Block pre-conditioner

- Using following form: $$\mathbf{P} = \begin{pmatrix} A & B^{\mathrm{T}} \\ 0 & M \end{pmatrix}$$
  - M is a by viscosity weighted unit-matrix
- Yet, we need the inverse of $\mathbf{P}$, which requires exact solutions of linear systems of the blocks A and M
- Those systems are
  - way smaller sub-problems
  - better conditioned
  - solvable using effective linear Algebra (=Krylov subspace methods) to solve them

ParStokes

- The previously described method is implemented in the ParStokes-solver (Mika Malinen, CSC)
- ParStokes is nowadays part of the Elmer-distribution
  - No extra compilation is needed
  - Interface in SIF has been tremendously simplified
  - Still needs extra dummy-routines for providing the solution space for the preconditioning blocks (pressure,velocity); but their syntax is simplified. Basically only procedure call, if happy with default settings
  - Full documentation under ElmerModelsManual (chapter 24)
  - Eplained on elmerice-Wiki
  - Test-case: `elmerfem/fem/tests/PasStokes_ISMIP_HOM_A010`

(New) direct parallel solver: cPardiso

- Solver very similar to MUMPS
- Included in MKL (comes with Intel compiler suite) – not open source (to my knowledge)
- cPardiso can run mixed MPI/OpenMP codes, which Elmer is
  - Also OK to just use MPI side, like we can test in a session on Friday
- Simply include these lines into Solver section:

```
Linear System Solver = Direct
Linear System Direct Method = "cPardiso"
```

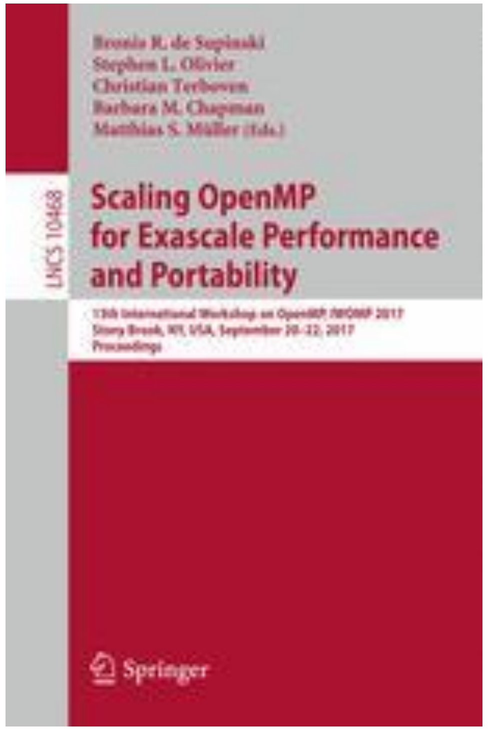
Future developments

Multi-threaded and SIMD version of advection/diffusion solver

- Modern CPU’s have 12+ cores
- Modern CPU’s have inproved vector units
- Multi-threading (OpenMP) might replace MPI for workstations
- Optimizing for that, using SIMD-instructions within the code

Optimization of Elmer(/Ice) using OpenMP Threading and SIMD

- Article in the proceedings which appear in the Lecture Notes in Computer Science (vol 10468), Springer
- The following slides are an excerpt of a presentation shown at this year's 13th International Workshop on OpenMP (IWOMP) in Stony Brook University, NY, USA
- Presentation permission granted by Intel Corp.


Bronis R. de Supinski
Stephen L. Olivier
Christian Terboven
Barbara M. Chapman
Matthias S. Müller (Eds.)

LNCS 10468

Scaling OpenMP for Exascale Performance and Portability

13th International Workshop on OpenMP, IWOMP 2017
Stony Brook, NY, USA, September 20–22, 2017
Proceedings

Springer

Byckling, M., J. Kataja, M. Klemm and T. Zwinger, 2017. *OpenMP SIMD Vectorization and Threading of the Elmer Finite Element Software*, **Proceedings 13th International Workshop on OpenMP, Springer Lecture Notes in Computer Sicnece**,123-137, doi:10.1007/978-3-319-65578-9_9

Suggested exercise

Bueler pofile parallel runs

- Based on the analytic equilibrium profile (SIA) from Bueler
- Training accounts + remote desktop on taito.csc.fi

1. Creating the footprint mesh
   - Partitioning of mesh
2. Setting up diagnostic parallel run using
   - Internal extrusion
   - Free surface imposed by external routine buelerprofile.f90
3. Running the case
   - With classic Navier-Stokes solver (MUMPS + cPardiso)
   - With ParStokes Solver

Bueler pofile parallel runs

- Two different kind of footprint meshes
  1. `footprint_bueler.geo` for small runs on laptop (ParStokes might not scale)
  2. `footprint_bueler_f.geo` for node-size runs on supercomputer (< 24 partitions)
  3. `footprint_bueler_ff.geo` for (massive) parallel runs
- Create mesh from gmsh:

```
$ gmsh -2 footprint_bueler.geo
```

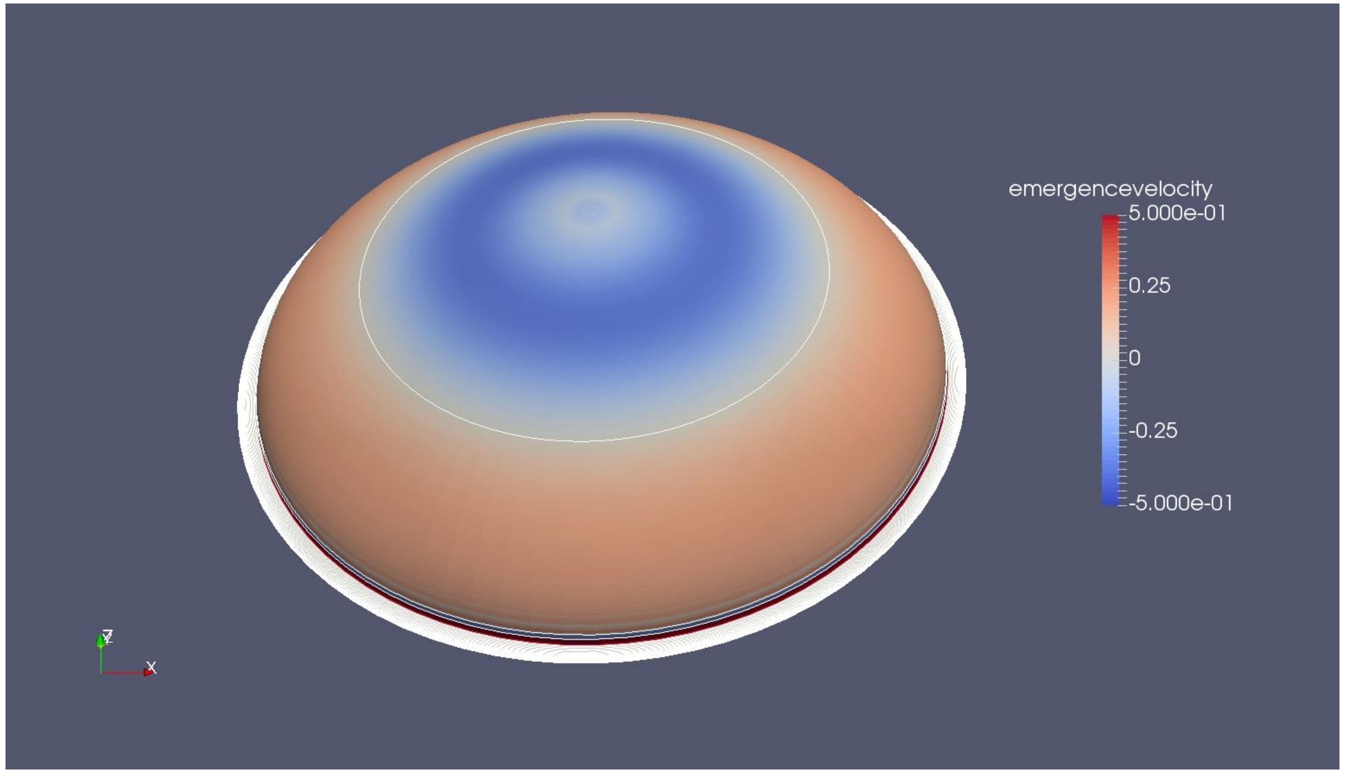
- Convert mesh into Elmer mesh

```
$ ElmerGrid 14 2 footprint_bueler.msh -autoclean
```

Add `-metis 24 4` at the end to create a 24 partition parallel mesh

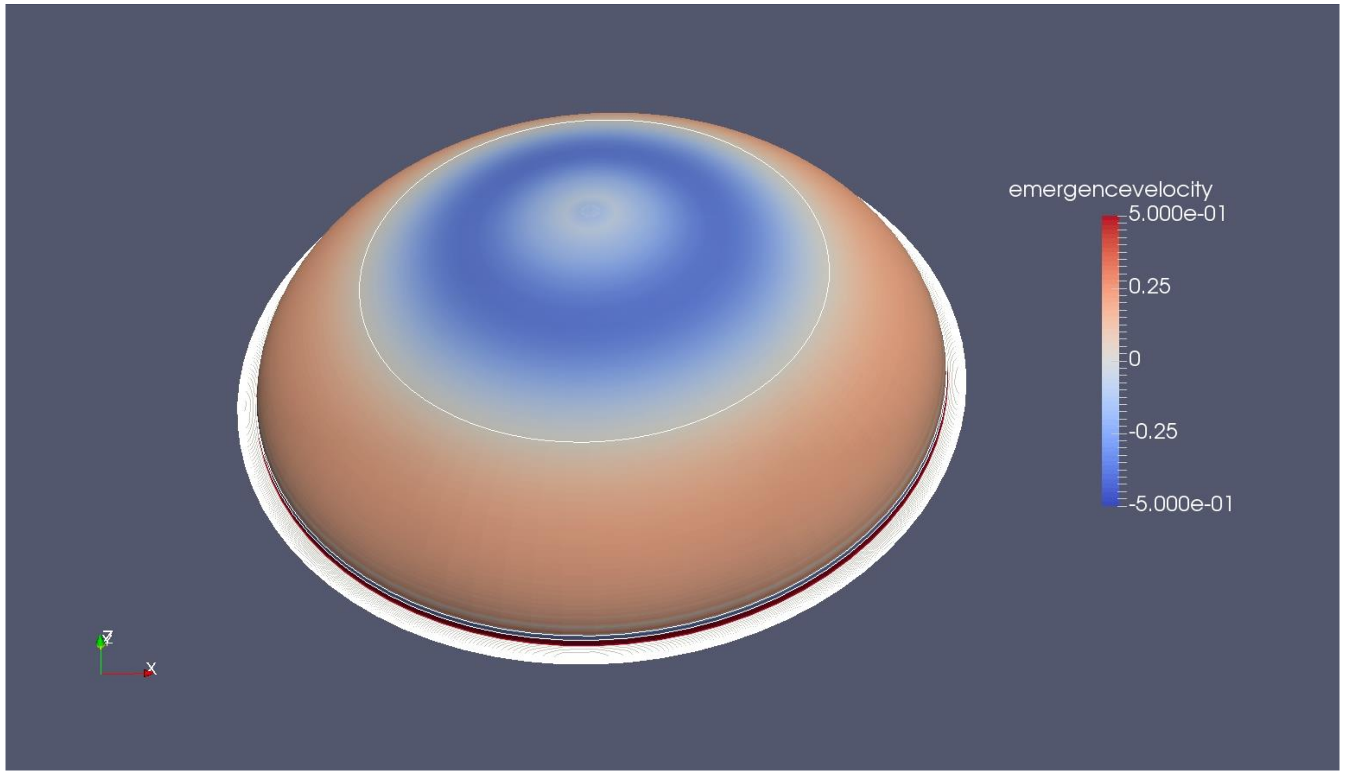
Bueler pofile parallel runs

- Using GetEmergenceVelocity after the Stokes solution to display the corresponding equilibrium accumulation-ablation function of the diagnostic solution
- Use footprint_



Bueler pofile parallel runs

MUMPS	cPardiso	ParStokes
> 30 mins	> 30 mins	132.66 s


emergencevelocity
5.000e-01
0.25
0
-0.25
-5.000e-01
Z
X