Elmer

Pre-processing utilities within ElmerSolver

ElmerTeam
CSC – IT Center for Science

Alternatives for increasing mesh resolution

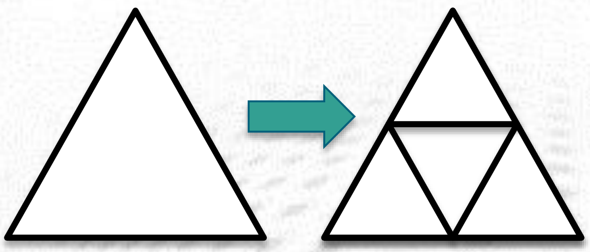
- Use of higher order nodal elements
  - Elmer supports 2nd to 4th order nodal elements
  - Unfortunately not all preprocessing steps are equally well supported for higher order elements
    - E.g. Netgen output supported only for linear elements
- Use of hierarhical p-element basis functions
  - Support up to 10th degree polynomials
  - In practice Element = p:2, or p:3
  - Not supported in all Solvers
- Mesh multiplication
  - Subdivision of elements by splitting

Note on bottle-necks in pre-processing

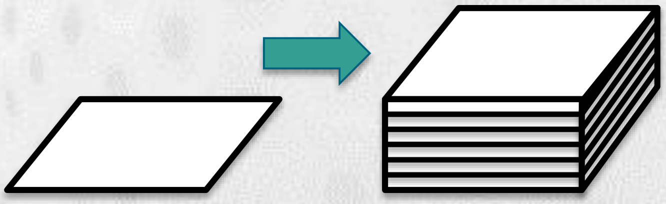
- After the solution pre-processing is typically the 2nd most time- and memory intensive task
- Mesh partitioning is typically less laborious than mesh generation
  - In Elmer we haven’t utilized parallel graph partitioning libraries (e.g. ParMetis)
- Serial mesh generation limited to around ~10 M elements
- Finalizing the mesh in parallel level within ElmerSolver may be used to eliminate this bottle-neck

Finalizing the mesh in parallel level

- First make a coarse mesh and partition it
- Bisection of existing elements in each direction
  - *2^DIM^n* -fold problem-size
  - Known as ”**Mesh Multiplication**”
  - Simple inheritance of mesh grading

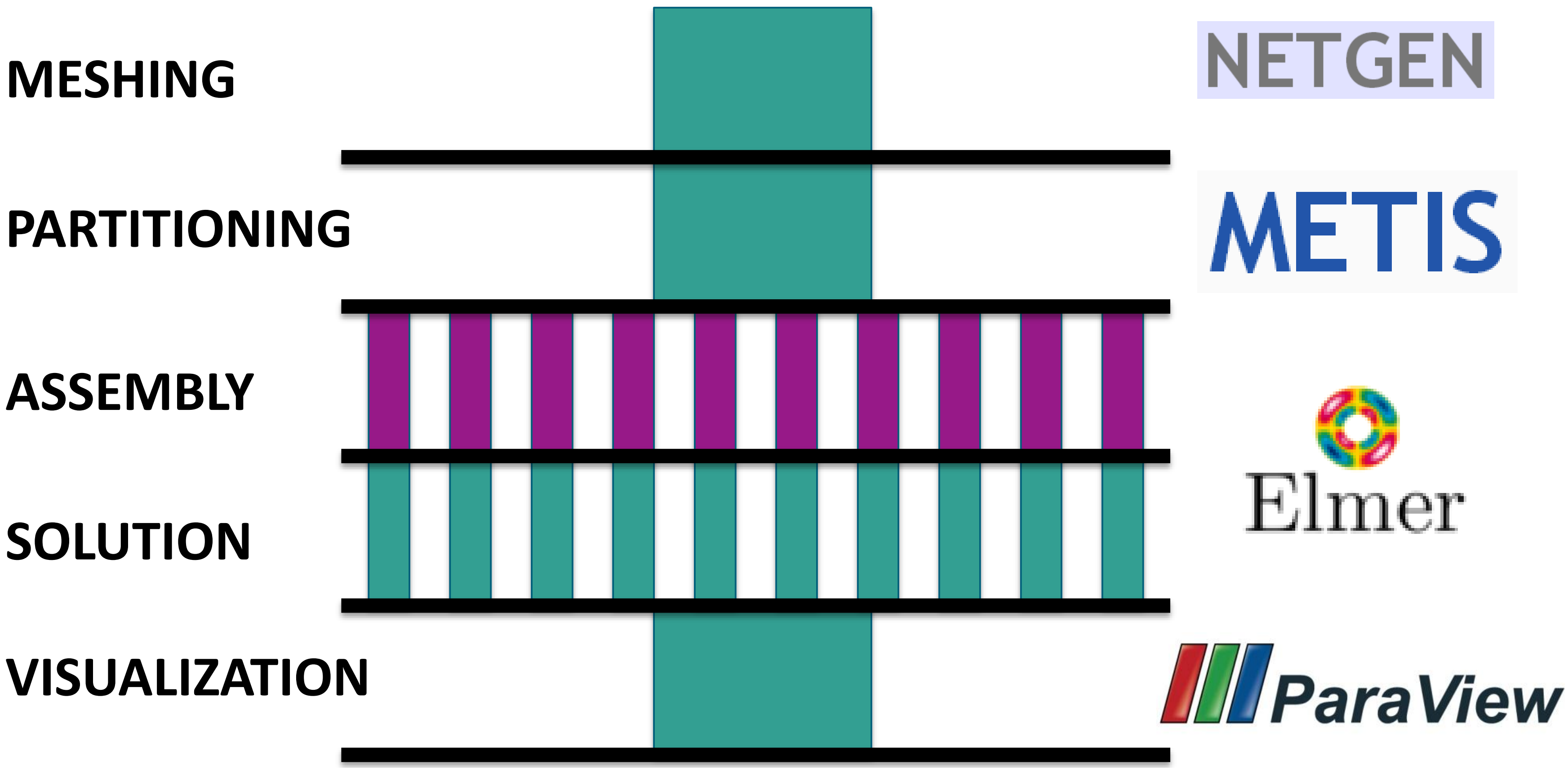


- Increase of element order (p-elements)
  - p-hierarchy enables the use of p-multigrid
- Extrusion of 2D layer into 3D for special cases
  - Example: Greenland Ice-sheet



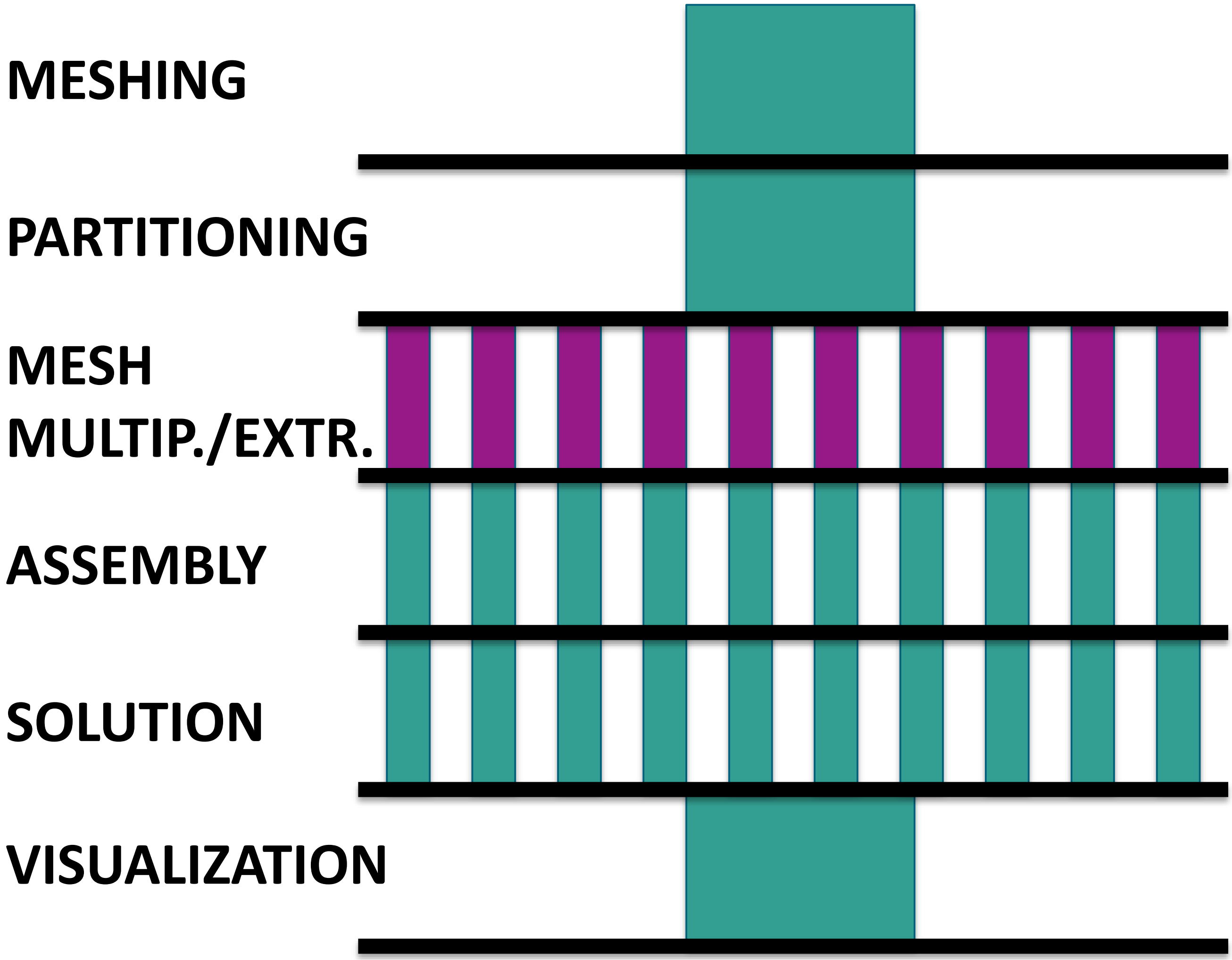
Standard parallel workflow

- Both assembly and solution is done in parallel using MPI
- Assembly is trivially parallel
- This is the basic parallel workflow used for Elmer



Parallel workflow

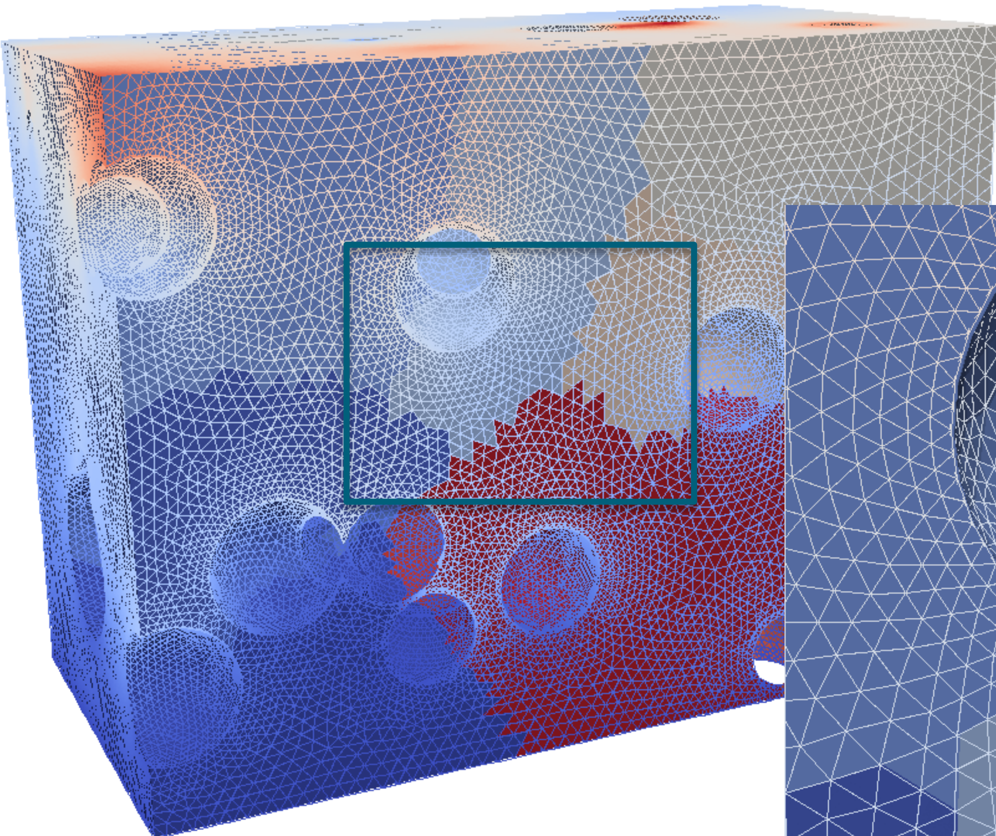
- Large meshes may be finilized at the parallel level

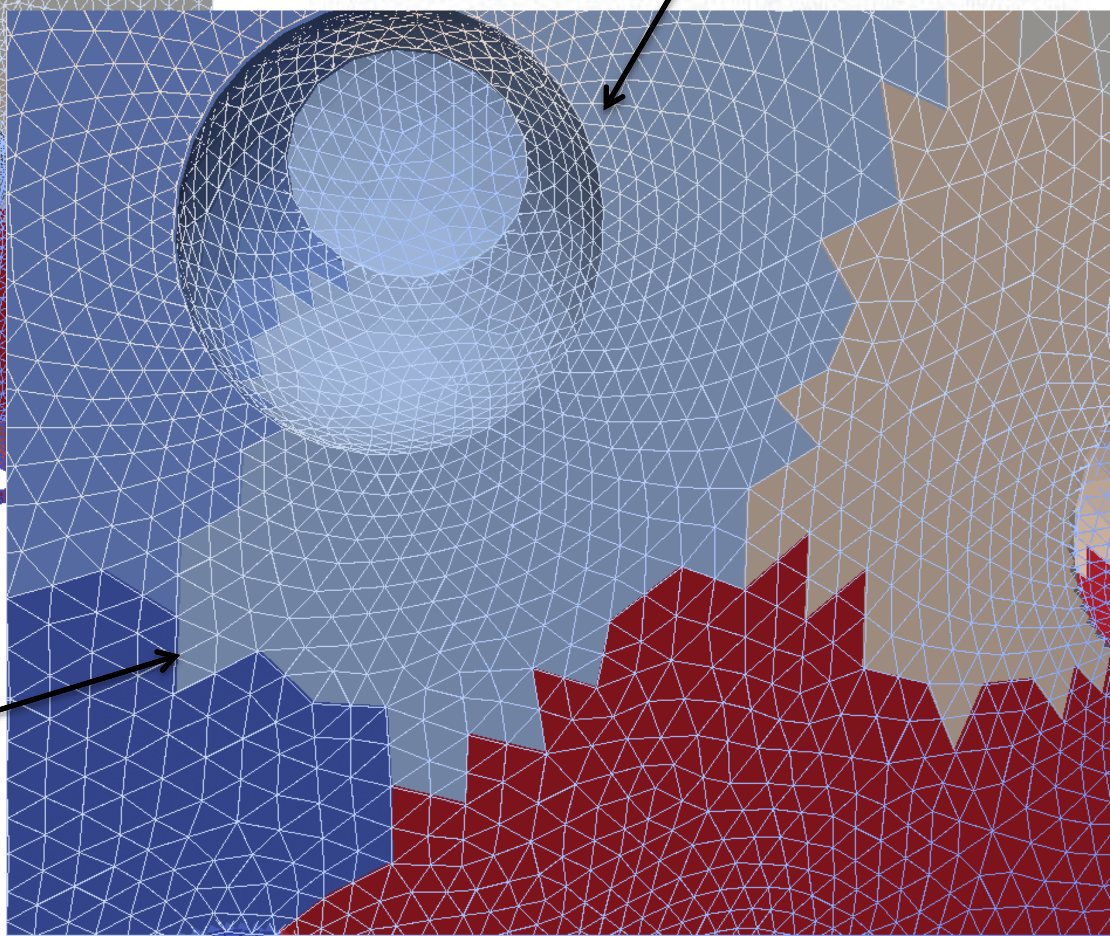


Mesh multiplication, example

`Mesh Levels = 2`

Mesh grading nicely preserved





Splitting effects visible in partition interfaces

Mesh Multiplication, example

- Implemented in Elmer as internal strategy ~2005
- Mesh multiplication was applied to two meshes
  - Mesh A: structured, 62500 hexahedrons
  - Mesh B: unstructured, 65689 tetrahedrons
- The CPU time used is negligible

Mesh	#splits	#elems	#procs	T_center (s)	T_graded (s)
A	2	4 M	12	0.469	0.769
	2	4 M	128	0.039	0.069
	3	32 M	128	0.310	0.549
B	2	4.20 M	12	0.369	
	2	4.20 M	128	0.019	
	3	33.63 M	128	0.201	

Limitations of mesh multiplication

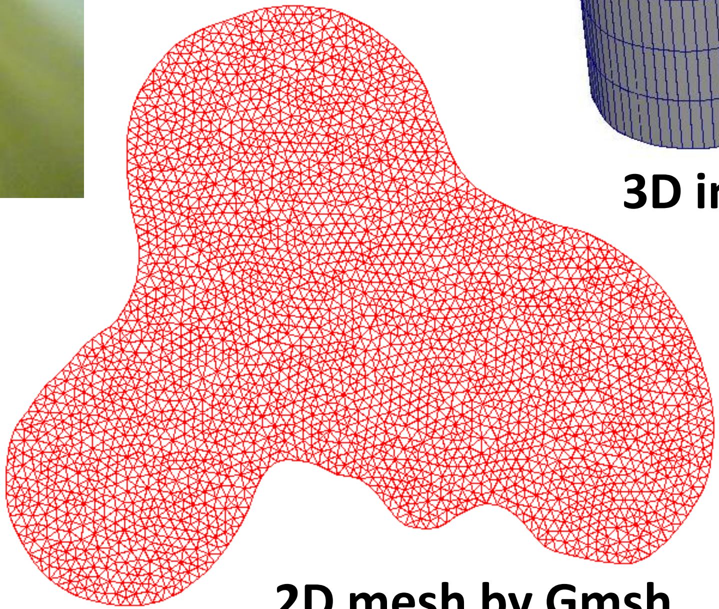
- Standard mesh multiplication does not increase geometric accuracy
  - Polygons retain their shape
  - Mesh multiplication could be made to honor boundary shapes but this is not currently done
- Optimal mesh grading difficult to achieve
  - The coarsest mesh level does not usually have sufficient information to implement fine level grading

Extrusion of partitioned meshes

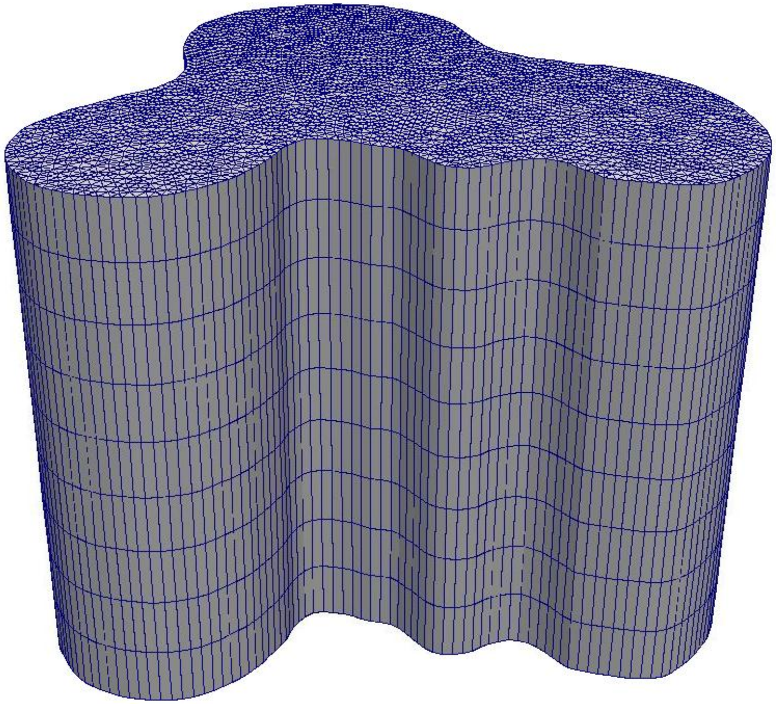
- Implemented as an internal strategy in ElmerSolver (2013)
- First partition a 2D mesh, then extrude into 3D
- Implemented also for partitioned meshes
  - Extruded lines belong to the same partition by construction!
- Deterministic, i.e. element and node numbering determined by the 2D mesh
  - Complexity: O(N)
- There are many problems of practical problems where the mesh extrusion of a initial 2D mesh provides a good solution
  - One such field is glasiology where glaciers are thin, yet the 2D approach is not always sufficient in accurary

Internal extrusion example: Aalto Vase

Design Alvar
Aalto, 1936



2D mesh by Gmsh



3D internally extruded mesh

Utilizing extruded structures

- If the mesh is extruded it makes sense to utilize this fact also in later steps
  - Operators in the extruded directions
  - Combination of full 3D and 2D higher order models
- Tailored solvers that assume extruded structure
  - `StructuredMeshMapper`
  - `StructuredProjectToPlane`
  - `StructuredFlowLine`
- No assumptions on the numbering of the nodes is needed

Deforming meshes

- Meshes may be internally deformed
- **MeshUpdate** solver uses linear elasticity to deform the mesh
- **RigidMeshMapper** uses rigid deformations and their smooth transition to deform the mesh
- Deforming meshes have number of uses
  - Deforming structures in multiphysics simultion
    - E.g. fluid-structure interaction
  - Rotating & sliding structures
  - Geometry optimization
    - Mesh topology remains unchanged

Conclusions on internal meshing features

- There are number of ways to increase the resolution of solution within ElmerSolver that eliminate meshing bottle-necks
  - For complex cases these may still be unsatisfactory
- Internal mesh deformation may be used to solve complex problems without a need for remeshing
  - Large deformations may be problematic and topological changes impossible