A Parallel Linear Solver Exploiting the Physical Properties of the Underlying Mechanical Problem

Kees Vuik
Delft University of Technology
http://ta.twi.tudelft.nl/nw/users/vuik/

SIAM Conference on Computational Science and Engineering,
March 14-18, 2015, in Salt Lake City, Utah, USA

March 18, 2015

Contents:

Introduction

Geomechanical problems are hard

- Geomechanical problems typically involve large volumes of soil/rock and various structural components.
- Non-linear finite element models are used to compute the deformation field.
- Difficult to solve because large variations in stiffness and many degrees of freedom.

Requirements

The parallel iterative solver had to:

- be able to solve the same systems of equations as the original iterative solver;
- have a similar performance on a single processor as the original solver;
- achieve a significant parallel speedup on ***capable*** machines for relatively large problems;

The parallel solver also had to:

- be a drop-in replacement of the original solver;
- hide all aspects concerning parallel computing from the end users;
- and refrain from relying on (expensive) third-party software components.

Parallelisation strategy

Domain decomposition

Two methods to create the sub-domains and the corresponding sub-domain matrices are:

- the traditional, element-based method;
- an alternative, node-based method.

Element-based method

Advantage: fits naturally in the traditional domain decomposition framework.

Disadvantage: involves significant modifications in the application implement the domain-wise assembly procedure.

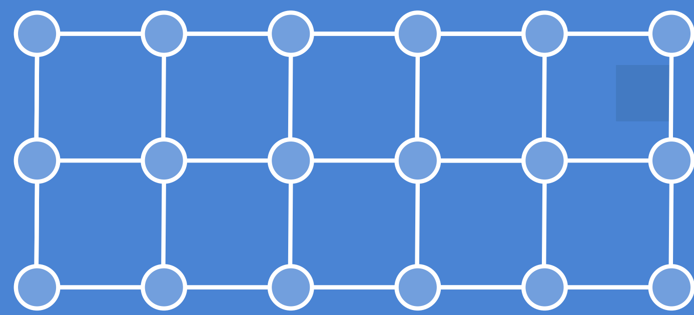
Node-based method

Advantage: sub-domain matrices can be created directly from the global stiffness matrix; no need to implement a modified assembly procedure.

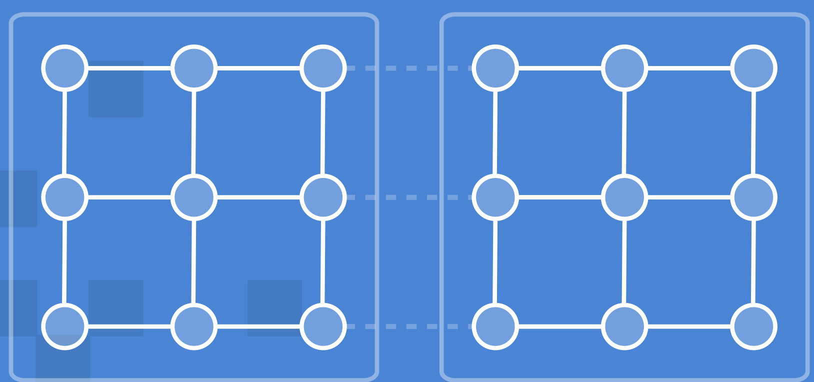
Linear constraints (***tyings***) are handled automatically because they have already been taken into account during the assembly of the global matrix.

First step: partition the nodes without overlap.

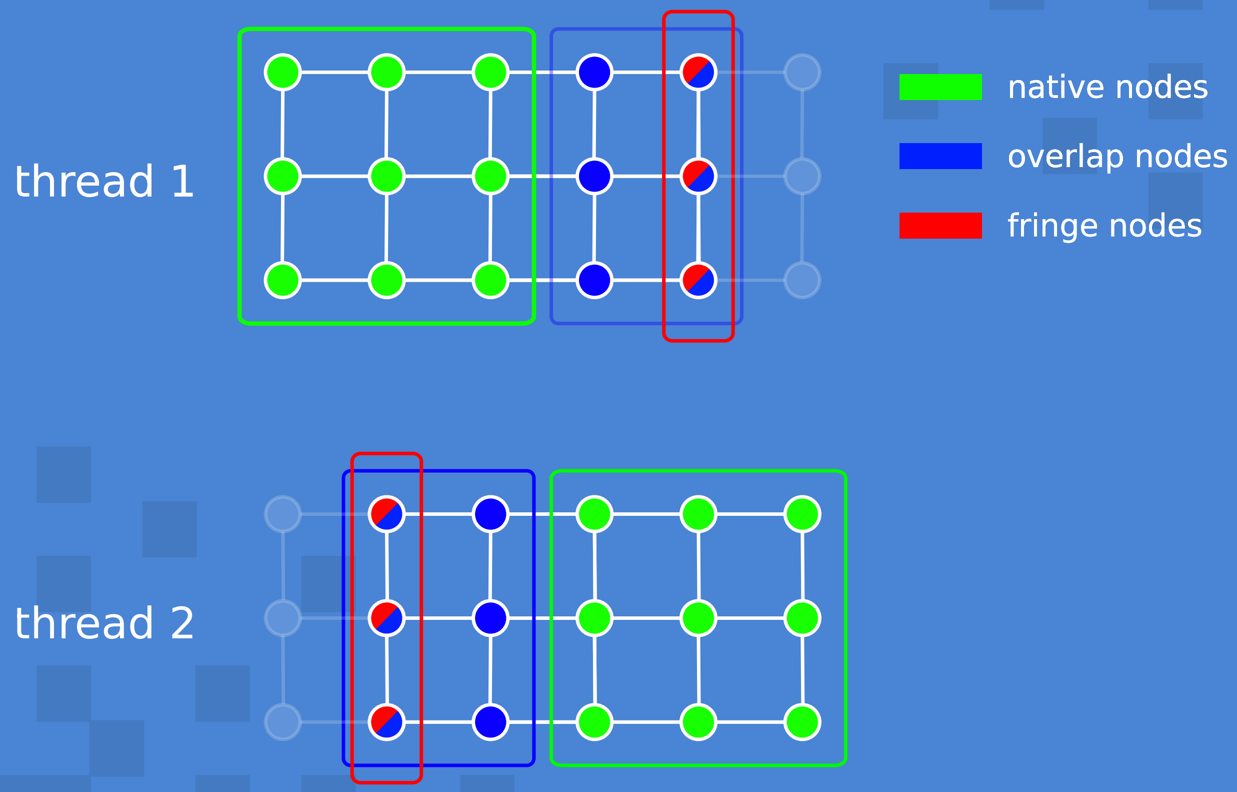
Mesh



Unique node partition = native nodes

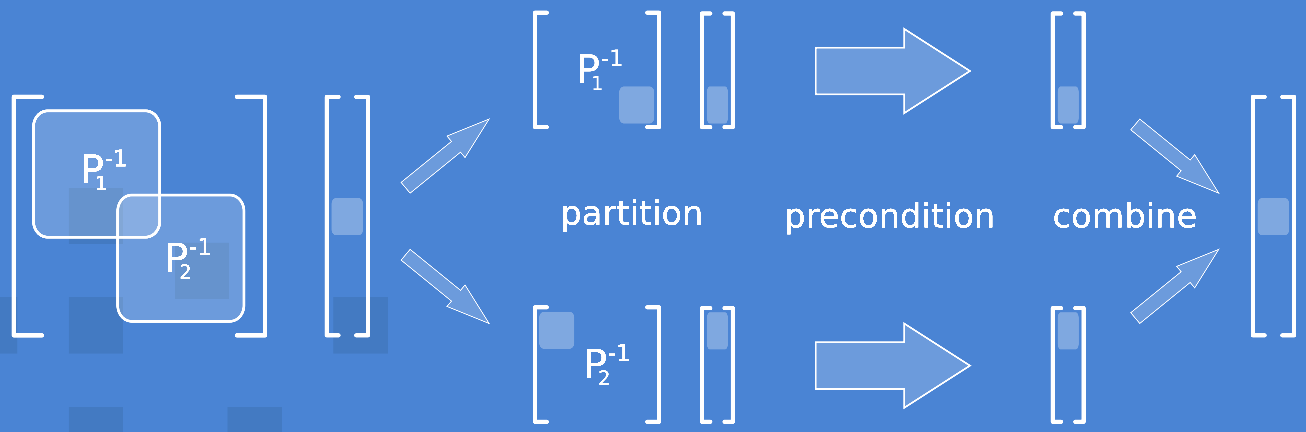


Second step: add overlap nodes.

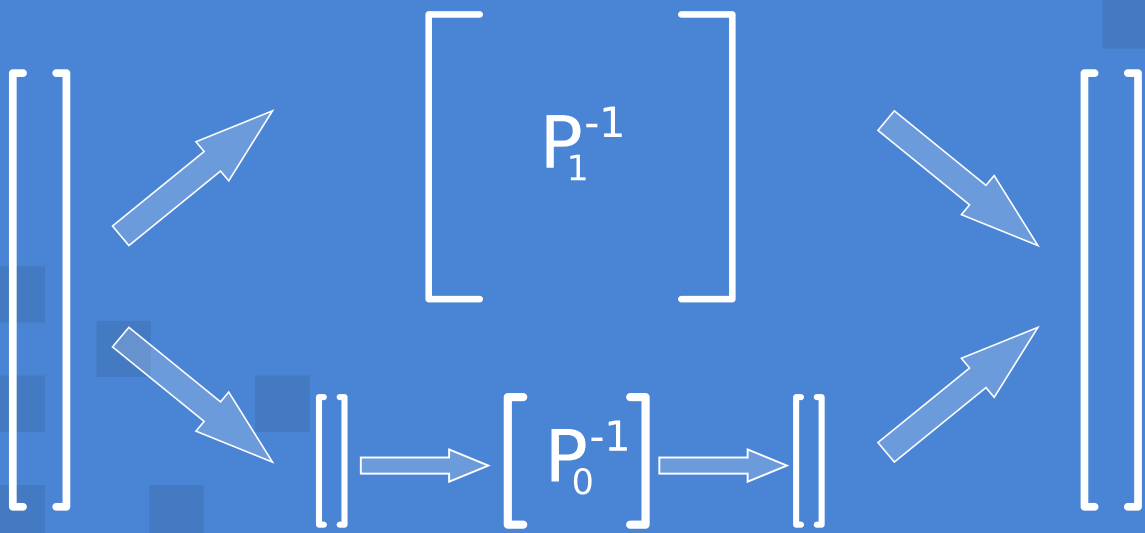


Preconditioner

Application of the preconditioner to a vector:



Augmenting the preconditioning with a coarse grid preconditioner (use the Rigid Body Modes of each sub-domain):



First results are mixed

- Good performance and speed up for uniform models.

  Coarse grid perconditioner works well.

- Worse performance than original solver for non-uniform models.
- Reason: large variations in material stiffness within sub-domains.

Physics-based partitioning

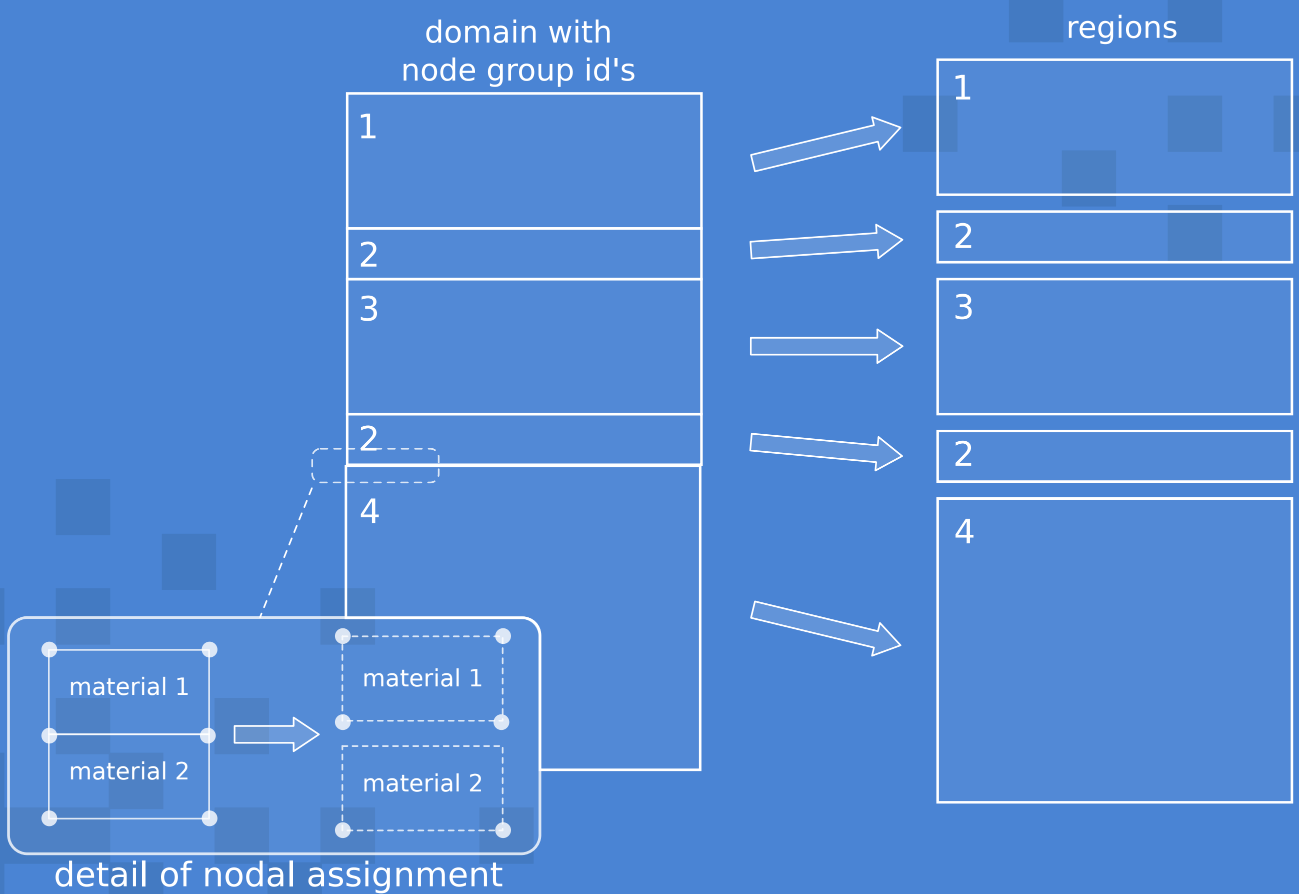
- Experiments indicated that the partitioning method is very important.
- Effective method: partition according material/element types.

  Number of iterations reduced by factor four for a test case comprising layers of soil, rock and concrete.
- Difficulty: create a specified number of sub-domains and avoid load imbalance.

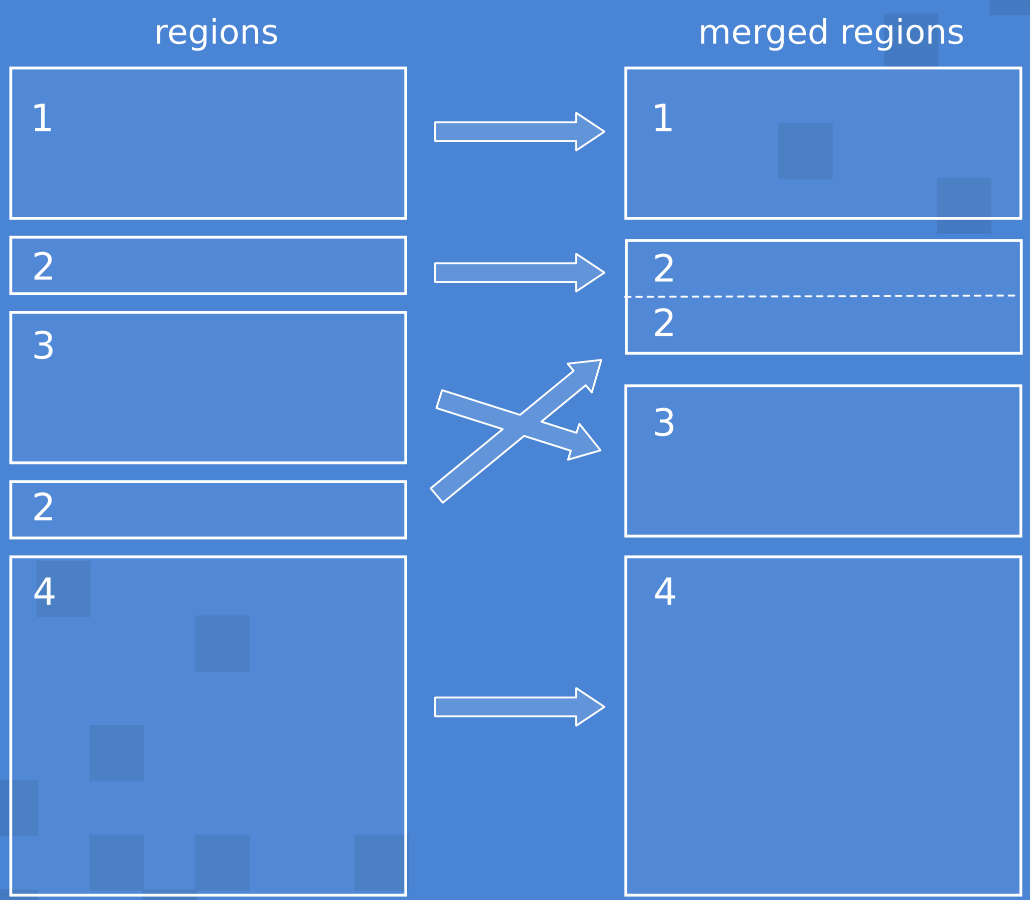
- Physics-based partitioning scheme:

  1. create ***node groups*** based on material/element types;
  2. create ***regions*** from connected nodes with the same group number;
  3. merge small regions;
  4. partition remaining regions with Metis.

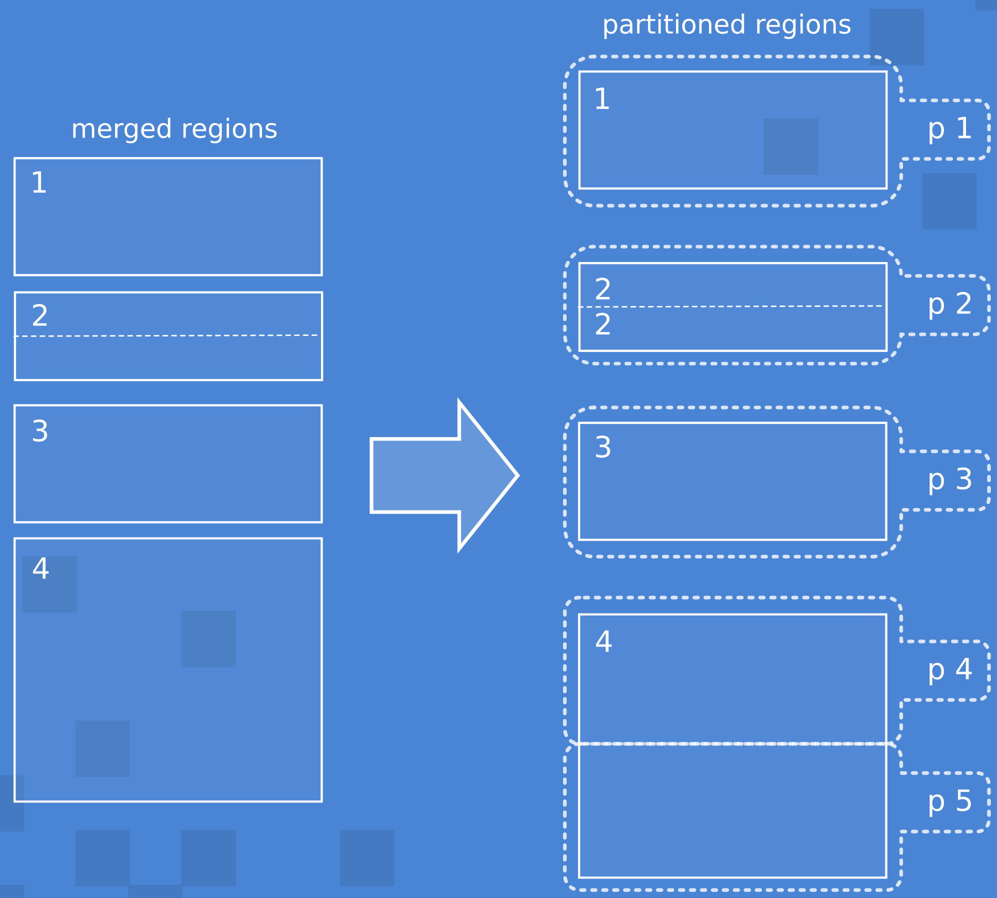
- Create ***node groups*** based on material/element types
- Create ***regions*** from connected nodes with the same group number



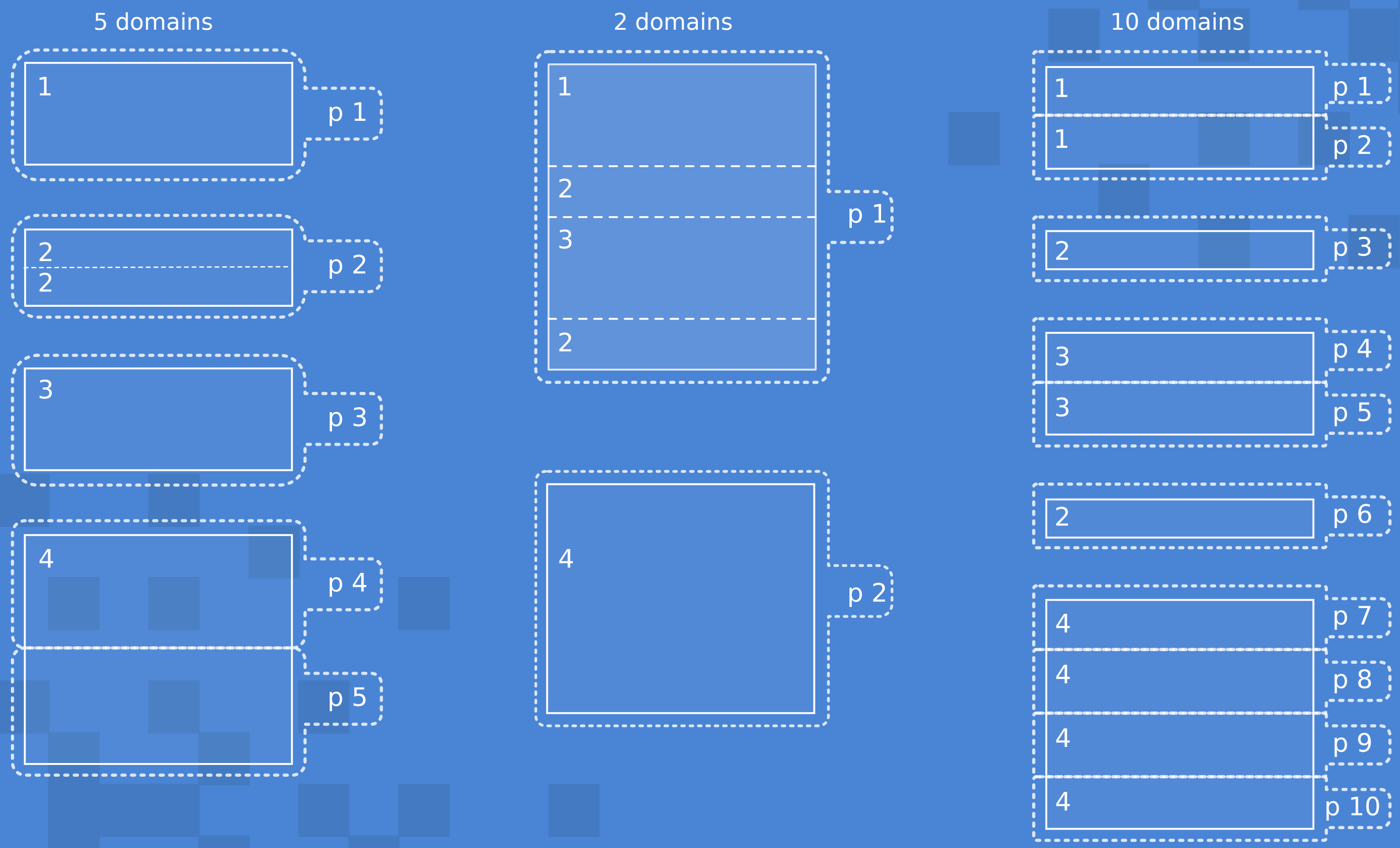
Merge small regions



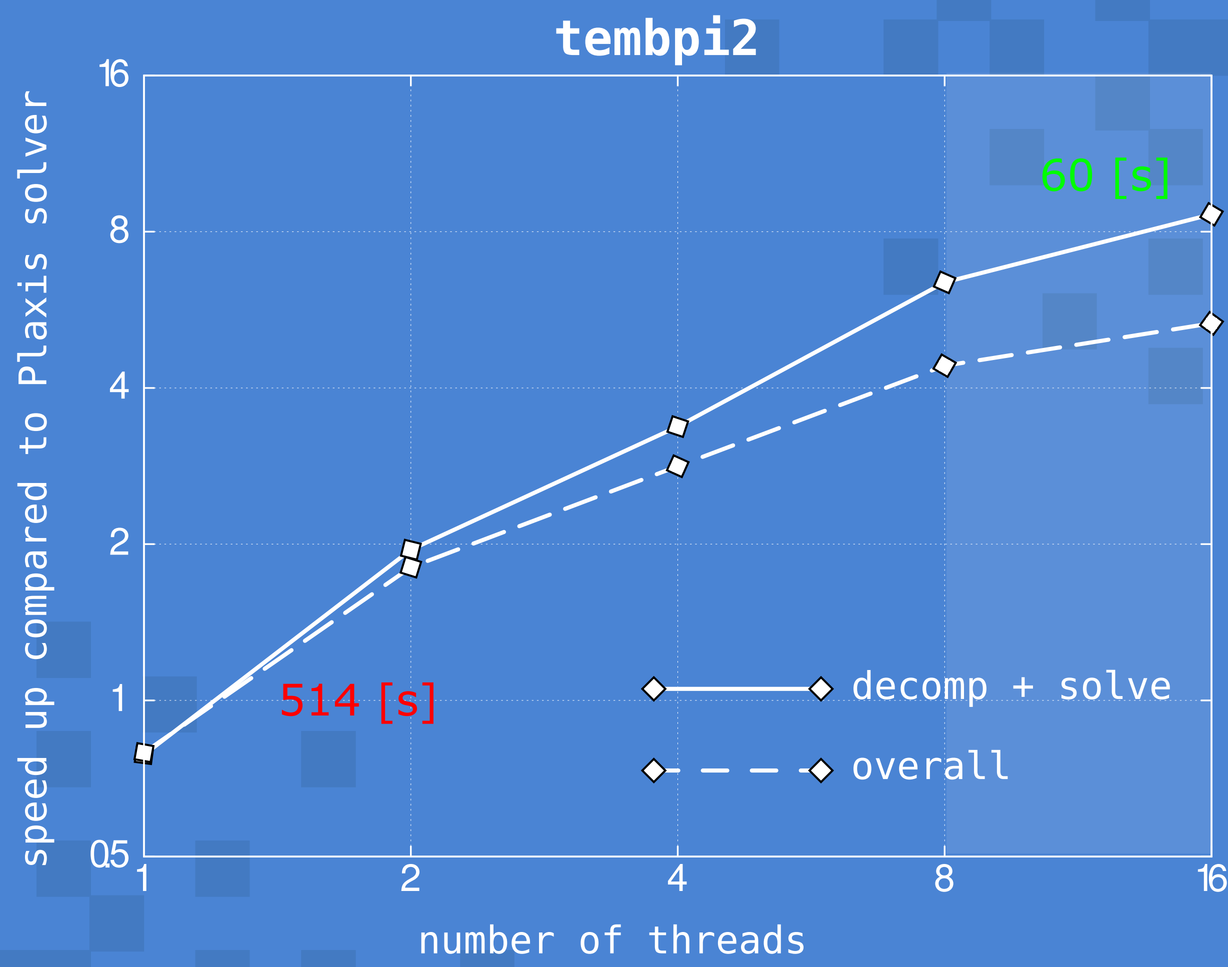
Partition remaining regions with Metis (5 sub-domains)



The more sub-domains (threads) the better the result

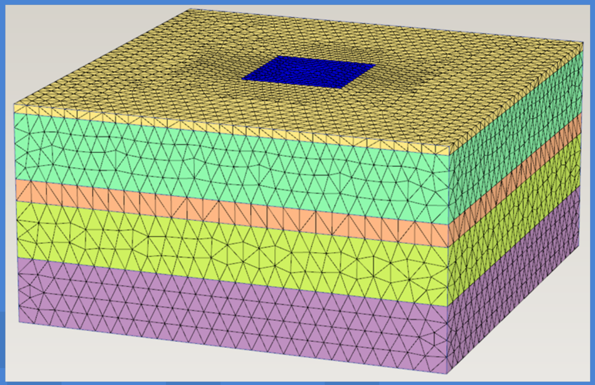


Performance results

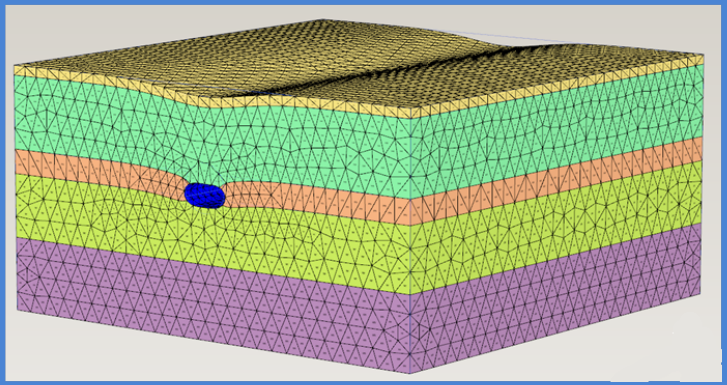

tembpi2
60 [s]
514 [s]
decomp + solve
overall
speed up compared to Plaxis solver
number of threads
16
8
4
2
1
0.5
1
2
4
8
16

Comparison with PARDISO

Model	E_{min} [MPa]	E_{max} [MPa]	# DOFs
1	1.5	$3.0 \cdot 10^4$	680,000
2	1.5	$3.0 \cdot 10^4$	414,000



Model 1



Model 2

Model 1

Solver	# threads	Precon [s]	Solve [s]	# iter
PARDISO	8	200	150	1
Original	1	320	680	140
New	1	140	550	134
	2	82	180	80
	4	43	150	111
	8	23	100	113

Model 2

Solver	# threads	Precon [s]	Solve [s]	# iter
PARDISO	8	71	72	1
Original	1	170	140	32
New	1	58	90	62
	2	22	84	74
	4	19	41	39
	8	14	29	45

Conclusions

- New solver is effective for models with large variations in material stiffness.
- Physics-based partitioning scheme is important to obtain good convergence rate.
- Rigid body modes result in an effective coarse grid preconditioner.
- Good speedup obtained on standard workstation.

- Read the full paper.
  F.J. Lingen and P.G. Bonnier and R.B.J. Brinkgreve and M.B. van Gijzen and C. Vuik
  A parallel linear solver exploiting the physical properties of the underlying mechanical problem
  Computational Geosciences, 18, pp. 913-926, 2014
  http://ta.twi.tudelft.nl/nw/users/vuik/papers/Lin14BBGV.pdf

- Contact the authors:

  - Kees Vuik (`c.vuik@tudelft.nl`)
  - Erik Jan Lingen (`erikjan.lingen@dynaflow.com`)
