Robust and Fast Solvers for Partial Differential Equations

September 17, 2015

Kees Vuik

Delft University of Technology

Deltares

Affiliation

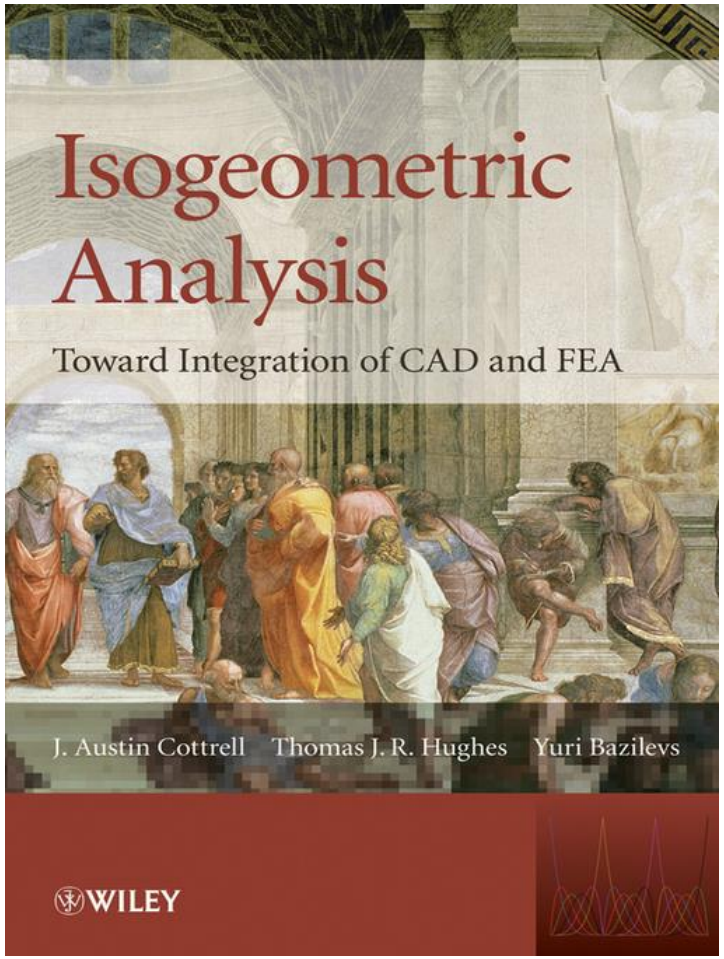
- Professor of Numerical Analysis
  http://ta.twi.tudelft.nl/users/vuik/
- Scientific Director of 3TU.AMI Applied Mathematics Institute
  http://www.3tu.nl/ami/en/
- Director of Delft Centre for Computational Science and Engineering
  http://www.cse.tudelft.nl/

Contents

- Introduction
- Two-phase flow
- Mechanical solver
- High Performance Computing

Discretization methods

- FDM
- FEM
- FVM
- DG
- isoGEO FEM



Smart algorithms

- Flexible
- Adaptive
- Robust
- Parallel
- Accuracy
- Physics-based

Future computers

- Slow increase in speed
- Double / single precision
- Parallel coarse / fine
- Memory bound
- Data movement
- Power requirements
- Heat problem

'CFD for Flow Instabilites in Multiphase Systems'

Project sponsored by:

Project description

Aim of the project:

Develop a dedicated flow solver that is able to simulate two-phase pipe flow instabilities (possibly for turbulent flows).

Project boundaries:

- Fixed (cylindrical) domain geometry and grid.
- Use FD/FV techniques for speed and efficiency.
- Use previously developed Mass Conserving Level Set method as interface model.

Calculation of the flow field

Solving the cylindrical Navier Stokes equations:

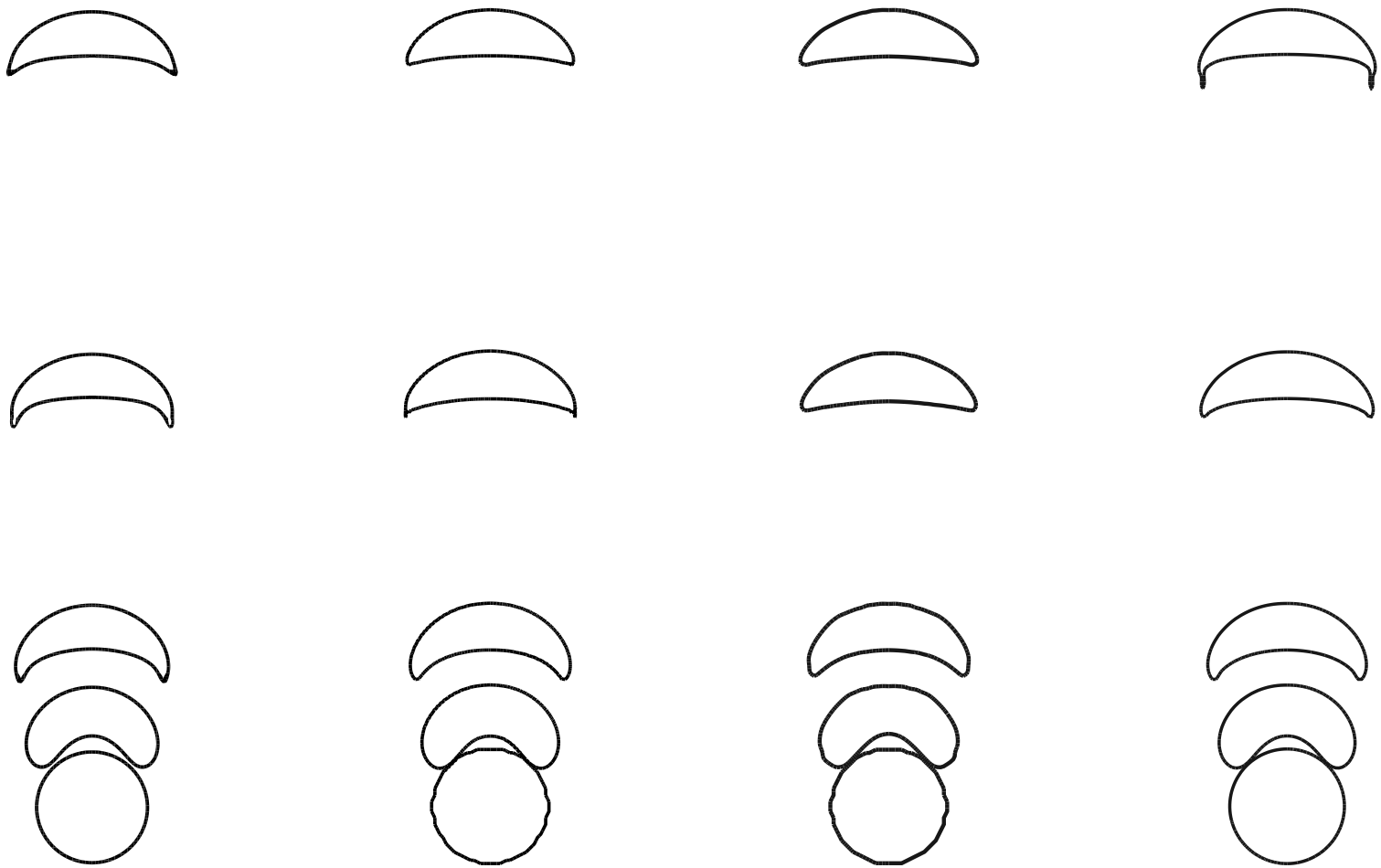
- Incompressible and isothermal on cylindrical grid.
  Structured grid for fast numerical methods and improved accuracy.
- Special attention to $r = 0$!
  Coordinate singularity.
- Conservative spatial FD and time integration scheme.
  Important for stability at high Reynolds numbers.
- Second order in space and time.
  Central discretization in space, Implicit Midpoint method in time.

Calculation of the interface

Mass Conserving Level Set (MCLS) approach:

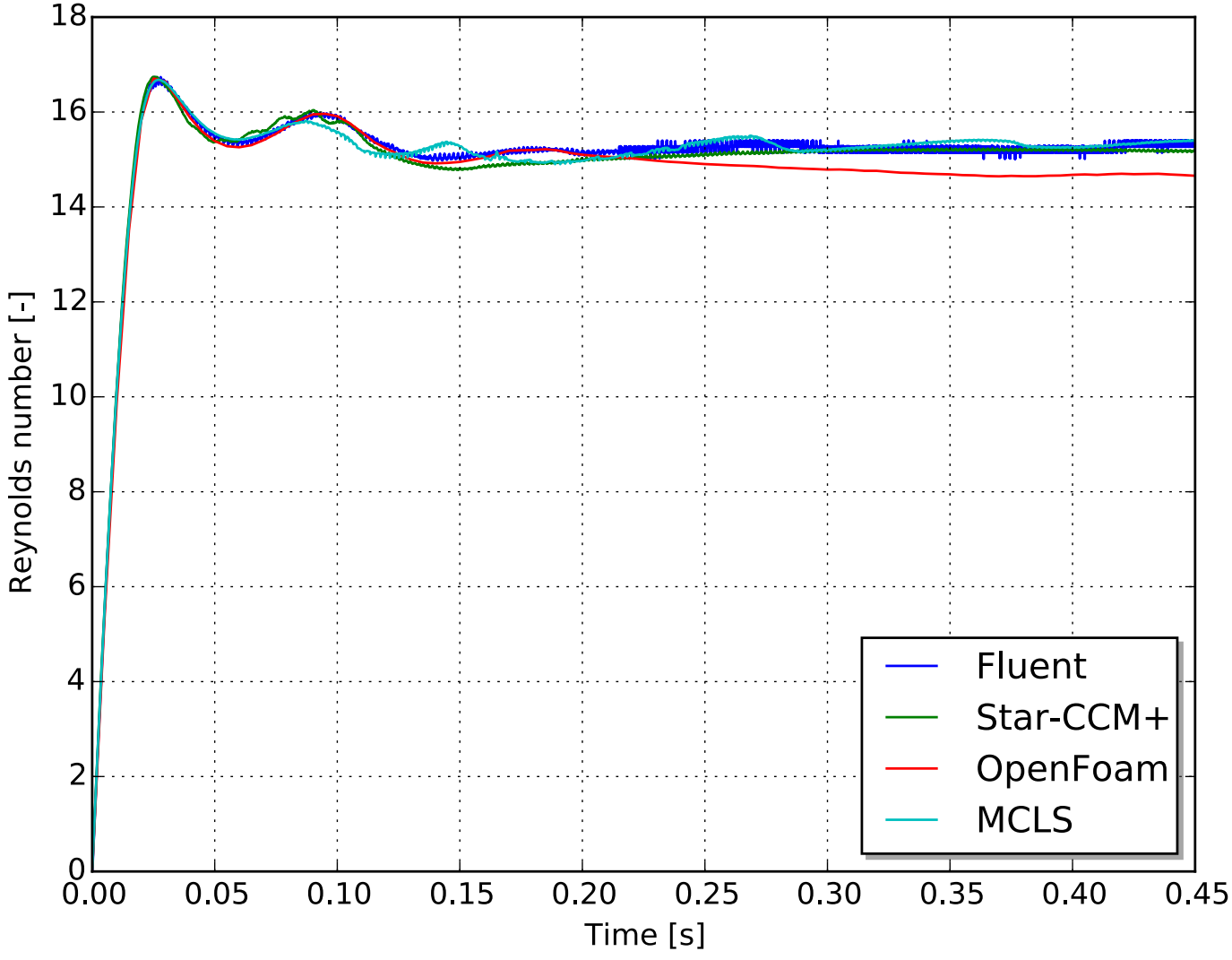
- Level Set is used for sharp interface properties.
  Main drawback: does not conserve volume over time.
- Volume of Fluid is used for conservation properties.
  Main drawback: requires complex interface reconstruction.
- Both are combined through a function $VOF = f(LS, \nabla LS)$.
- Key: LS is locally corrected using VOF to conserve mass.
- Both methods' strengths are used to form a superior hybrid method.
  Proof of concept on uniform Cartesian grids.

Some results - rising bubble



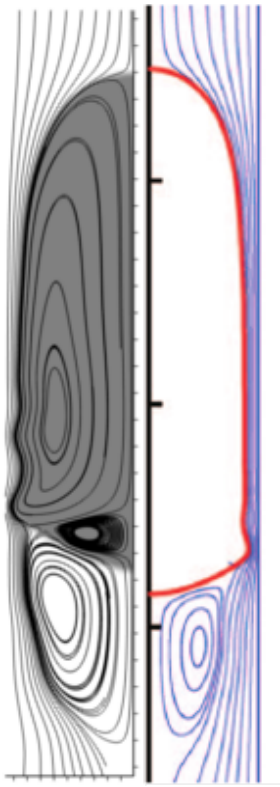
MCLS OpenFoam Star-CCM+ Fluent

Some results - rising bubble

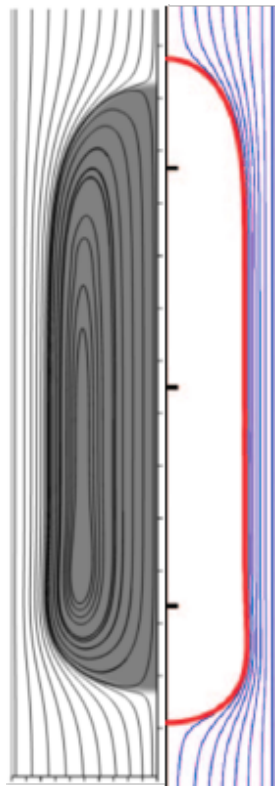


Some results - Taylor bubble

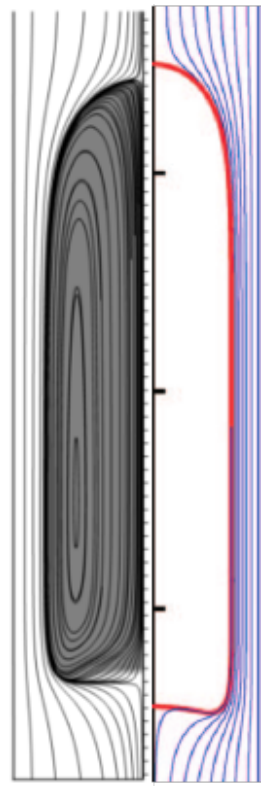
Comparison with Prosperetti and Lu, 2009, *'A Numerical Study of Taylor Bubbles'*



Eo = 15, Fr = 0.23

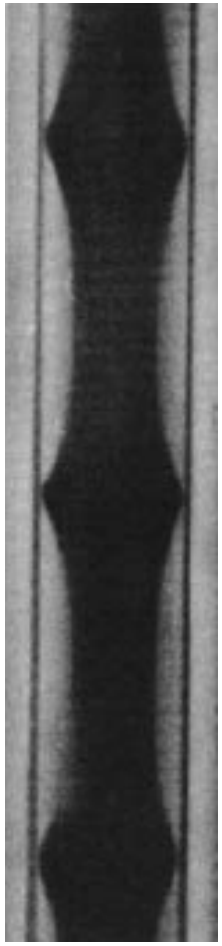


Eo = 18.7, Fr = 0.1

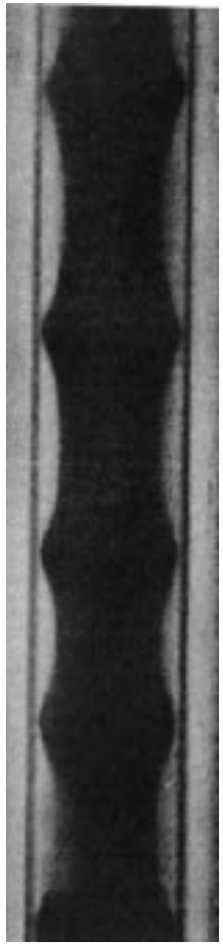


Eo = 74.6, Fr = 0.27

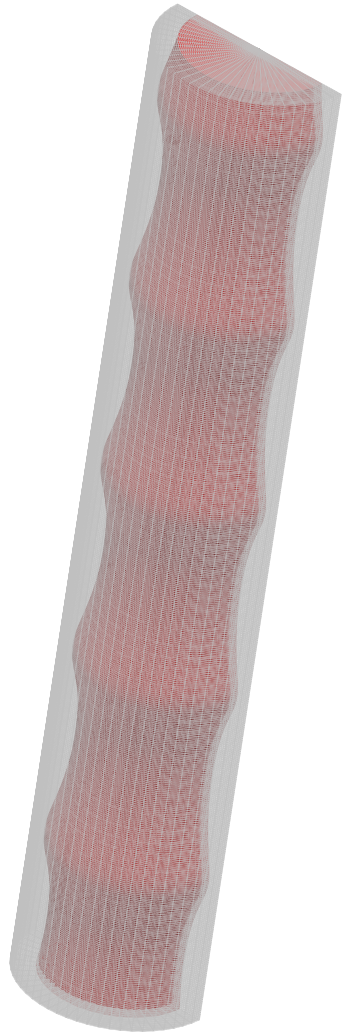
Some results - Bamboo waves



Bai, Chen, Joseph, 1991

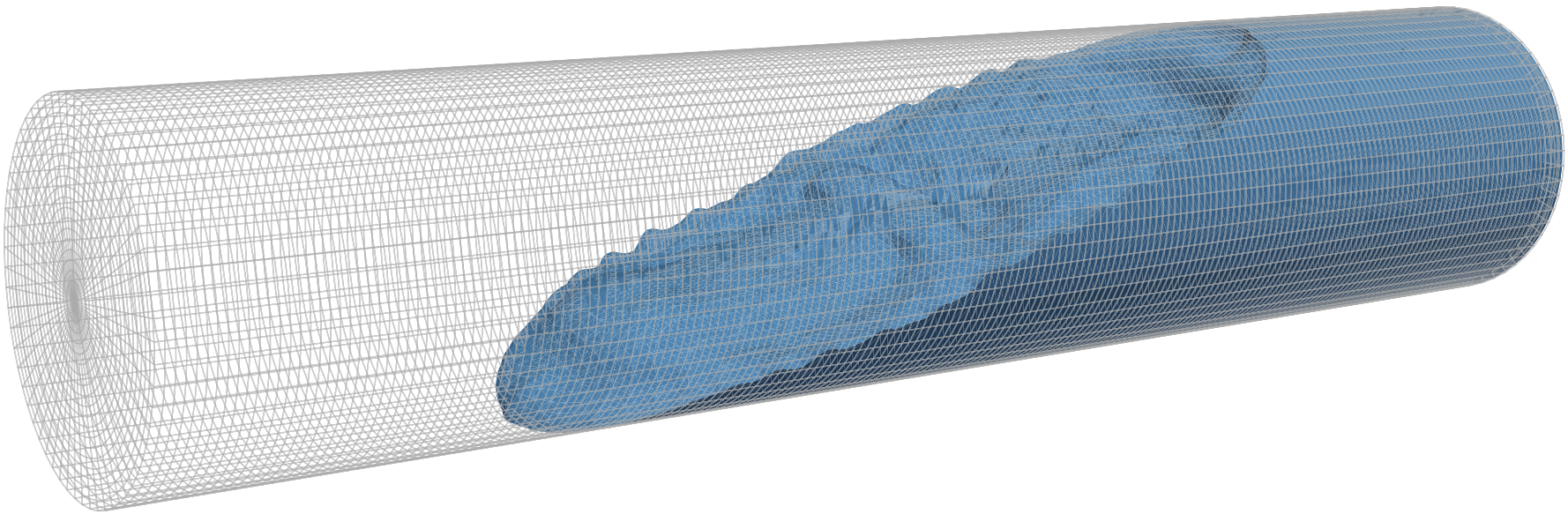


Bai, Chen, Joseph, 1991

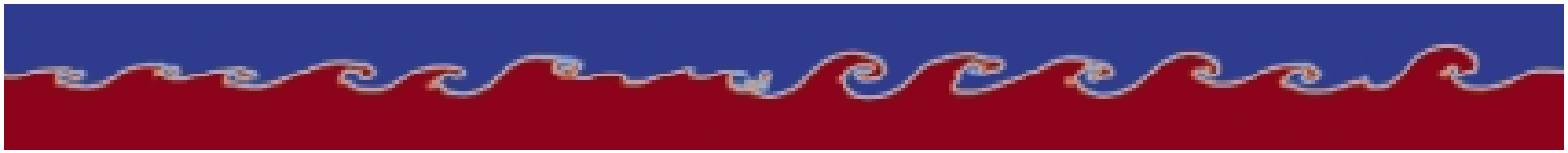


MCLS

Some results - Benjamin bubble / Breaking dam problem



Some results - Kelvin-Helmholtz instability



MCLS

OpenFOAM

Fluent (geometric reconstruction)

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.

- geomechanical problems are hard

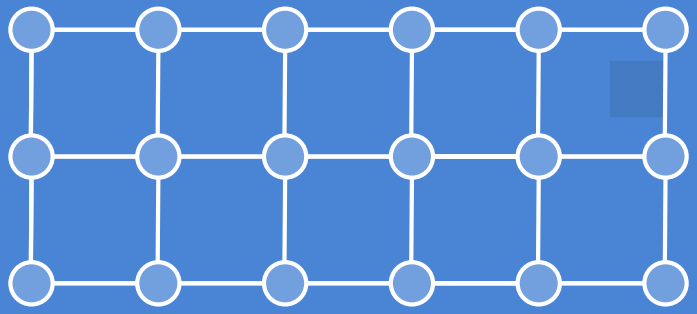
Domain decomposition

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

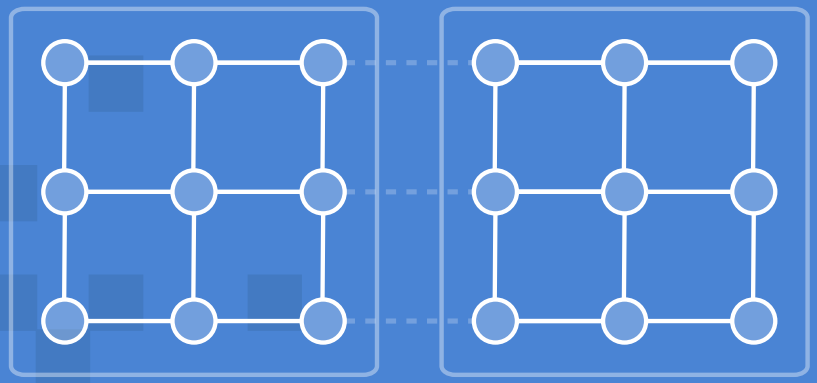
- the traditional, element-based method;
- an alternative, node-based method.

First step: partition the nodes without overlap.

Mesh

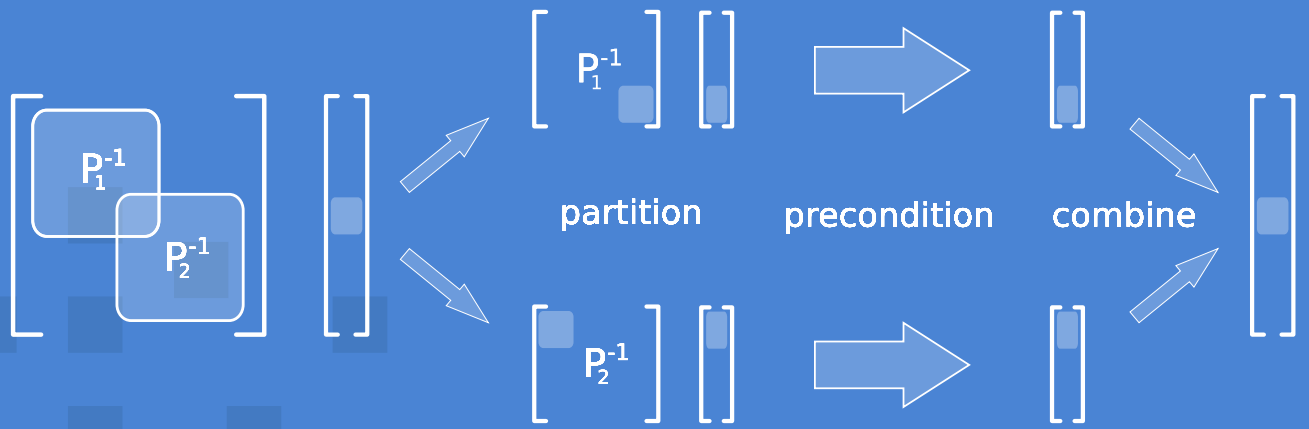


Unique node partition = native nodes

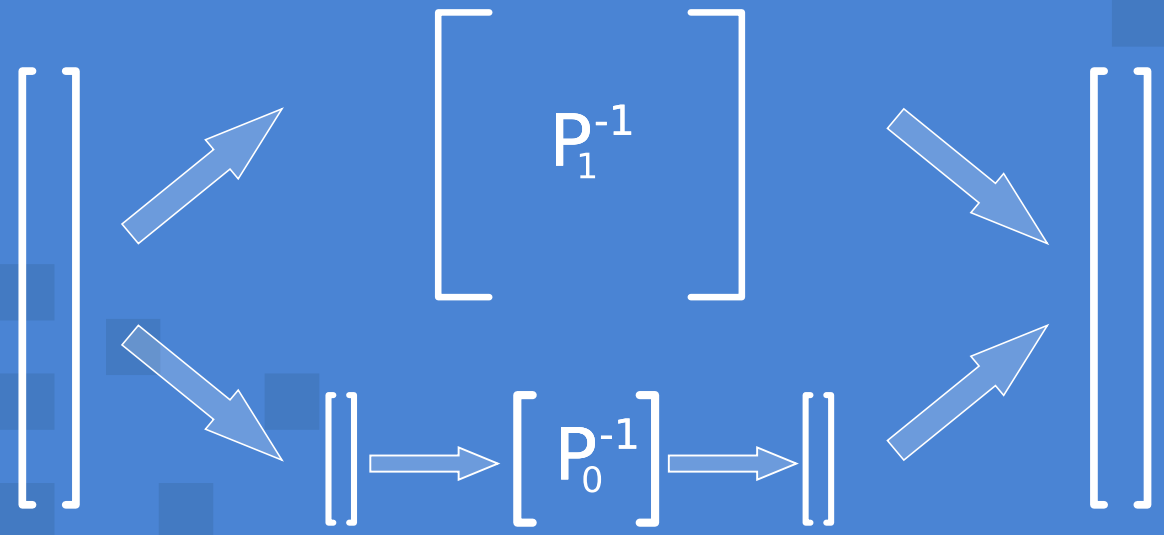


Preconditioner

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.

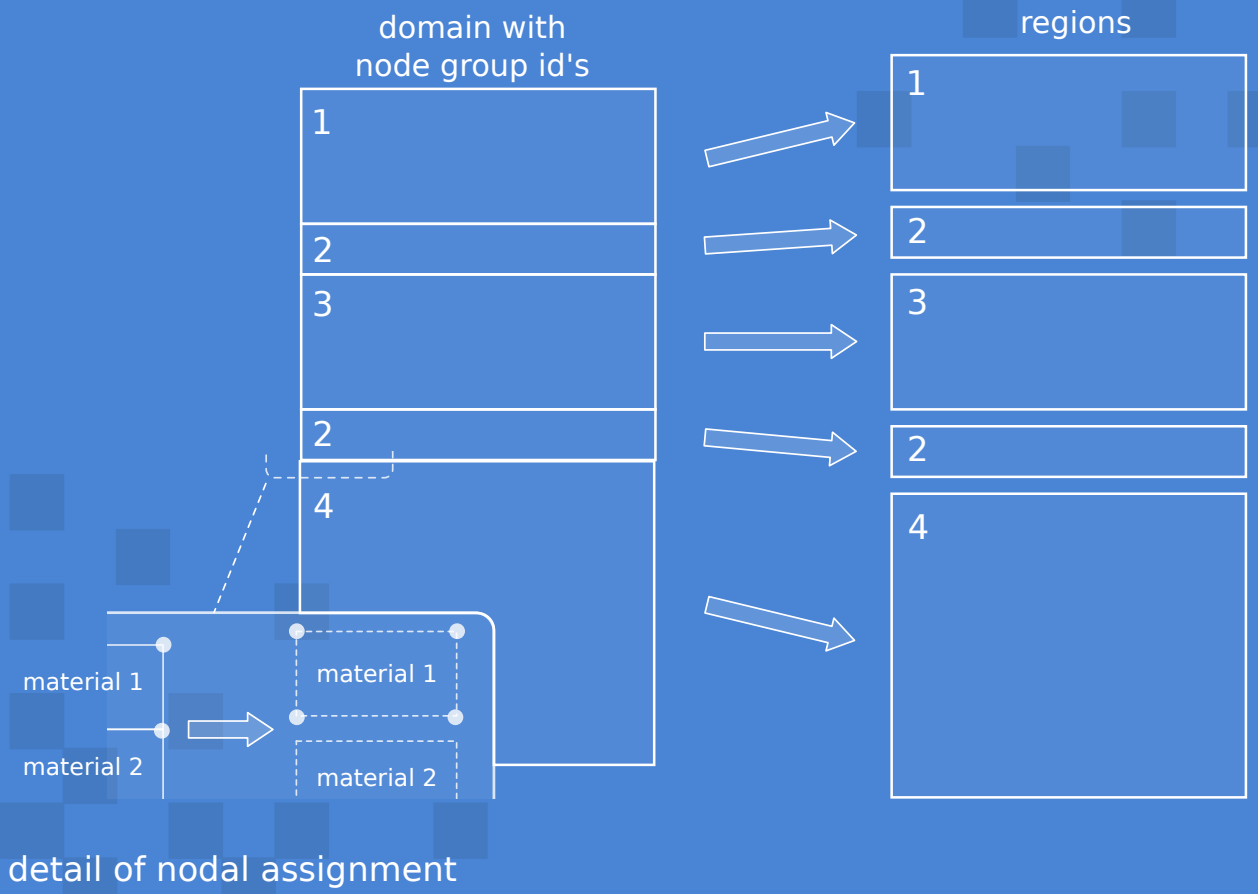
Physics-based partitioning

- 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.

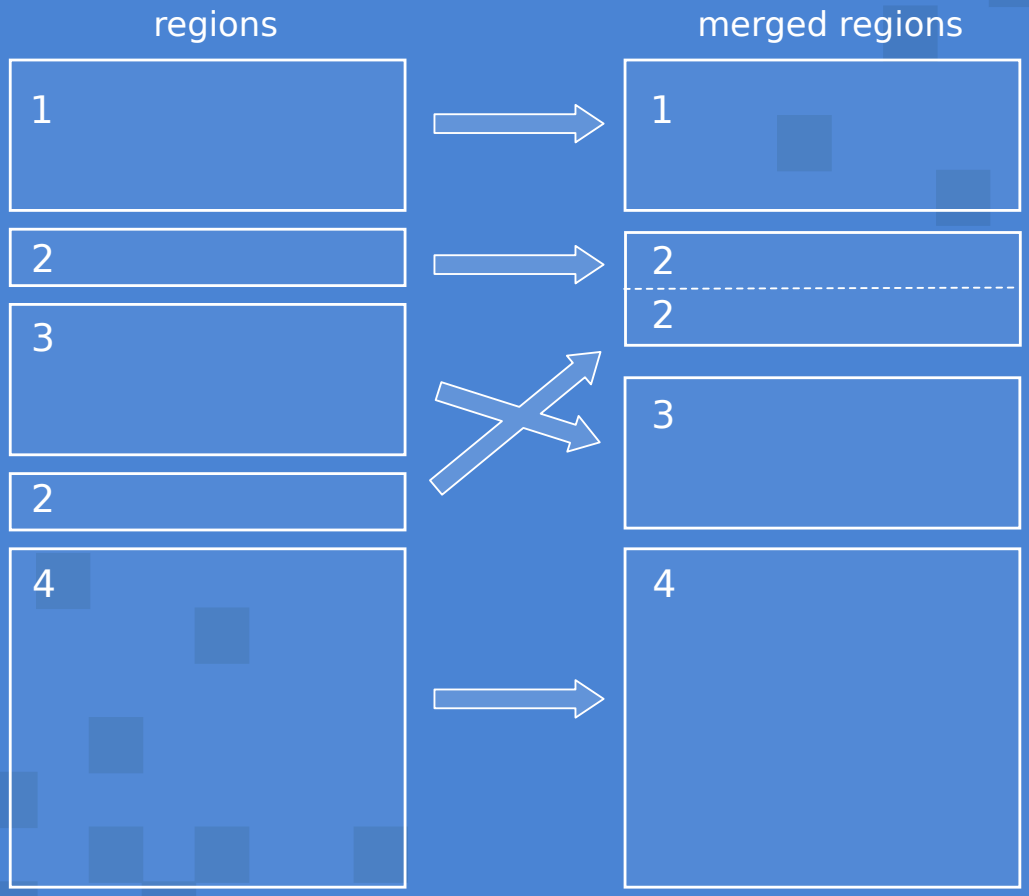
- Physics-based partitioning scheme:
  1. create ***node groups*** based on material/element types;
  2. create ***regions*** from connected nodes with the same group number;
  3. merge small regions;
  4. partition remaining regions with Metis.

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

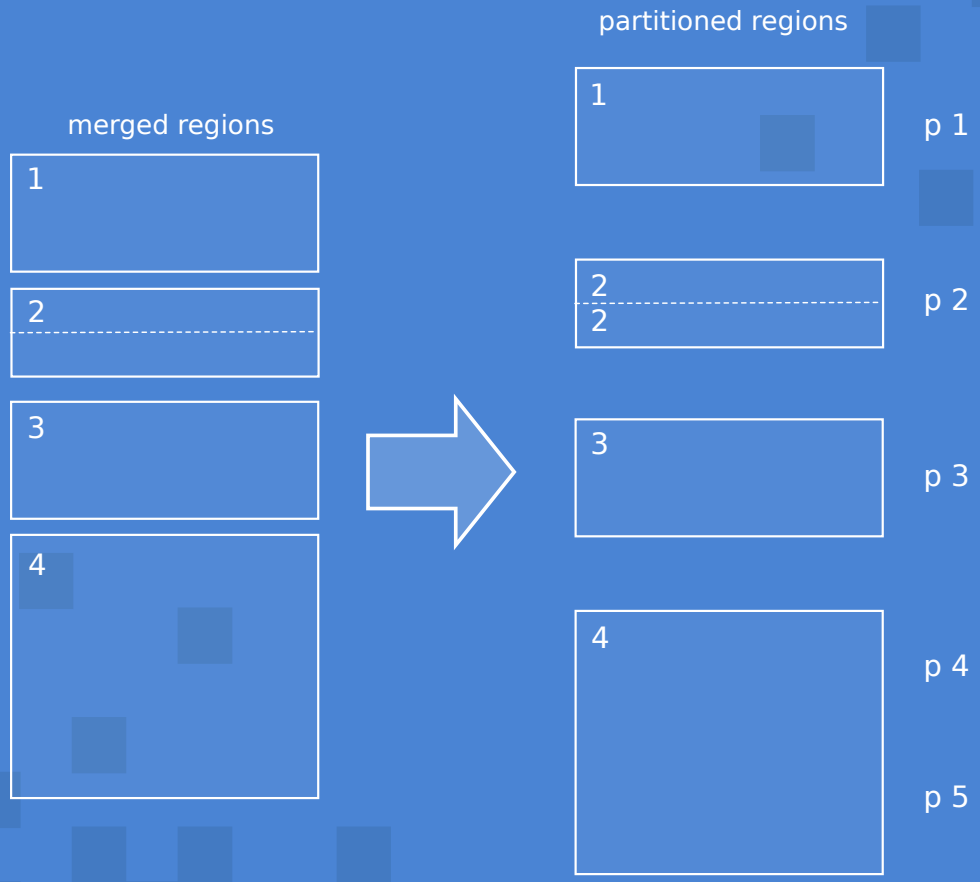


detail of nodal assignment

Merge small regions



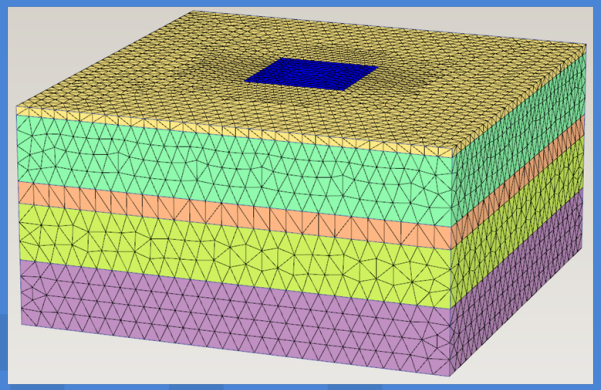
Partition remaining regions with Metis (5 sub-domains)



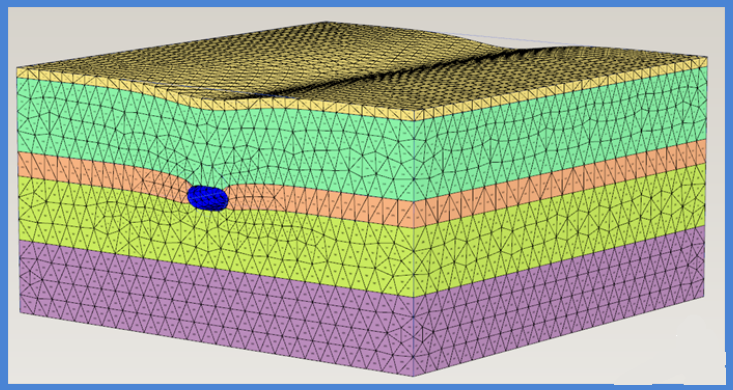
Performance results

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

- 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`)

Fast Solvers for Linear Systems on the GPU

Kees Vuik, Rohit Gupta, Martijn de Jong

Martin van Gijzen, Auke Ditzel (MARIN), Auke van der Ploeg (MARIN)

Delft Institute of Applied Mathematics

TU Delft

Delft University of Technology

Contents

1. Problem description
2. Preconditioners
   - RBB preconditioner
   - Truncated Neumann Series (TNS)
   - Deflation
3. Numerical results
4. Conclusions

1. Problem description: ship simulator

Linearized Variational Boussinesq for interactive waves:

$$\frac{\partial \zeta}{\partial t} + \nabla \cdot (\zeta \mathbf{U} + h\nabla\varphi - h\mathcal{D}\nabla\psi) = 0, \tag{1a}$$

$$\frac{\partial \varphi}{\partial t} + \mathbf{U} \cdot \nabla\varphi + g\zeta = -P_s, \tag{1b}$$

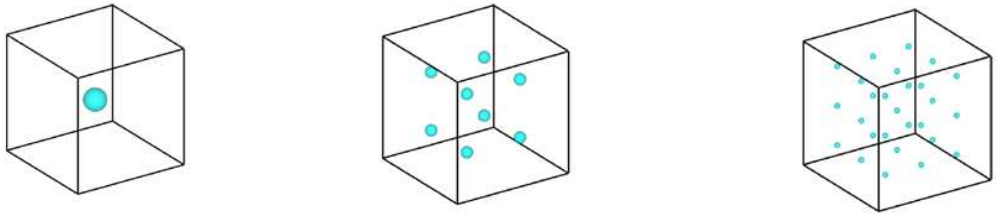
$$\mathcal{M}\psi + \nabla \cdot (h\mathcal{D}\nabla\varphi - \mathcal{N}\nabla\psi) = 0. \tag{1c}$$

After discretization (FVM for space, Leapfrog for time):

$$A\vec{\psi} = \mathbf{b}, \tag{2}$$

$$\frac{\mathrm{d}\mathbf{q}}{\mathrm{d}t} = L\mathbf{q} + \mathbf{f}. \tag{3}$$

Problem Description: Bubbly Flow



Mass-Conserving Level-Set method for Navier Stokes

$$-\nabla.(\frac{1}{\rho(x)}\nabla p(x)) = f(x),\ x \in \Omega \tag{4}$$

$$\frac{\partial}{\partial n}p(x) = 0,\ x \in \partial\Omega \tag{5}$$

- Pressure-Correction equation is discretized to $Ax = b$.
- Most time consuming part is the solution of this SPSD system

2. Preconditioners: RRB

The RRB-solver:

- is a PCG-type solver (Preconditioned Conjugate Gradient)
- uses as preconditioner: the RRB preconditioner

RRB stands for “Repeated Red-Black”.

The RRB preconditioner determines an incomplete factorization:

$$A = LDL^T + R \quad \Longrightarrow \quad M = LDL^T \approx A$$

Preconditioners: RRB

As the name RRB reveals: multiple levels

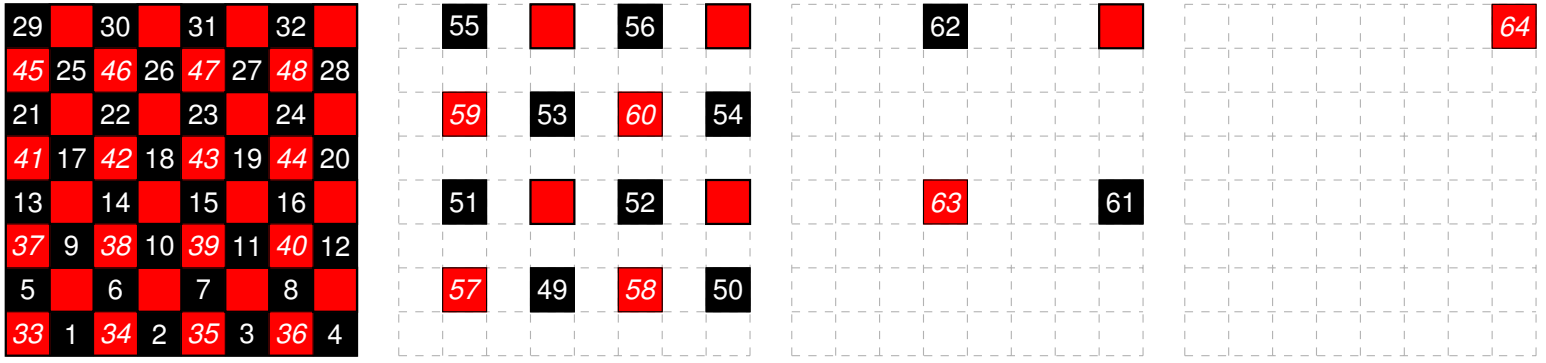
Therefore the RRB-solver has good scaling behaviour (Multigrid)

Method of choice because:

- shown to be robust for all of MARIN's test problems
- solved all test problems up to 1.5 million nodes within 7 iterations(!)

Special ordering

An 8×8 example of the RRB-numbering process



(1) (2) (3) (4)

All levels combined:

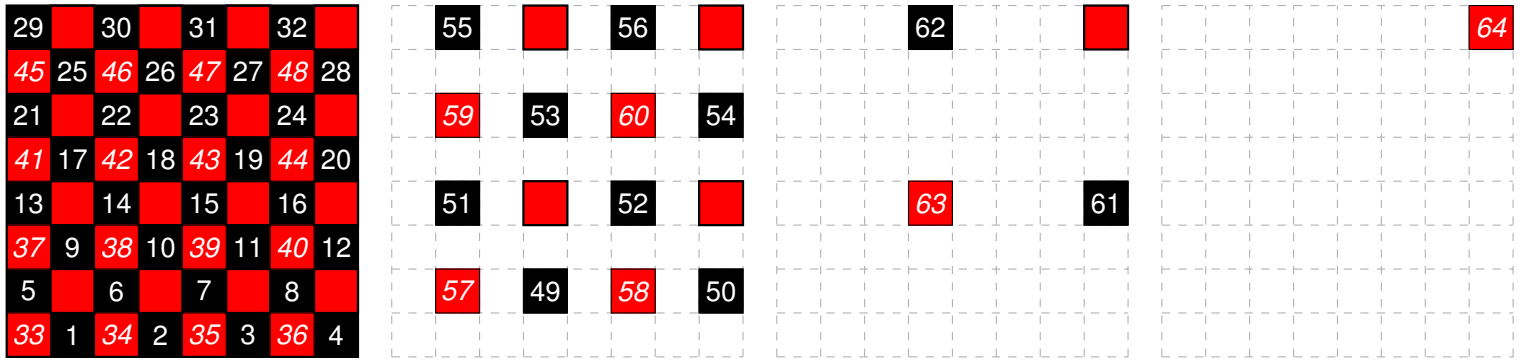
29	55	30	62	31	56	32	64
45	25	46	26	47	27	48	28
21	59	22	53	23	60	24	54
41	17	42	18	43	19	44	20
13	51	14	63	15	52	16	61
37	9	38	10	39	11	40	12
5	57	6	49	7	58	8	50
33	1	34	2	35	3	36	4

CUDA implementation (1)

Besides the typical Multigrid issues such as idle cores on the coarsest levels, in CUDA the main problem was getting “coalesced memory transfers”.

Why is that?

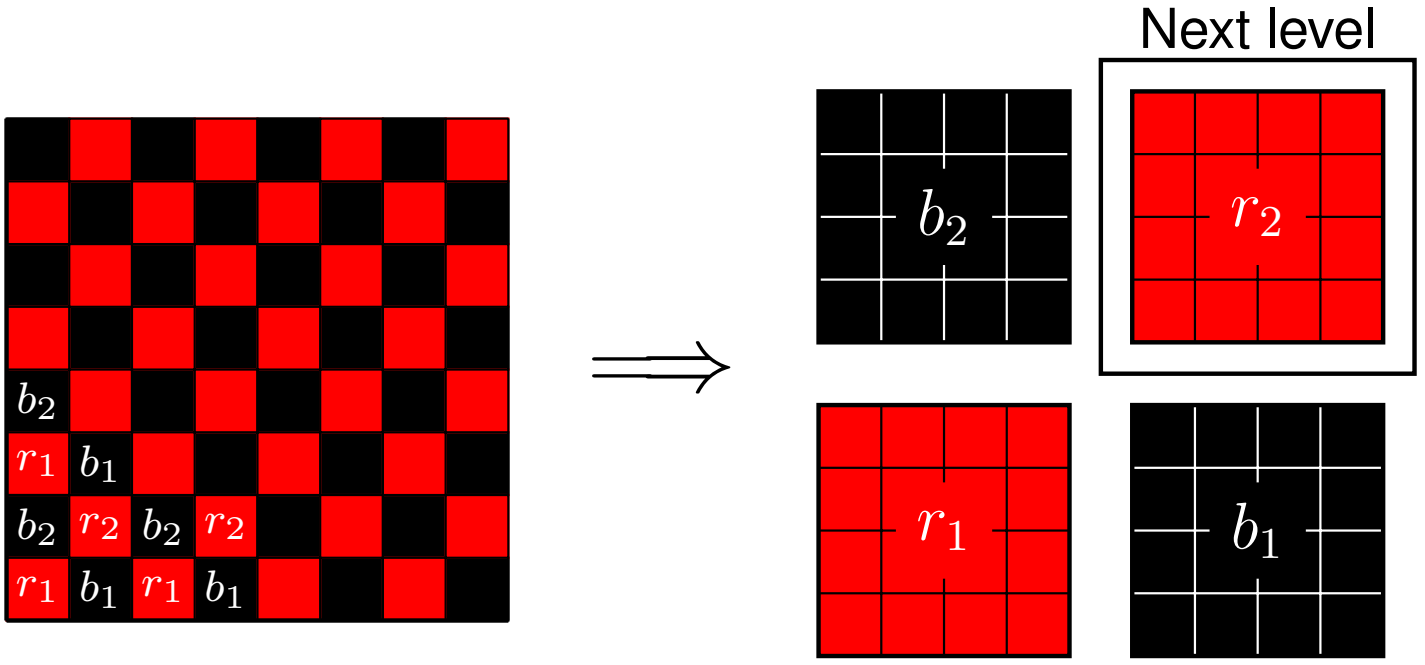
Recall the RRB-numbering: the number of nodes becomes 4× smaller on every next level:



CUDA implementation (2)

New storage scheme: $r_1/r_2/b_1/b_2$

Nodes are divided into four groups:



Preconditioners: TNS

Truncated Neumann Series Preconditioning[a],[b]

$$M^{-1} = K^T D^{-1} K, \text{ where } K = (I - LD^{-1} + (LD^{-1})^2 + \cdots)$$

L is the strictly lower triangular of A, and D=diag(A).

1. More terms give better approximation.
2. In general the series converges if $\| LD^{-1} \|_\infty < 1$.
3. As much parallelism as Sparse Matrix Vector Product.

---

[a] A vectorizable variant of some ICCG methods. Henk A. van der Vorst. SIAM Journal of Scientific Computing. Vol. 3 No. 3 September 1982.

[b] Approximating the Inverse of a Matrix for use in Iterative Algorithms on Vector Processors. P.F. Dubois. Computing (22) 1979.

Preconditioners: Deflation

Removes small eigenvalues from the spectrum of $M^{-1}A$.

The linear system $Ax = b$ can be solved by the splitting,

$$x = (I - P^T)x + P^T x \text{ where } P = I - AQ. \tag{6}$$

$$\Leftrightarrow Pb = PA\hat{x}. \tag{7}$$

$Q = ZE^{-1}Z^T$, $E = Z^T AZ$.

$Em = a1$ is the coarse system

Z is an approximation of the 'bad' eigenvectors of $M^{-1}A$.

For our experiments Z consists of piecewise constant vectors.

Preconditioners: Deflation

Operations involved in deflation[a] [b].

- $a1 = Z^T p$.
- $m = E^{-1} a1$.
- $a2 = AZm$.
- $\hat{w} = p - a2$.

where, $E = Z^T AZ$ is the Galerkin Matrix and Z is the matrix of deflation vectors.

[a] Efficient deflation methods applied to 3-D bubbly flow problems. J.M. Tang, C. Vuik Elec. Trans. Numer. Anal. 2007.

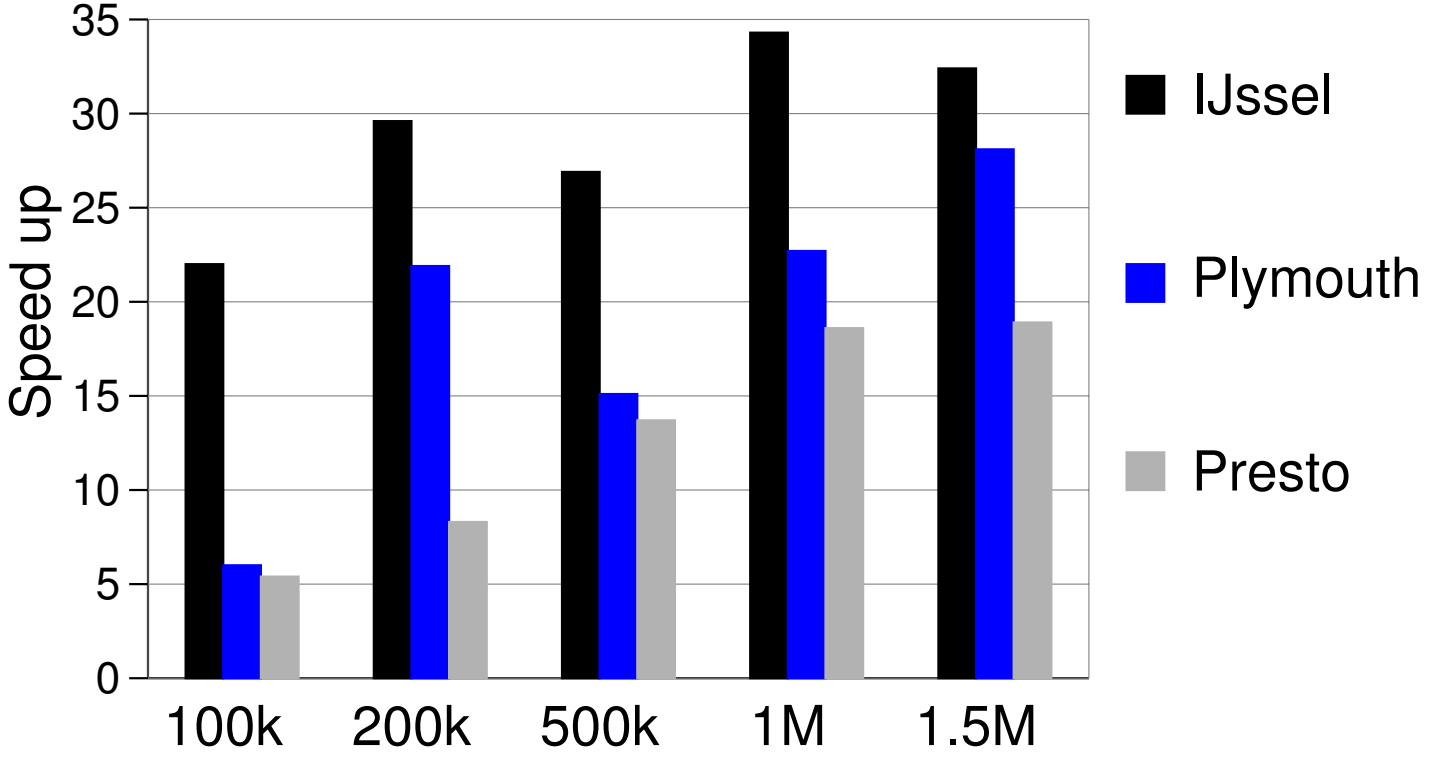
[b] An efficient preconditioned CG method for the solution of a class of layered problems with extreme contrasts in the coefficients. C. Vuik, A. Segal, J.A. Meijerink J. Comput. Phys. 1999.

3. Numerical results: ship simulator



- Including: 2D Poisson, Gelderse IJssel (NL), Plymouth Sound (UK)
- Realistic domains up to 1.5 million nodes

Numerical results: ship simulator



Speed up numbers for the realistic test problems.

Numerical results: Bubbly flow

$$Speedup = \frac{T_{CPU}}{T_{GPU}} \tag{8}$$

- Number of Unknowns = 128^3.
- Tolerance set to 10^{-6}.
- Density Contrast is 10^{-3}

Naming deflation vectors

- SD-i -> Sub-domain deflation with i vectors.
- LS-i -> Level-Set deflation with i vectors.
- LSSD-i -> Level-Set Sub-domain deflation with i vectors.

Numerical results: Bubbly flow

9 bubbles - 64 Sub-domains

	CPU	GPU-CUSP	
	DICCG(0)	DPCG(TNS)	
	SD-64	SD-63	LSSD-135
Number of Iterations	472	603	136
Total Time	81.39	13.61	5.58
Iteration Time	81.1	10.61	2.48
Speedup	-	7.64	32.7

4. Conclusions

- ILU type preconditioners can be used on GPU's by a Neumann series approach or a carefull reordering
- Deflation type preconditioners are very suitable for GPU's
- The combination of Neumann series and Deflation preconditioners leads to robust and fast solvers on the GPU
- A special ordering of a red black reordering can lead to speedup of a factor 30-40 on the GPU.

References

- H. Knibbe and C.W. Oosterlee and C. Vuik GPU implementation of a Helmholtz Krylov solver preconditioned by a shifted Laplace multigrid method Journal of Computational and Applied Mathematics, 236, pp. 281-293, 2011
- R. Gupta, M.B. van Gijzen and C. Vuik 3D Bubbly Flow Simulation on the GPU - Iterative Solution of a Linear System Using Sub-domain and Level-Set Deflation, 21st Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), 2013, ISBN 978-1-4673-5321-2, pp. 359-366, 2013
  http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6498576
- M. de Jong Developing a CUDA solver for large sparse matrices for MARIN, MSc Thesis, Delft University of Technology, 2012
  http://ta.twi.tudelft.nl/nw/users/vuik/numanal/jong_afst.pdf

Questions and Remarks