Solution of Large-Scale Black Oil Recovery Problem in Parallel Using INMOST Platform

Igor Konshin[1,2,3,4], Kirill Terekhov[1]

[1]Marchuk Institute of Numerical Mathematics of the Russian Academy of Sciences
[2]Moscow Institute of Physics and Technology
[3]Dorodnicyn Computing Centre of the Russian Academy of Sciences
[4]Sechenov University

Russian Supercomputing Days

Septermber 28, 2021

INMOST

INMOST (www.inmost.org, www.inmost.ru) is a short for:

Integrated

Numerical

Modeling and

Object-oriented

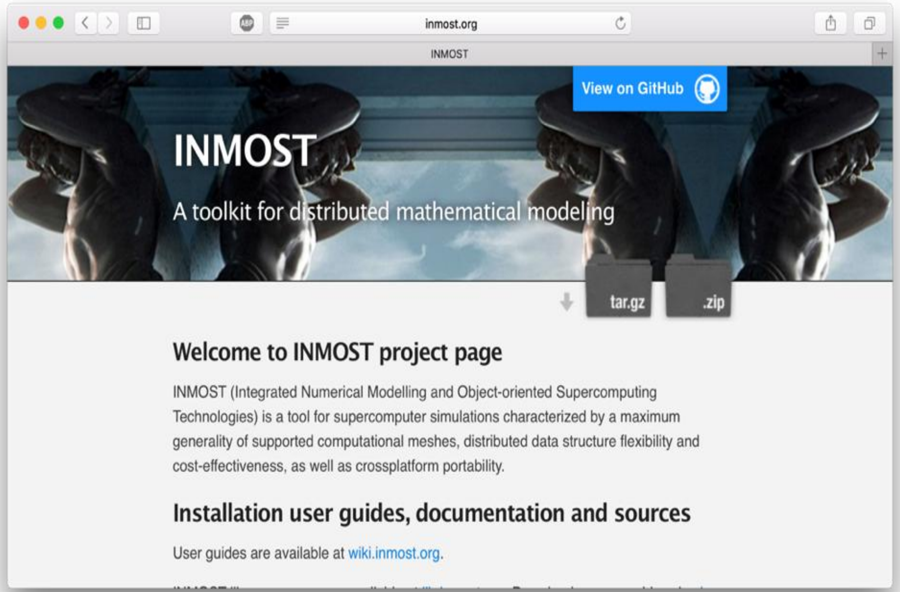
Supercomputing

Technologies

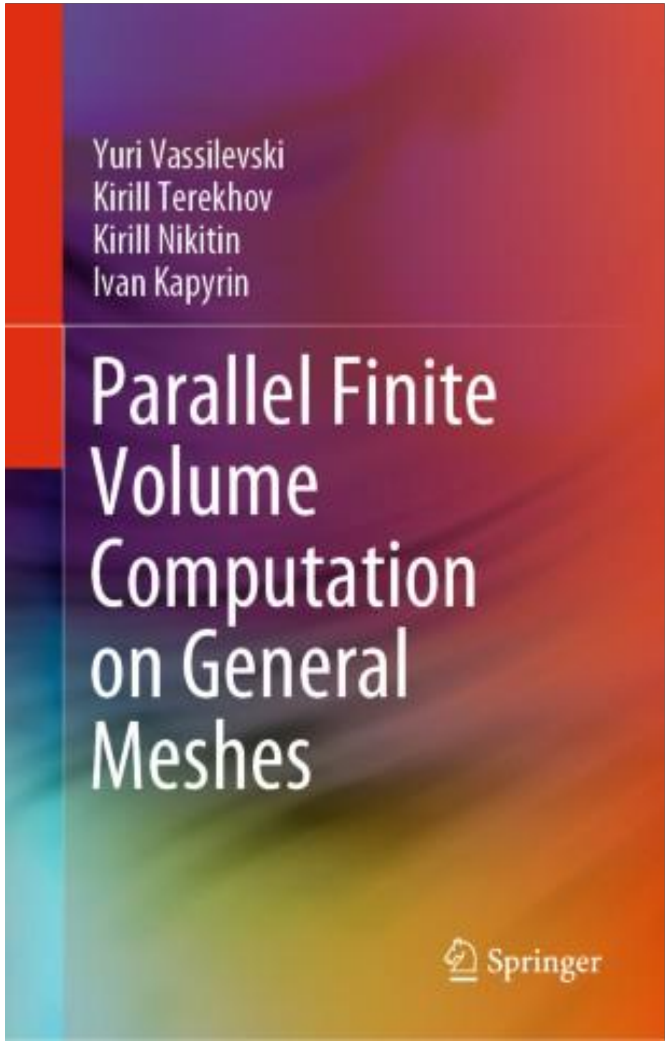
- Distributed meshes
- Distributed linear system assembly
- Parallel linear solver
- Automatic differentiation
- Nonlinear system assembly
- Coupling of unknowns and models

First version during 2012 internship at ExxonMobil

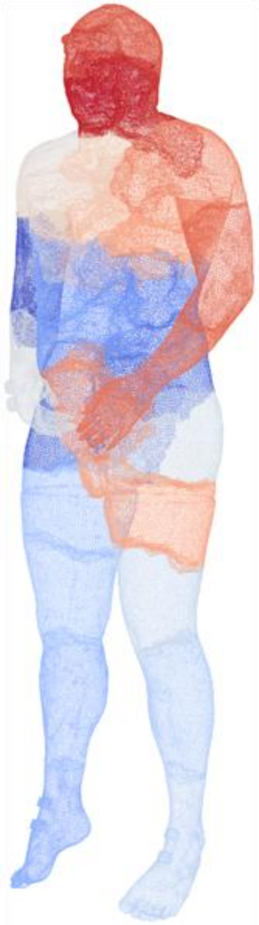
Contributors: Igor Konshin, Kirill Nikitin, Alexander Danilov, Ivan Kapyrin, Yuri Vassilevski, Alexei Chernyshenko (INM RAS, IBRAE RAS), Igor Kaporin (CMC RAS) Dmitri Bagaev, Andrei Burachkovski (MSU), Ruslan Yanbarisov, Alexei Logkiy, Sergei Petrov, Ivan Butakov (MIPT), German Kopytov (BFU), Timur Garipov, Pavel Tomin, Christine Mayer (Stanford), Ahmad Abushaikha, Longlong Li (HBKU), et al


inmost.org
INMOST
View on GitHub
INMOST
A toolkit for distributed mathematical modeling
tar.gz
.zip
Welcome to INMOST project page
INMOST (Integrated Numerical Modelling and Object-oriented Supercomputing Technologies) is a tool for supercomputer simulations characterized by a maximum generality of supported computational meshes, distributed data structure flexibility and cost-effectiveness, as well as crossplatform portability.
Installation user guides, documentation and sources
User guides are available at wiki.inmost.org.

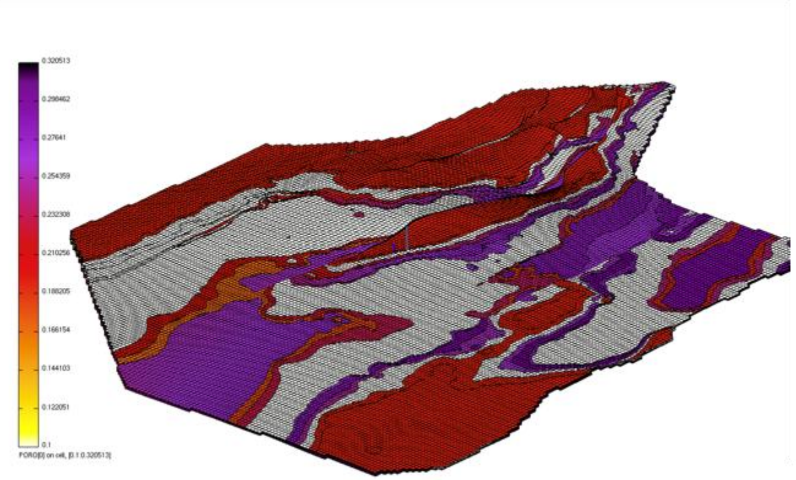

Серия
Суперкомпьютерное
Образование
Ю. В. Василевский, И. Н. Коньшин,
Г. В. Копытов, К. М. Терехов
INMOST программная платформа и графическая среда
для разработки параллельных численных моделей
на сетках общего вида
INMOST

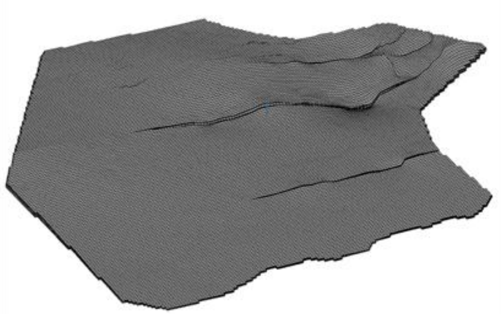

Yuri Vassilevski
Kirill Terekhov
Kirill Nikitin
Ivan Kapyrin
Parallel Finite Volume Computation on General Meshes
Springer

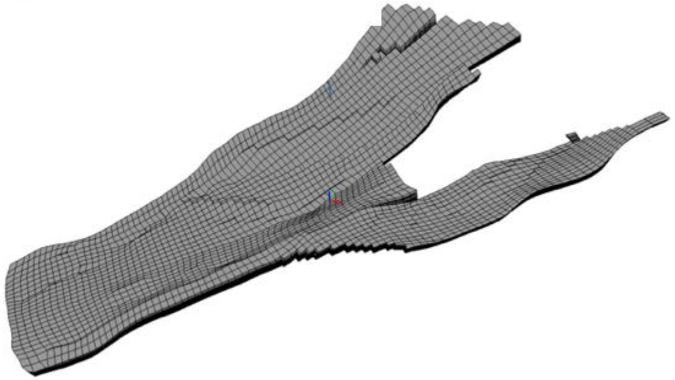
Grids



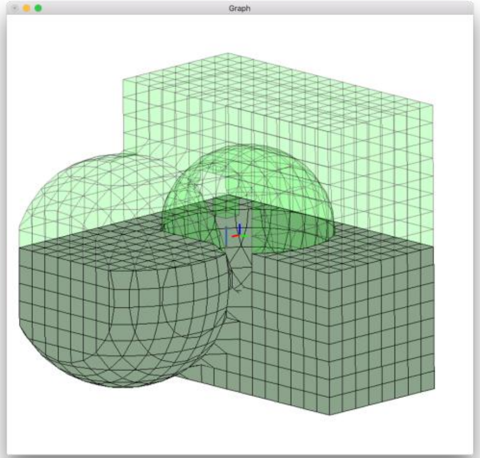
Human body



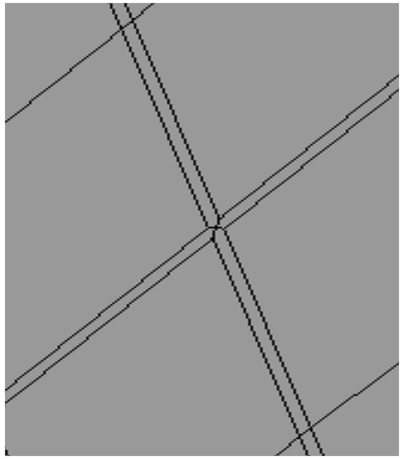




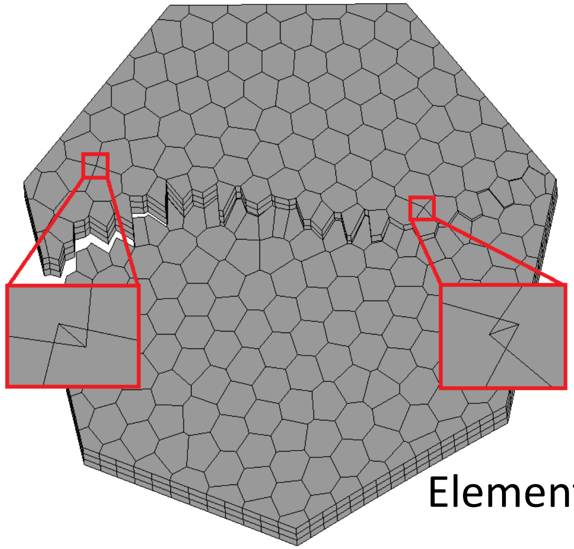
Geological grids with faults and pinch-outs, support for commercial formats of oil & gas simulators



Complex modifications

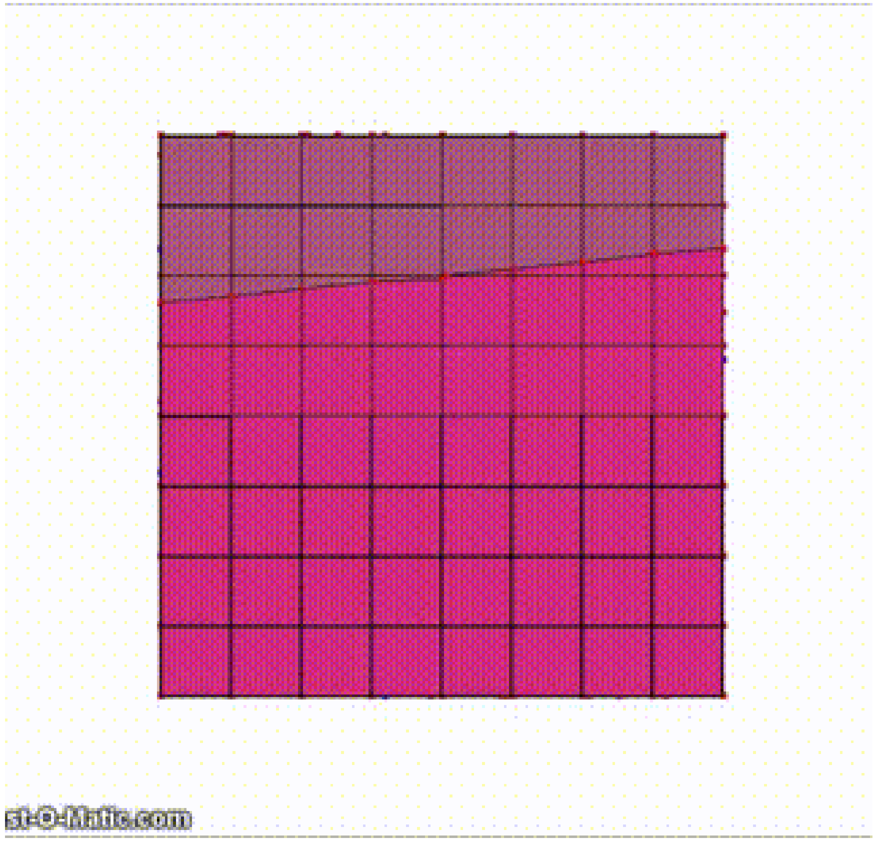


Fracture opening

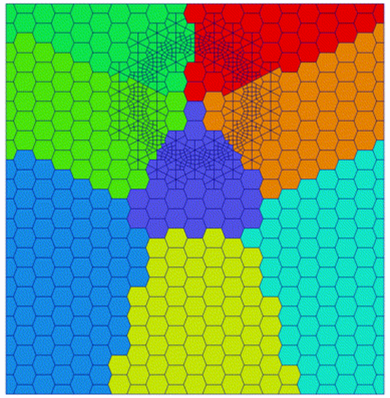


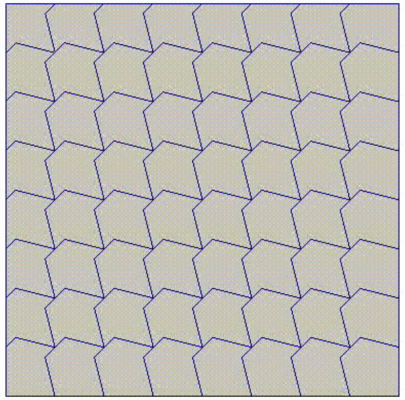
Element collapse

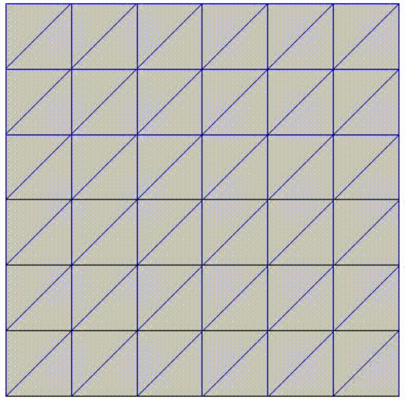
Dynamic grids



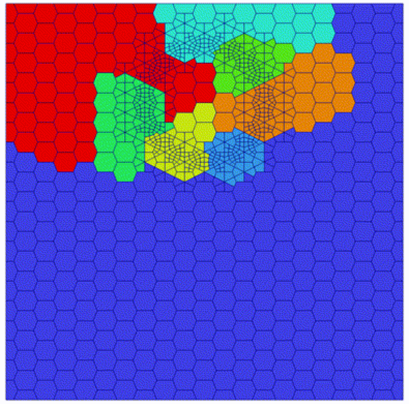
OctreeCutcell example in INMOST-Graphics repository

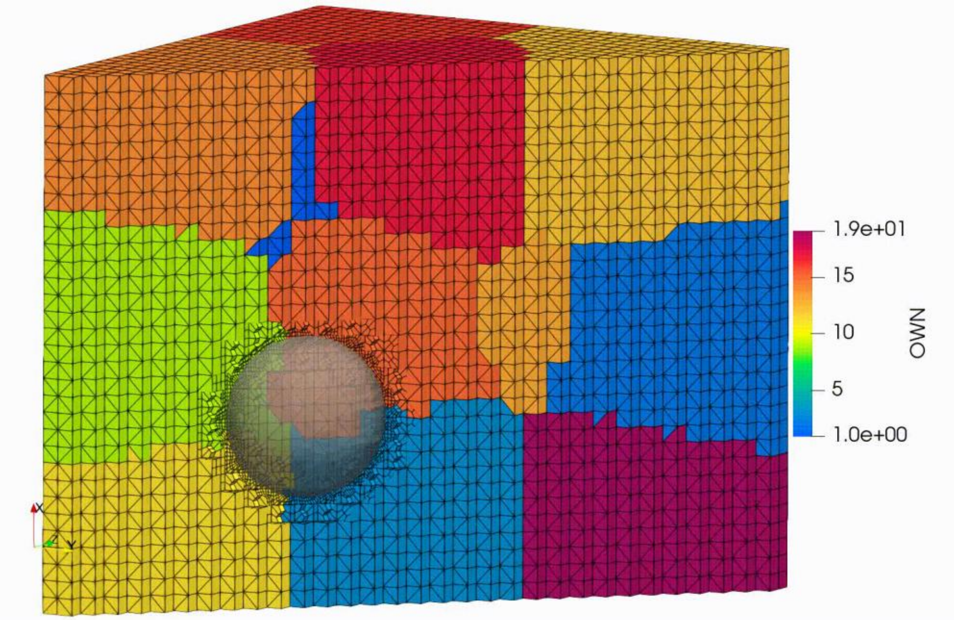






AdaptiveMesh example for general grid adaptation



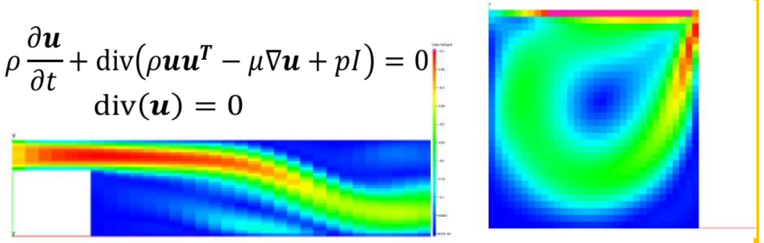


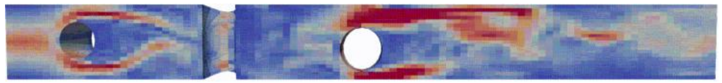
Parmetis_AdaptiveRepart

Models

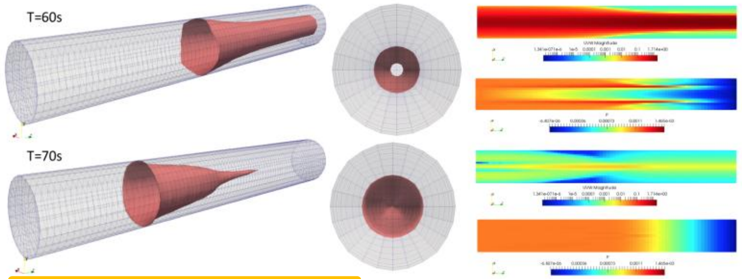
Navier-Stokes

$$\rho \frac{\partial \boldsymbol{u}}{\partial t} + \operatorname{div}\left(\rho \boldsymbol{u}\boldsymbol{u}^T - \mu \nabla \boldsymbol{u} + pI\right) = 0$$
$$\operatorname{div}(\boldsymbol{u}) = 0$$



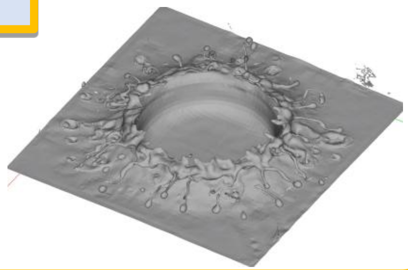


Blood coagulation



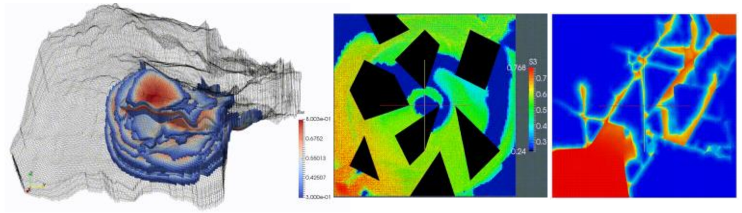
Freesurface flows

$$\frac{\partial \varphi}{\partial t} + \operatorname{div}(\varphi \boldsymbol{u}) = 0$$
$$|\nabla \varphi| = 1$$



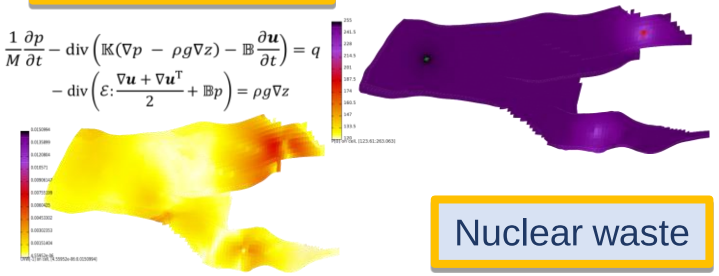
Multiphase filtration

$$\frac{\partial \rho_w \theta S_w}{\partial t} - \nabla \cdot \left(\lambda_w \mathbb{K}(\nabla p - \rho_w g \nabla z)\right) = q_w$$
$$\frac{\partial \rho_o \theta S_o}{\partial t} - \nabla \cdot \left(\lambda_o \mathbb{K}(\nabla p - \nabla P c_o - \rho_w g \nabla z)\right) = q_o$$
$$\frac{\partial \rho_g \theta (R S_o + S_g)}{\partial t} - \nabla \cdot \left(\lambda_g \mathbb{K}(\nabla p - \nabla P c_g - \rho_g g \nabla z)\right) = q_g$$

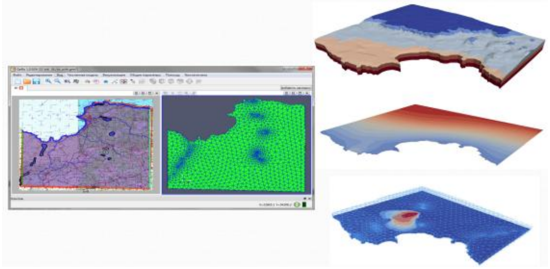


Poromechanics

$$\frac{1}{M}\frac{\partial p}{\partial t} - \operatorname{div}\left(\mathbb{K}(\nabla p - \rho g \nabla z) - \mathbb{B}\frac{\partial \boldsymbol{u}}{\partial t}\right) = q$$
$$-\operatorname{div}\left(\mathcal{E}: \frac{\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T}{2} + \mathbb{B}p\right) = \rho g \nabla z$$

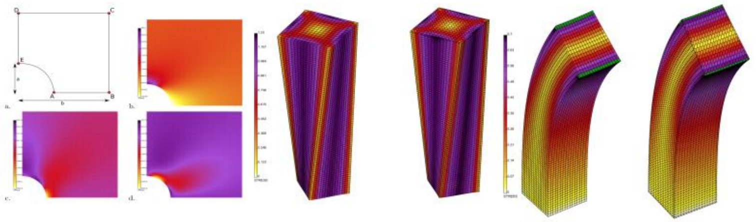


Nuclear waste

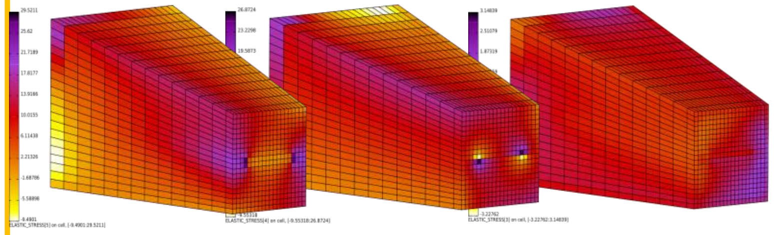


Mechanics

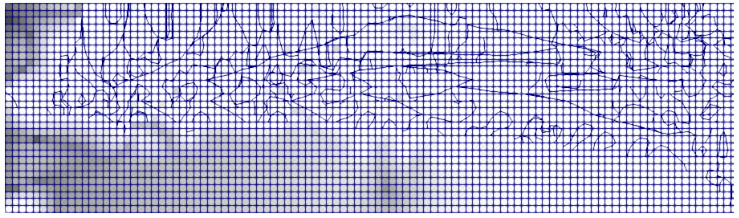
$$-\operatorname{div}(\boldsymbol{\sigma}) = 0, \quad C\colon \boldsymbol{\sigma} = \frac{\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T}{2}$$



Contact mechanics



Fracturing



Large problems

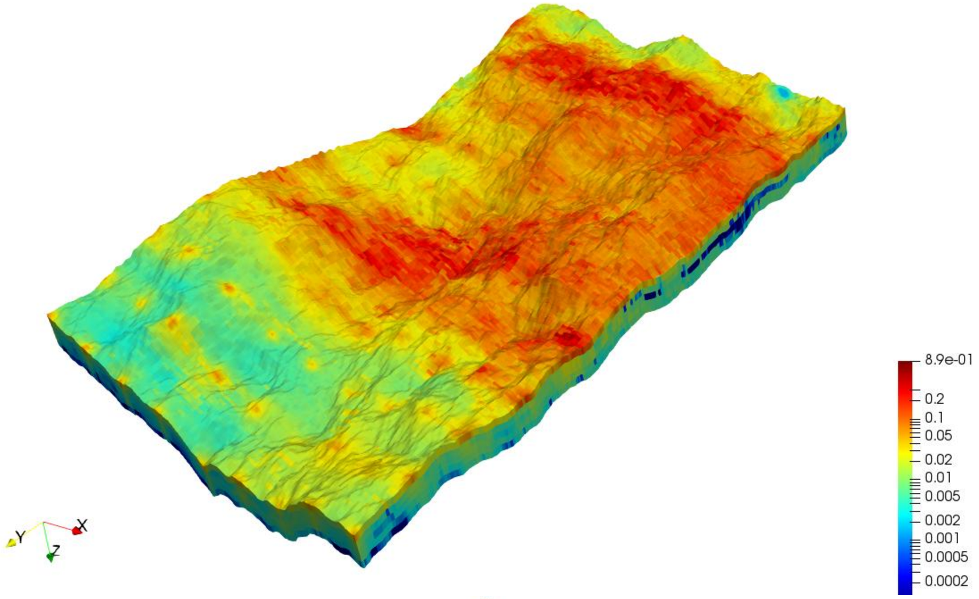
- Suitable for large problem solutions:
  - Black oil problem (detailed further)
  - 3x unknowns per cell
  - **100M** and **200M** cells (320 cores, INM RAS cluster):

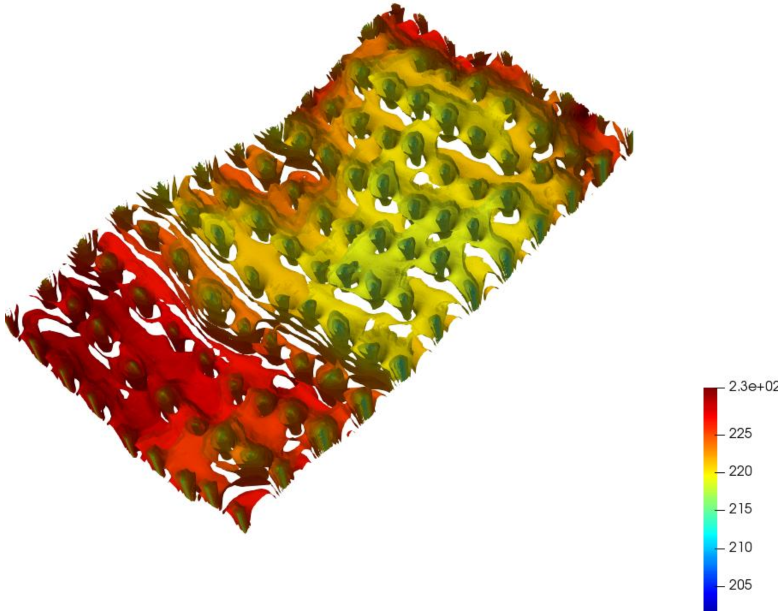
Case	T_{mat}	T_{prec}	T_{iter}	T_{sol}	T_{upd}	N_n	N_l
SPE10_100M	14	18.5	55.4	78.6	0.2	402	3.5
SPE10_200M	29.6	34.7	64.1	107.5	0.38	428	3.96

  - **Peak memory** per core (**significant issue!**):

Case	M_{grid}	M_{mat}	M_{prec}	M_{tot}
SPE10_100M	856.7	165.6	558.4	1943.6
SPE10_200M	1624	346.5	1054.3	4365.6

- Scaled up to **1B of cells** on 9600 Cray cores by Ahmad Abushaika and Longlong Li at HBKU, Qatar.





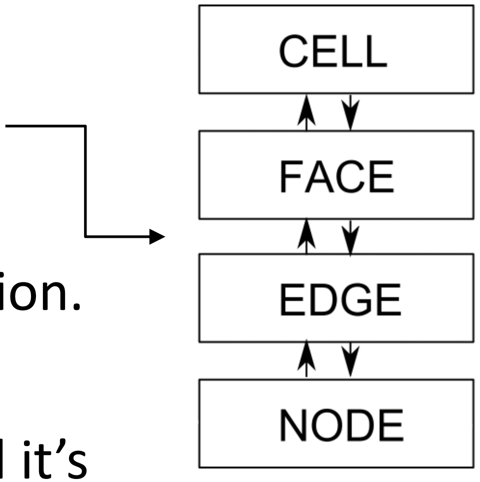
Used Functionality

- **Mesh**:
  - Distributed mesh and data I/O, migration and balancing of a mesh, assembly of ghost layers, parallel organization of elements into hierarchy of sets, custom ghost cells.
- **Partitioning:** standalone K-means clustering.
- **Management of coupled problems:**
  - Automatic differentiation**,** management of unknowns, evaluation, storage and communication of an expression results with the derivatives, operations on the residual vector with an automatic assembly of Jacobian.
- **Linear solver:** biconjugate stabilized gradient with the preconditioner based on additive Schwarz method with multilevel incomplete factorization.

Issues and Limitations

Mesh memory issues:

- **New** adjacency graph
- Connections are **not needed** during simulation.
- Two-way dependency represents a **graph** and it's **transpose**.
- **Allow to remove** and restore connections.



Solver memory issues:

- **Isolated** all memory usage.

MPI Limitations:

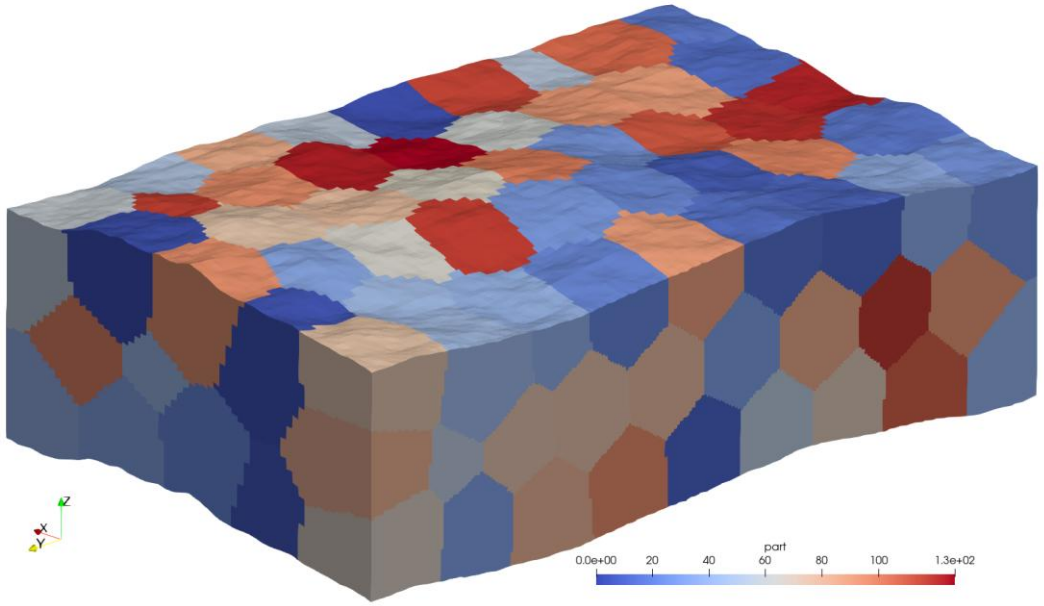
- Complex data storage due to dynamic grids requires intermediate **binary buffers** for communications and I/O.
- Only **2GB** buffer size for ***MPI_BYTE*** type.
- All communications and I/O are **split** into **2GB** chunks of buffers, no large collective operations.

Datatype limitation:

- **Allow to set** a custom integer and floating-point datatypes.

Standalone K-means Clustering

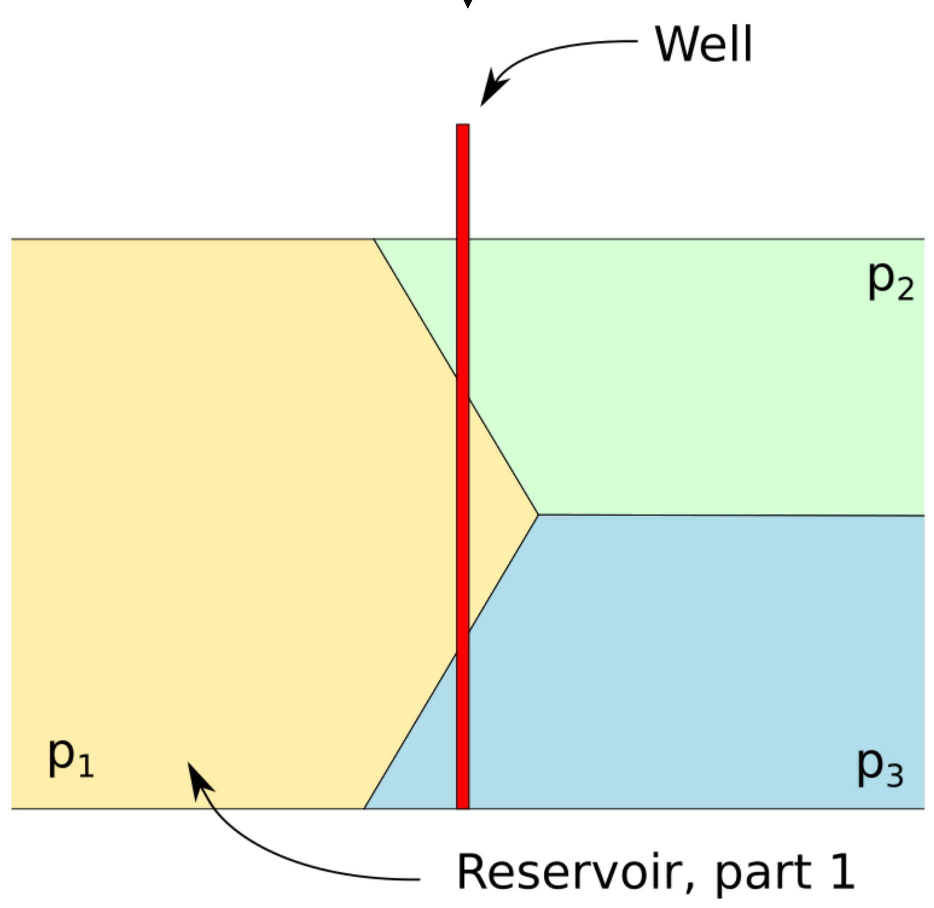
- Mesh is too large for a single node:
  - Memory **issues**.
  - **Very long** startup time.
- Mesh partitioning preprocessor:
  - Perform **multiple** read passes over file if file does not fit memory.
  - Compute and store **only** cell centers.
  - Perform K-means clustering.
  - Process **only** geometry for each part.
  - Process **only** data for each part.
- OpenMP-**parallel** ☺
- **https://github.com/kirill-terekhov/vtu2pvtu**



Wells in Parallel

Partitioner **does not account** for **wells**:

- **Flux-controlled** well requires a pressure **unknown**.
- This unknown contributes to the equation of each well **completion**.
- Each well is represented by a **distributed set** of completions.
- All completions should be **collected** on the owner-processor of the set.
- **Requires** specific solvers or change in calculation of Schwarz overlapping layers.

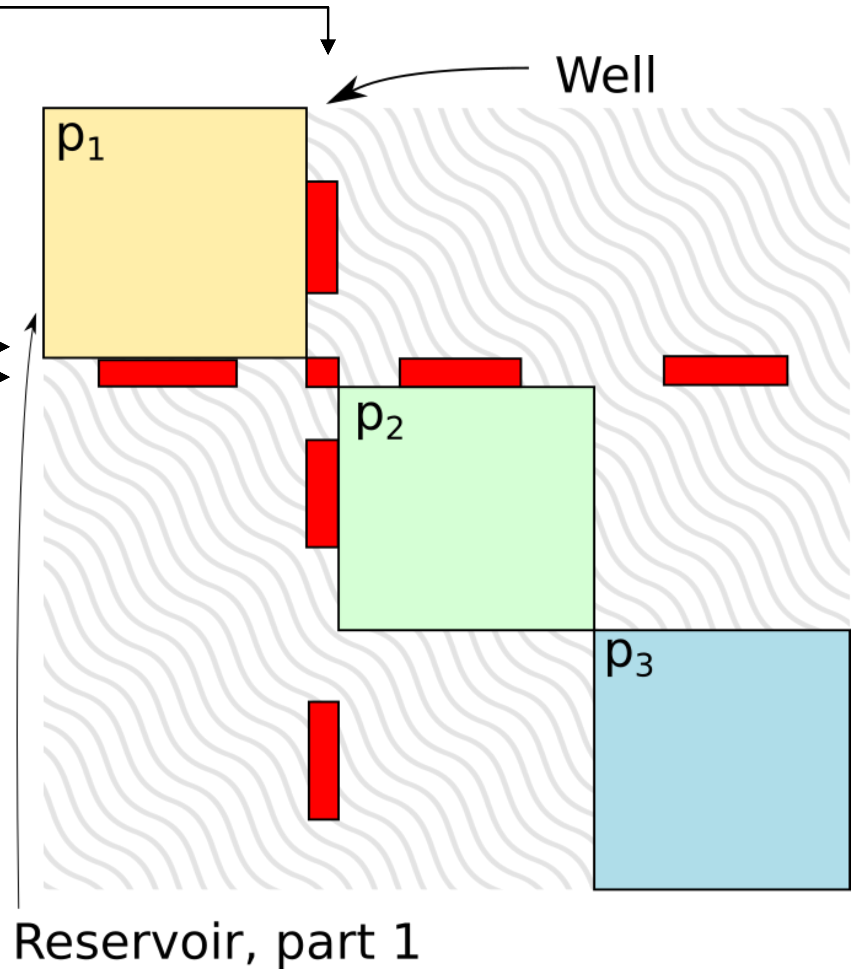


Geometrical reservoir representation

Wells in Parallel

Partitioner **does not account** for **wells**:

- **Flux-controlled** well requires a pressure **unknown**.
- This unknown contributes to the equation of each well **completion**.
- Each well is represented by a **distributed set** of completions.
- All completions should be **collected** on the owner-processor of the set.
- **Requires** specific solvers or change in calculation of Schwarz overlapping layers.

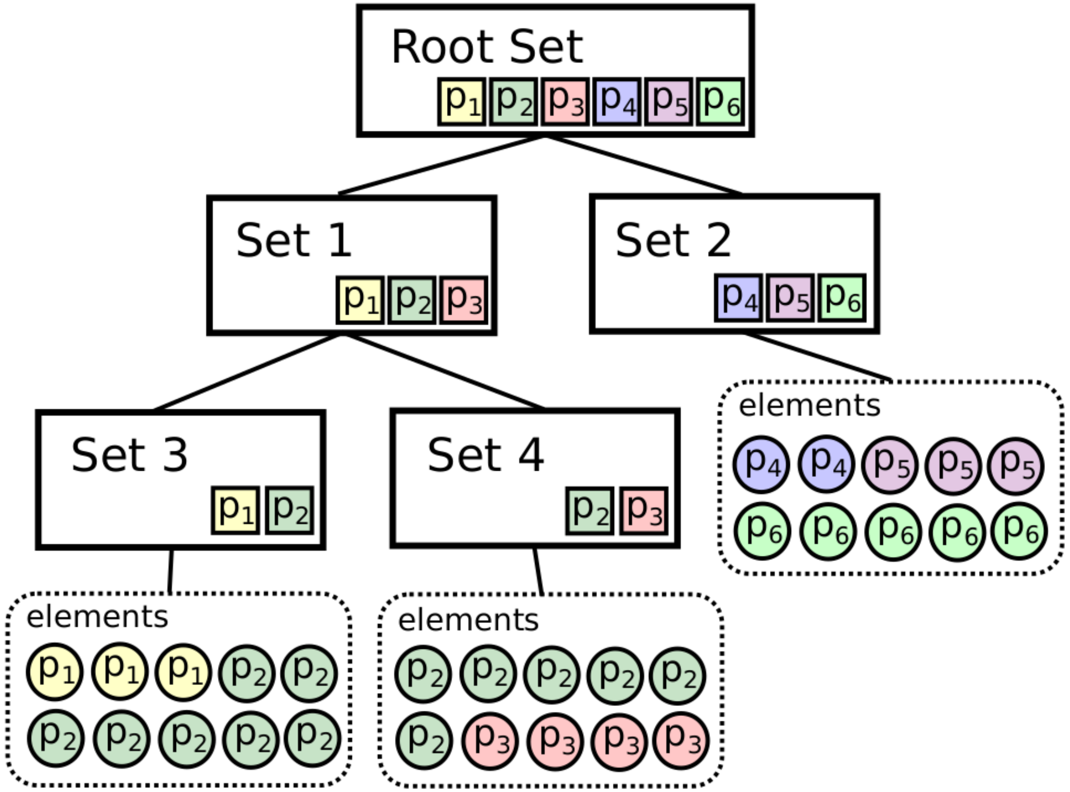


Jacobian reservoir representation

Wells in Parallel

Partitioner **does not account** for **wells**:

- **Flux-controlled** well requires a pressure **unknown**.
- This unknown contributes to the equation of each well **completion**.
- Each well is represented by a **distributed set** of completions.
- All completions should be **collected** on the owner-processor of the set.
- **Requires** specific solvers or change in calculation of Schwarz overlapping layers.



Linear Solver

- Preconditioned **BiCGStab(l)** method[1].
- Preconditioner **MPI-parallelization** using **Additive Schwarz Method**.
- Preconditioner **OpenMP-parallelization** using **Bordered Block-Diagonal Form**[2,3]. **(New, next talk!)**
- **Multi-level** preconditioner with **deferred pivoting** based on the second-order **Crout-ILU**[4,5].
- **Condition estimation** of the inverse factors determines the **coarse system** and tunes dropping tolerances[6,7].
- **Scaling** and **reordering** of the local system before **factorization**[8,9,10].

References

1) Sleijpen, G.L.G., Diederik R. F.: ***BiCGstab (l) for linear equations involving unsymmetric matrices with complex spectrum.*** **Electronic Transactions on Numerical Analysis** 1.11 (1993): 2000**.** **(Krylov method)**

2) Grigori, L., Boman, E. G., Donfack, S., Davis, T. A: ***Hypergraph-based unsymmetric nested dissection ordering for sparse LU factorization***. **SIAM Journal on Scientific Computing**, *32*.6 (2010): 3426-3446. **(Bordered block-diagonal form)**

3) Duff, I. S., Scott, J. A.: ***Stabilized bordered block diagonal forms for parallel sparse solvers*****. Parallel Computing**, 31.3-4 (2005): 275-289. **(Bordered block-diagonal form)**

4) Li N., Saad Y., Chow E.: ***Crout versions of ILU for general sparse matrices.*** **SIAM Journal on Scientific Computing** 25.2 (2003): 716-728. **(Crout-ILU)**

5) Kaporin, I.E.: ***High quality preconditioning of a general symmetric positive definite matrix based on its UTU+ UTR+ RTU-decomposition*****. Numerical linear algebra with applications** 5.6 (1998): 483-509. **(Second-order ILU)**

6) Bollhöfer, M.: ***A robust ILU with pivoting based on monitoring the growth of the inverse factors*****. Linear Algebra and its Applications** 338.1-3 (2001): 201-218. **(Tuning dropping tolerances)**

7) Bollhöfer, M., Saad Y.: ***Multilevel preconditioners constructed from inverse-based ILUs*****. SIAM Journal on Scientific Computing** 27.5 (2006): 1627-1650. **(Computing coarse system)**

8) Cuthill, E., McKee J.: ***Reducing the bandwidth of sparse symmetric matrices.*** **Proceedings of the 1969 24th national conference.** 1969. **(Reordering for bandwidth reduction)**

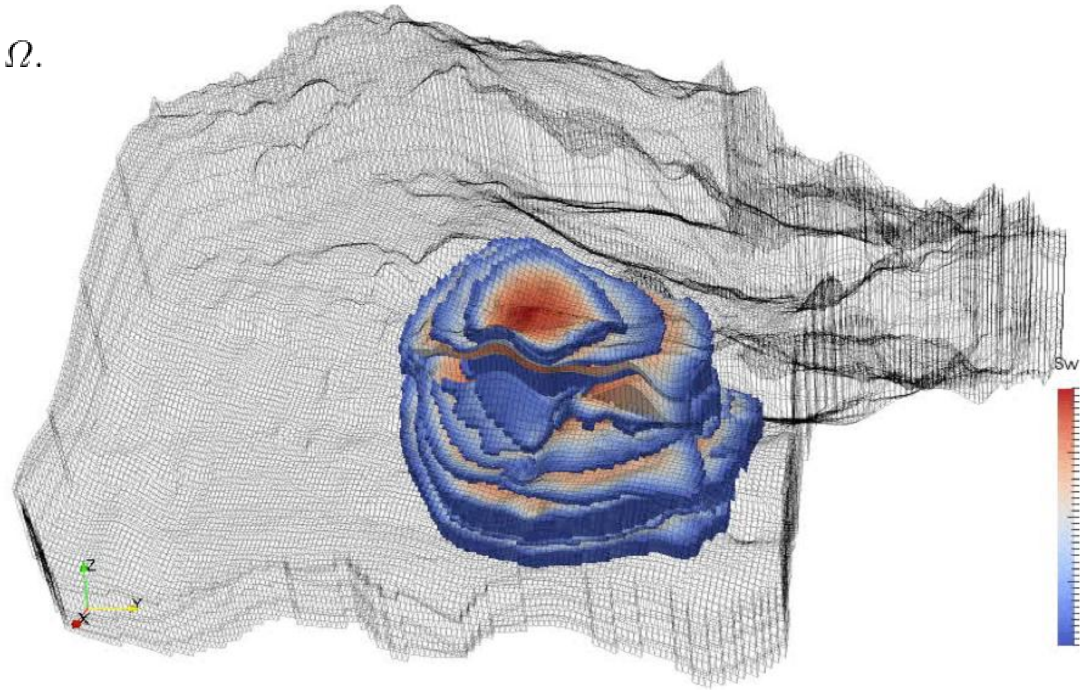
9) Olschowka, M., Arnold N.: ***A new pivoting strategy for Gaussian elimination*****. Linear Algebra and its Applications** 240 (1996): 131-151. **(Maximizing diagonal product)**

10) Kaporin, I.E.: ***Scaling, reordering, and diagonal pivoting in ILU preconditionings*****. Russian Journal of Numerical Analysis and Mathematical Modelling** 22.4 (2007): 341-375. **(Rescaling for condition reduction)**

Black-Oil Model

System of equations:

$$\begin{cases} \dfrac{\partial}{\partial t}\left(\dfrac{\phi S_w}{B_w}\right) - \operatorname{div}\left(\lambda_w \mathbb{K}\left(\nabla p_w - \rho_w g \nabla z\right)\right) = q_w, \\ \dfrac{\partial}{\partial t}\left(\dfrac{\phi S_o}{B_o}\right) - \operatorname{div}\left(\lambda_o \mathbb{K}\left(\nabla p_o - \rho_o g \nabla z\right)\right) = q_o, \\ \dfrac{\partial}{\partial t}\left(\phi \dfrac{S_g}{B_g} + \phi R_s \dfrac{S_o}{B_o}\right) - \operatorname{div}\left(\lambda_g \mathbb{K}\left(\nabla p_g - \rho_g g \nabla z\right)\right) \\ \qquad - \operatorname{div}\left(R_s \lambda_o \mathbb{K}\left(\nabla p_o - \rho_o g \nabla z\right)\right) = q_g + R_s q_o, \end{cases} \quad \text{in } \Omega.$$



Additional relations:

- $S_w + S_o + S_g = 1$,
- $p_w = p_o - P_{cow}(S_w), \quad p_g = p_o + P_{cog}(S_g)$,
- $\phi(p_o) = \phi^0\left(1 + C_R(p_o - p^0)\right)$.

and others... **(see paper)**

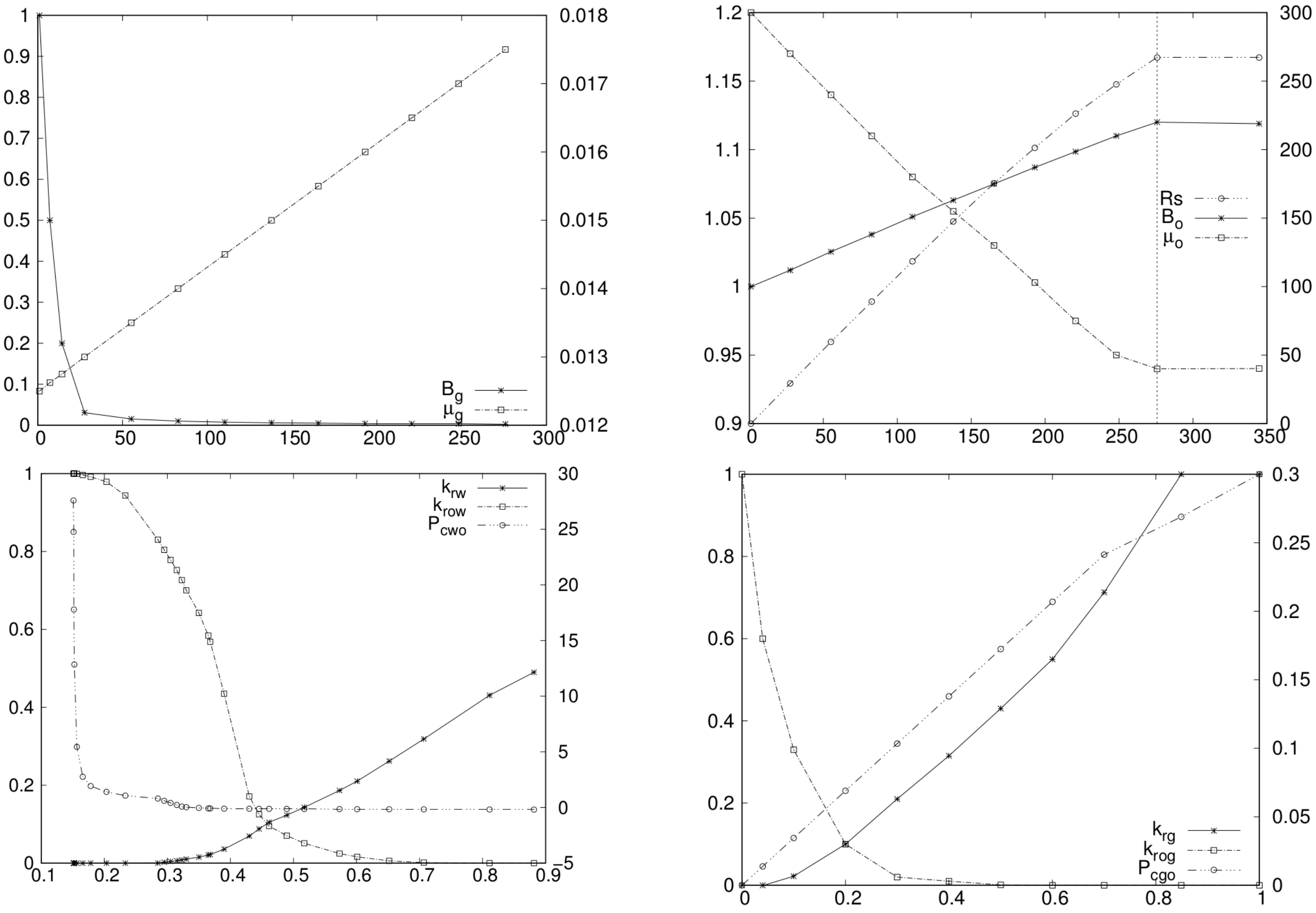
Black-Oil Model

Nonlinear correlations:

- viscosities,
- volume formation factors,
- gas solubility,
- relative permeabilities,
- capillary pressures.

Stones’ II model for oil relative permeability.

Phase state **switching**.



Wells Equation

Well is a source/sink:

$$q_\alpha = \lambda_\alpha WI \left(p_{bh} - p_o - \rho_\alpha g(z_{bh} - z)\right) \delta(\mathbf{x} - \mathbf{x}_0),$$

Peaceman formula:

$$WI = \frac{2\pi h_z \sqrt{k_x k_y}}{\log\left(\frac{0.28}{r_w}\frac{\sqrt{h_x^2 (k_y/k_x)^{1/2} + h_y^2 (k_x/k_y)^{1/2}}}{(k_y/k_x)^{1/4} + (k_x/k_y)^{1/4}}\right) + s}.$$

Using upstream for mobility based on flow direction, to prevent crossflow instability.

Single rate-controlled well, and four BHP wells. Equation for the rate-controlled well:

$$\sum_{\alpha\in\{o,w,g\}} \sum_{i=1}^{N} q_\alpha^i = \sum_{\alpha\in\{o,w,g\}} \sum_{i=1}^{N} \lambda_\alpha^i WI^i \left(p_{bh} - p_o^i - \rho_\alpha^i g(z_{bh} - z^i)\right) = q_{tot}$$

CPR Matrix Scaling

Pre-scaling the system to zero-out off-diagonal block:

$$\begin{bmatrix} A_{pp} & A_{ps} \\ A_{sp} & A_{ss} \end{bmatrix} \cdot \begin{bmatrix} p \\ s \end{bmatrix} = \begin{bmatrix} b_p \\ b_s \end{bmatrix} \cdot$$

$$\begin{bmatrix} B_{pp} & Z_{ps} \\ A_{sp} & A_{ss} \end{bmatrix} \cdot \begin{bmatrix} p \\ s \end{bmatrix} = \begin{bmatrix} b_p - D_{ps} D_{ss}^{-1} b_s \\ b_s \end{bmatrix}$$

where

$$B_{pp} \equiv A_{pp} - D_{ps} D_{ss}^{-1} A_{ps}$$

$$Z_{ps} \equiv A_{ps} - D_{ps} D_{ss}^{-1} A_{ss} \approx 0$$

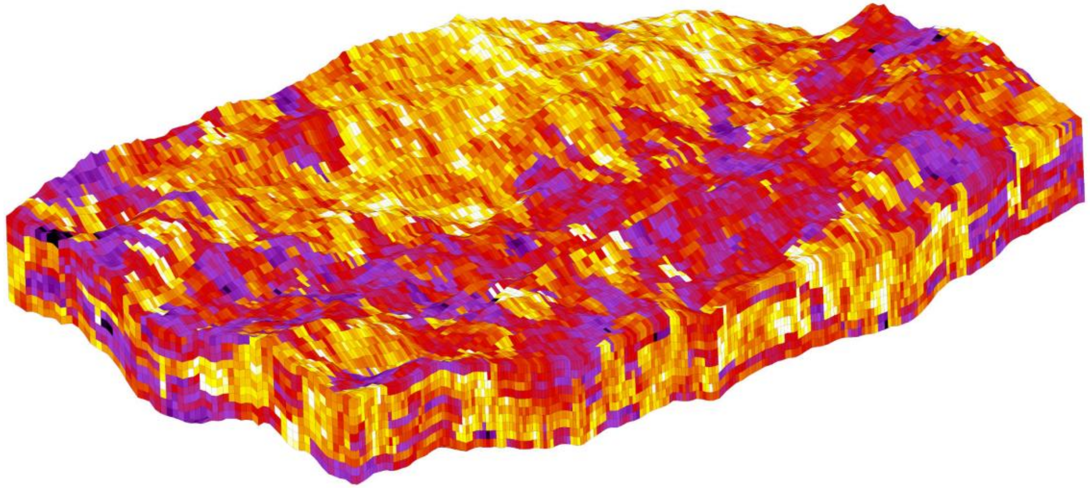
and scaling matrices are computed from the derivatives in the Jacobian. **(see paper)**

Some related data from S^3M package and two-phase problem:

		MT-ILUC2	CPR-MT-ILUC2	CPR-TS	CPR-TSGS
spe10 (tpfa)	T	**283**	187	92	**46**
size 2 244 000	Ts	**99**	63	71	**16**
nnz 31 120 000	Tit	**184**	123.7	**20**	29
	Nit	**405**	356	**38**	76
	Lvl	—	—	10	10
	Mem	2.5 GB	2.2 GB	**2.6** GB	1.4 GB
spe10 (tpfa)	T	**4332**	2940	1067	**97** 5
size 20 196 000	Ts	**687**	522	597	**150**
nnz 281 222 400	Tit	**3645**	2417	**470**	825
	Nit	**1065**	799	**93**	225
	Lvl	—	—	12	12
	Mem	21 GB	19 GB	**22** GB	**13** GB
spe10 (tpfa)	T	**20857**	20276	**3758**	3976
size **60 588 000**	Ts	**1693**	1440	1564	**466**
nnz **845 568 000**	Tit	**19164**	18836	**2194**	2510
	Nit	2156	**2241**	**150**	321
	Lvl	—	—	12	12
	Mem	54 GB	52 GB	**63** GB	**39** GB

SPE10 Dataset

- **Open-source** reservoir data.
- Regular 60x220x85 = 1 112 000 entries of permeability and porosity.
- The grid is refined, vertically distorted and the permeability is rotated to a full-tensor according to distortion. **(see paper)**



- https://github.com/kirill-terekhov/spe10grdecl

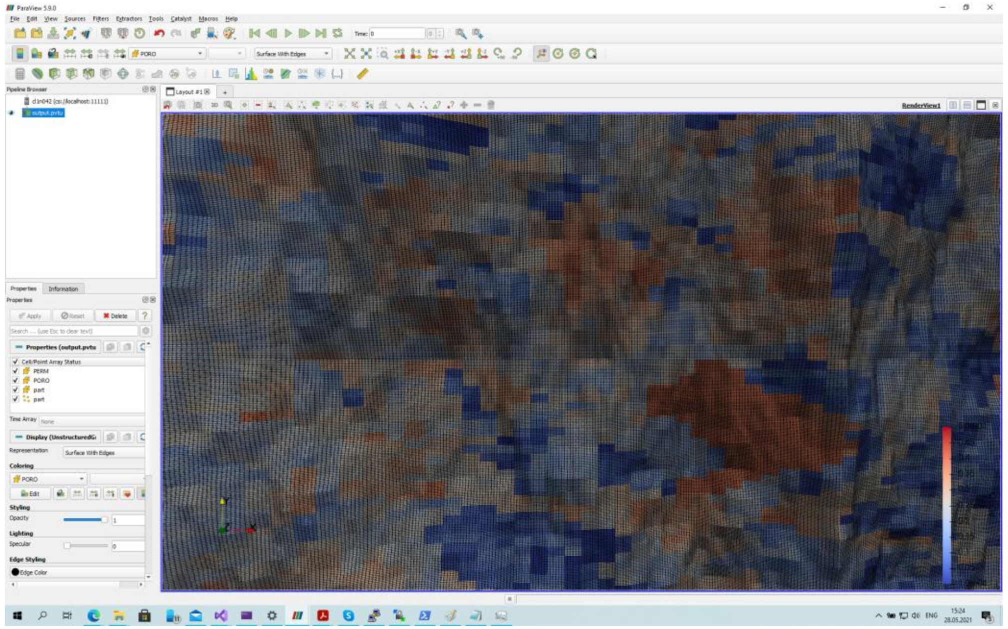
Results

- Suitable for large problem solutions:
  - Black oil problem (detailed further)
  - 3x unknowns per cell
  - **100M** and **200M** cells (320 cores, INM RAS cluster):

Case	T_{mat}	T_{prec}	T_{iter}	T_{sol}	T_{upd}	N_n	N_l
SPE10_100M	14	18.5	55.4	78.6	0.2	402	3.5
SPE10_200M	29.6	34.7	64.1	107.5	0.38	428	3.96

  - **Peak memory** per core (**significant issue!**):

Case	M_{grid}	M_{mat}	M_{prec}	M_{tot}
SPE10_100M	856.7	165.6	558.4	1943.6
SPE10_200M	1624	346.5	1054.3	4365.6



Zoom into mesh

Conclusions and future directions

- INMOST is capable of solving huge problems.
- Future directions:
  - **Reduce** memory consumed by overlapping using hybrid MPI-OpenMP. (**Next talk!**)
  - Consider more **large** complex problems: Navier-Stokes, Geomechanics, Maxwell, etc.
  - **Specific** solvers for densely populated matrix rows originating from wells and integral constraints.
  - **Parallel** visualization tools.
  - We are working on a very flexible linear solvers framework S^3M to address multiphysics problems in parallel:
    - Konshin, I., Terekhov, K.: ***Sparse System Solution Methods for Complex Problems*. In International Conference on Parallel Computing Technologies,** (2021, September): 53-73. Springer, Cham.

Supported by RSF grant №21-71-20024

Thank you for your attention

Contacts

- IGOR.KONSHIN@GMAIL.COM
- KIRILL.TEREHOV@GMAIL.COM

Links

- WWW.INMOST.ORG
- WWW.INMOST.RU