Комплекс программ INMOST:
МКО дискретизации, реакции, и
методы решения линейных систем

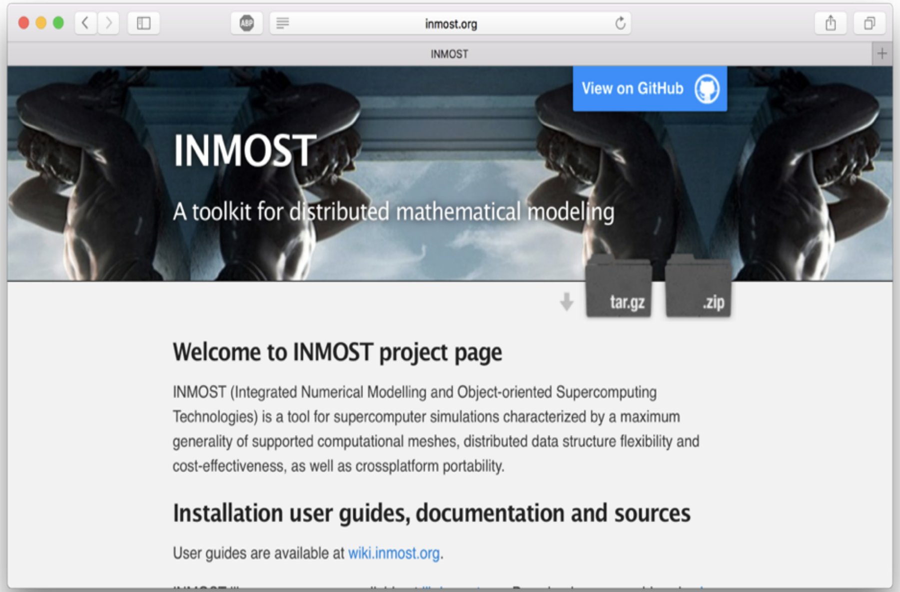
Sparse System Solution Methods (S^3M)

Кирилл М. Терехов[1,2], Иван Д. Бутаков[2], Игорь Н. Коньшин[1,2]

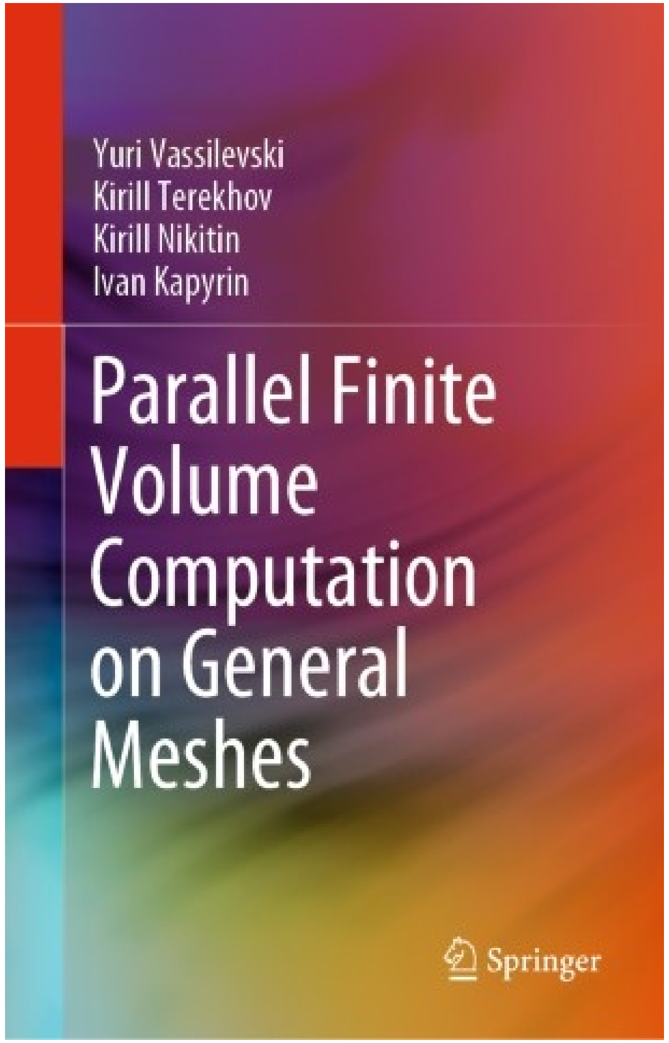
[1] Институт вычислительной математики им. Г.И.Марчука РАН
[2] Московский физико-технический институт

CFD Weekend
18 декабря, 2021

INMOST


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
Суперкомпьютерный консорциум университетов России


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

Флагманские линейные решатели INMOST

- Preconditioned **BiCGStab(l)** method[1].
- Preconditioner parallelization using **Additive Schwarz Method**.
- **Multi-level** preconditioner based on the second-order **Crout-ILU**[2,3].
- **Condition estimation** of the inverse factors determines the **coarse system** and tunes dropping tolerances[4,5].
- **Scaling** and **reordering** of the local system on each successive level[6,7,8].

Литература

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) 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)**

3) 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)**

4) 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)**

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

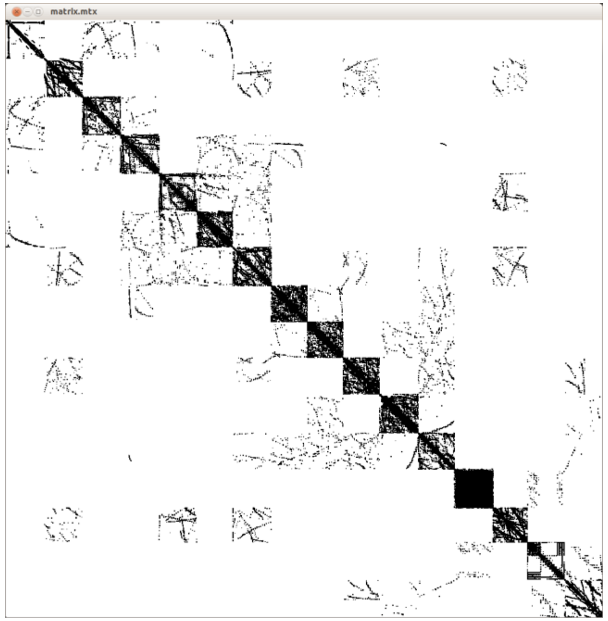
6) Cuthill, E., McKee J.: ***Reducing the bandwidth of sparse symmetric matrices.*** **Proceedings of the 1969 24th national conference.** 1969. **(Reordering for bandwidth reduction)**

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

8) 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)**

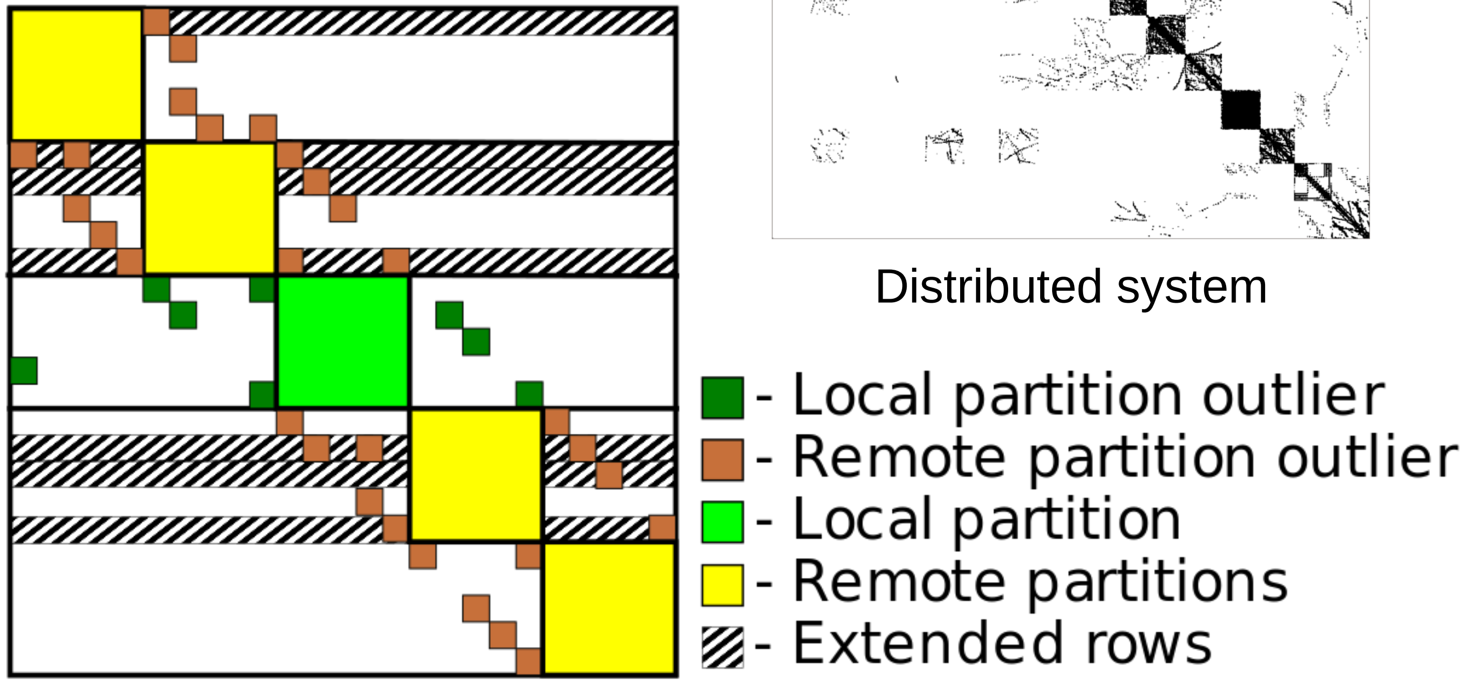
Additive Schwarz Method (AS)

- Global matrix is composed of local **blocks**.
- Extend blocks to localize the **connection**.
- **Restricted** version.



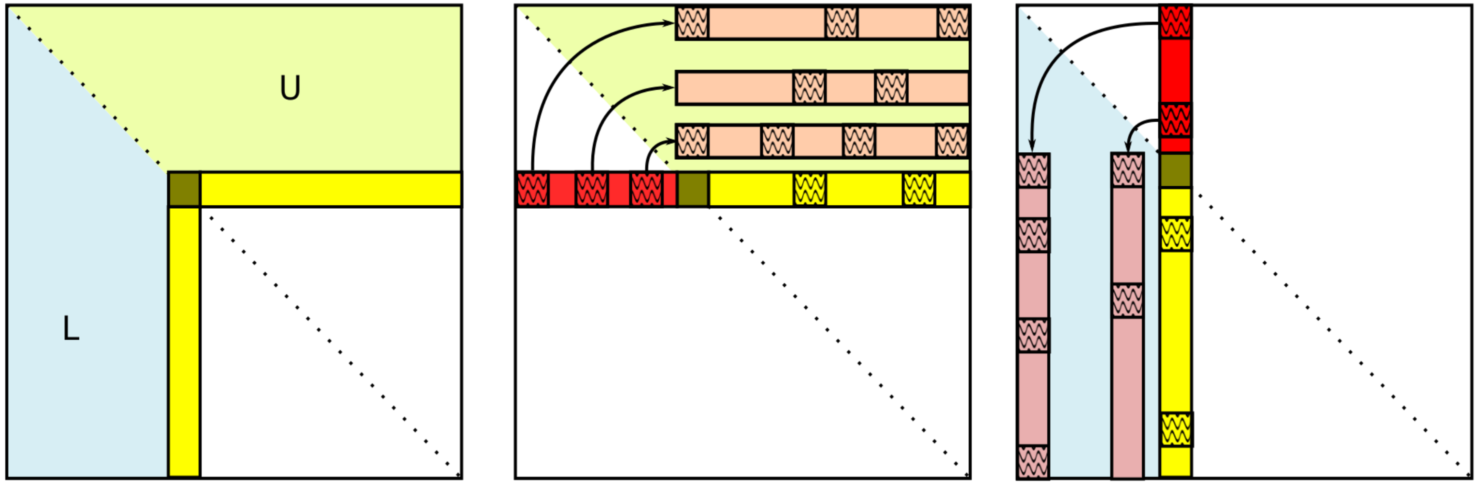
Distributed system

Matrix overlapping



Second-order Crout Incomplete LU (ILUC2)

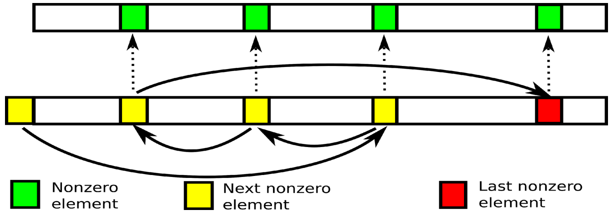
- Dual-threshold dropping:
  - τ^2 for factorization.
  - τ for iterations.
- Running condition estimation:
  - $\kappa = \max(\|L^{-1}\|, \|U^{-1}\|)$
  - $\tau/\kappa = \text{const}$ tuning.
  - Limit growth of κ.



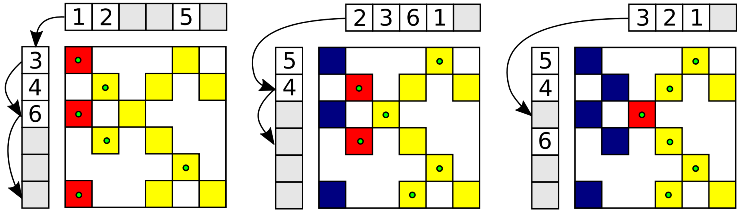
U-factor elimination

L-factor elimination

Dense row accumulator:

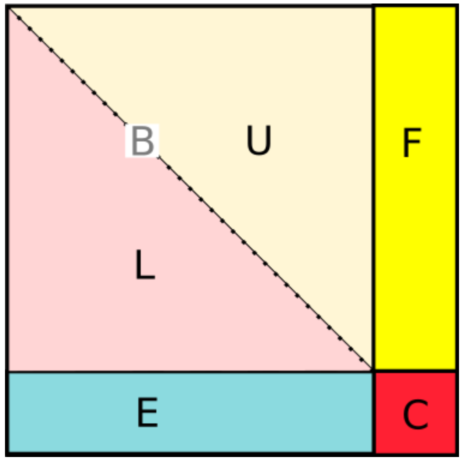


Transposed matrix traversal:

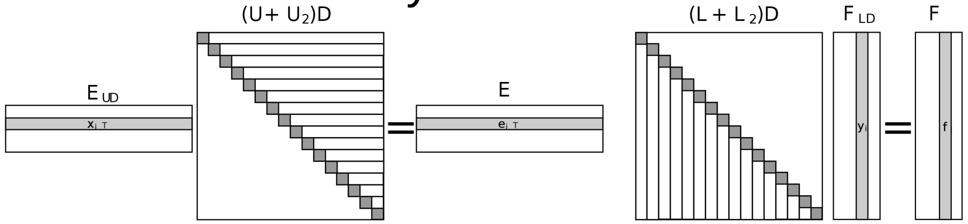


Schur Complement

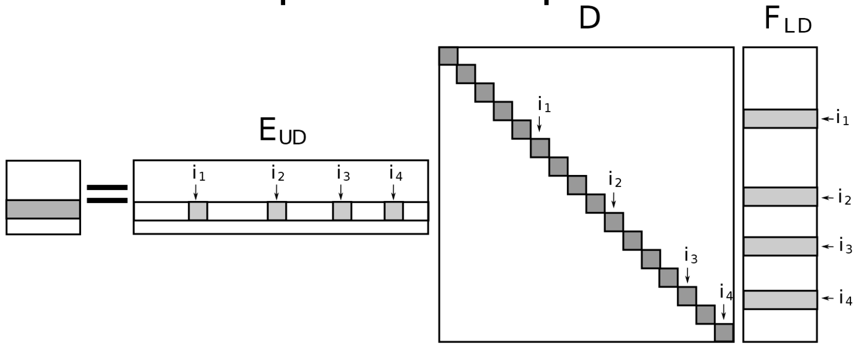
- Part that leads to growth of κ is accumulated in **C**:
  - system reordering after factorization.
- Next level system is the Schur complement:
  - $\mathbf{S} = \mathbf{C} - \mathbf{E}\,(DU)^{-1}\,D(LD)^{-1}\mathbf{F}$.
  - Requires forward and backward substitution with sparse right hand side.
  - Fill-in control is critical.



Partially factorized matrix.



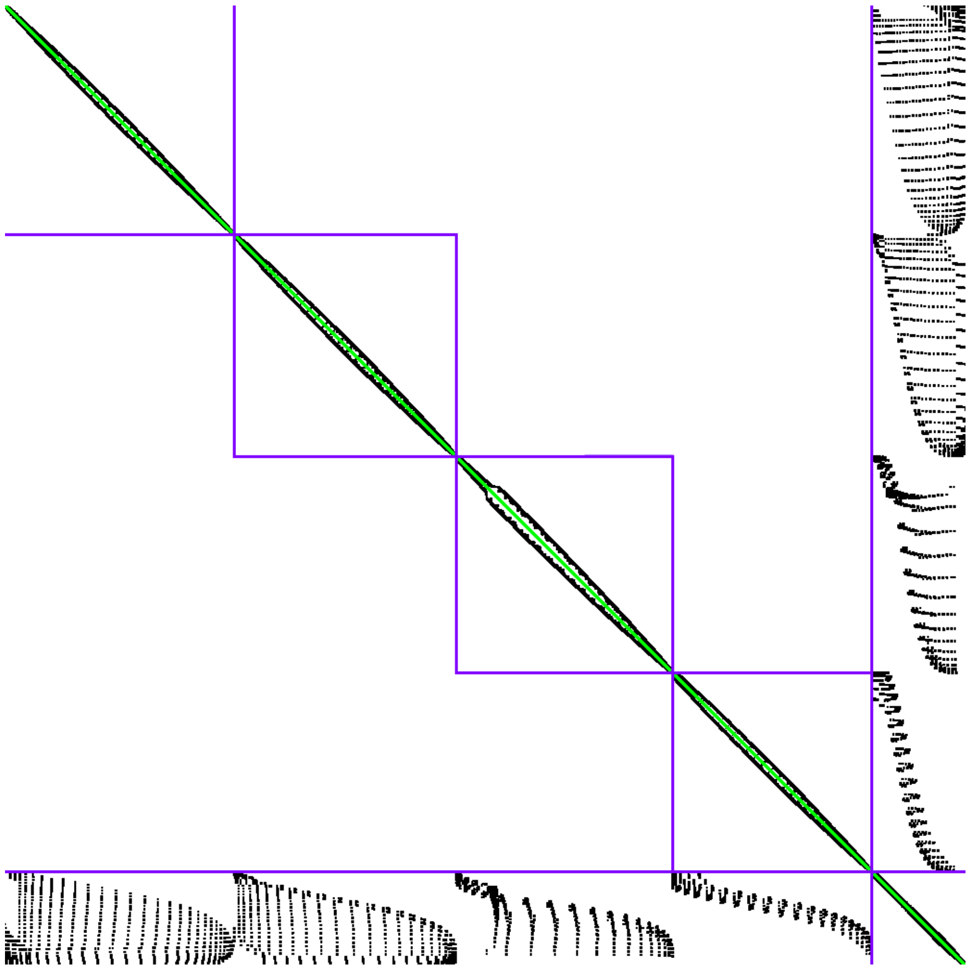
Computation of operators



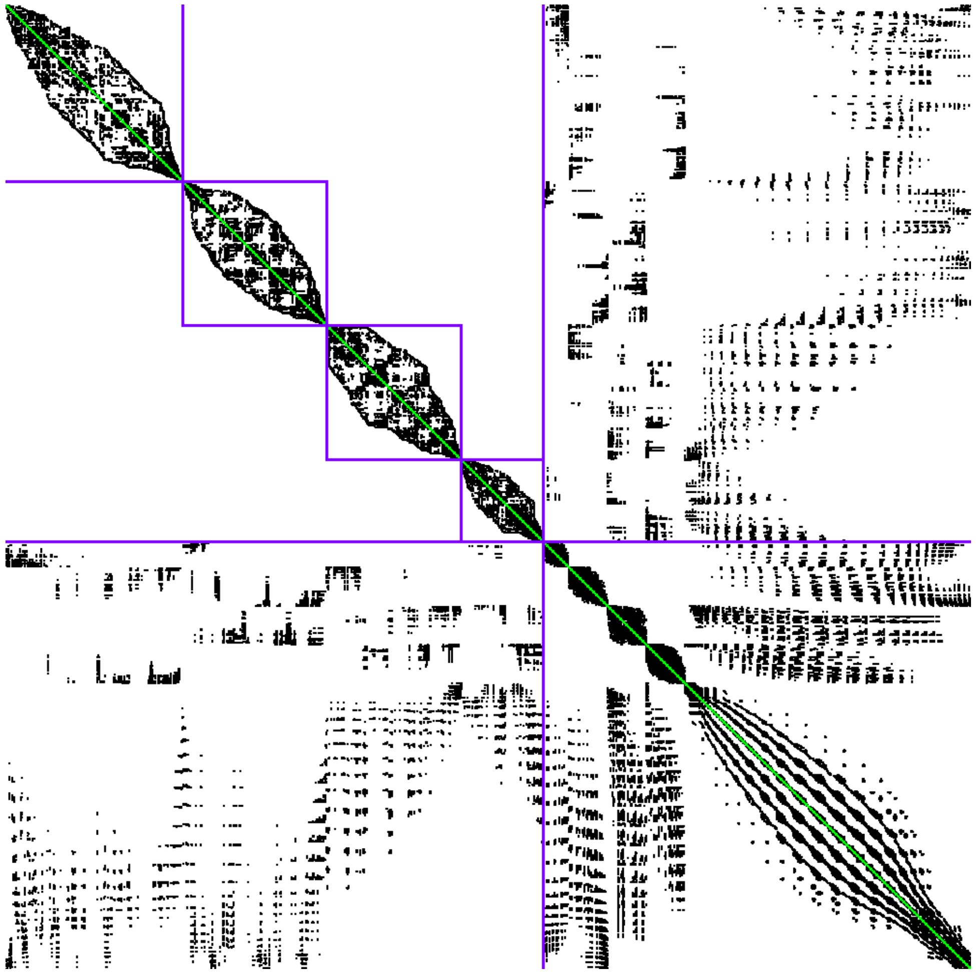
Schur complement computation

Doubly-Bordered Block-Diagonal Form

First level



Schur complement



Аналогия с Algebraic Multigrid (AMG)

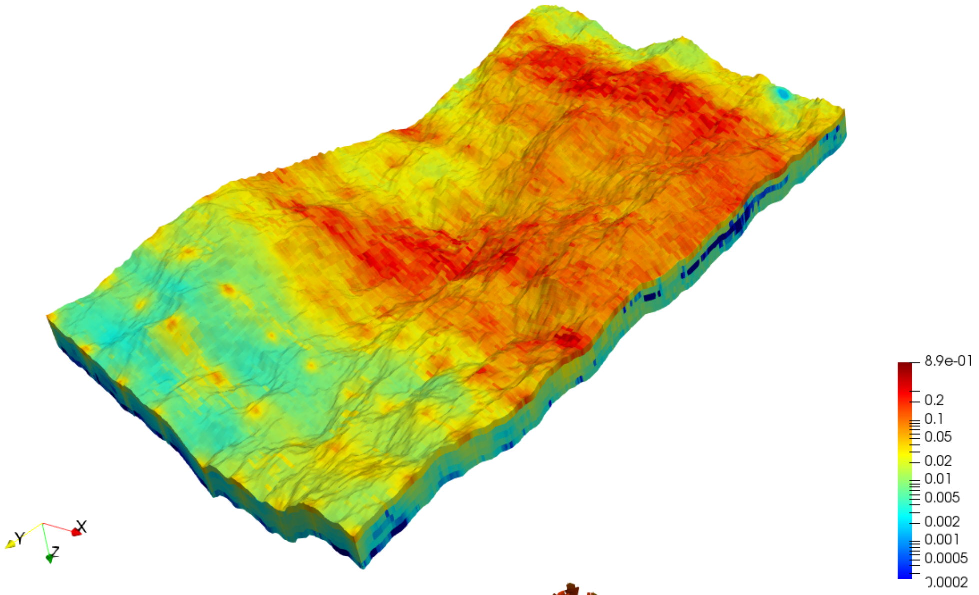
- **Coarse** system should contain the **largest error** of the **smoother**.
- Condition estimation reveals the **error** in the **smoother** and provides the ***coarse-fine splitting*** of the system.
- Ideal prolongation $P=(-EB^{-1},I)$ and restriction $R=(-FB^{-1},I)^T$.
  - (not satisfied by the present method).
- Schur complement corresponds to the **coarse** system.

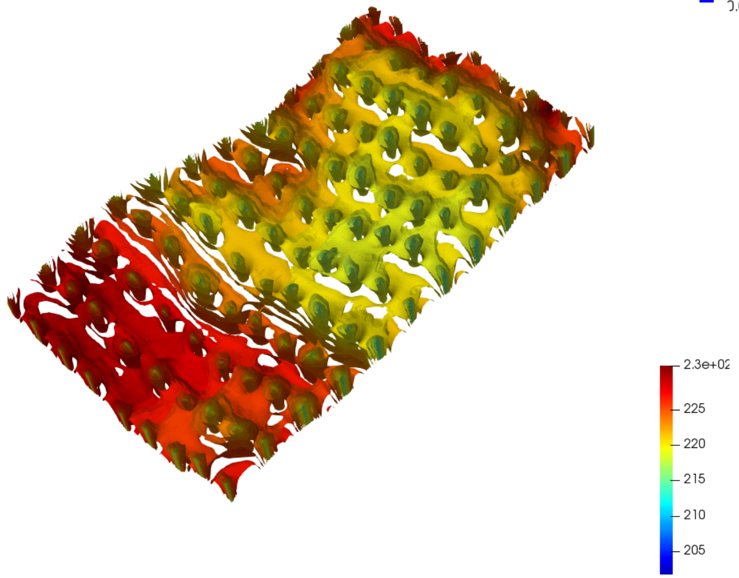
- **Universal but much more computationally complex.**
  - (definitely not linear computational complexity)

Большие задачи

- Suitable for large problem solutions:
  - Black oil problem
  - 3x unknowns per cell
  - **100M** and **200M** cells (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





- Scaled upto **1B of cells** on 9600 Cray cores by Ahmad Abushaika at HBKU, Qatar.
- Optimal preconditioner is **CPR+AMG**.

Sparse System Solution Methods (S^3M)

Concept:

- Basic operations:
  - Operations for vectors and sparse matrix-vector
  - Sparse matrix-matrix addition and multiplication, substitutions with triangular systems
  - Hidden **OpenMP** and **MPI**
    - **+ OpenMP target**
- **Portability** and ease of use:
  - MIPT practical course, SIRIUS lections

```
while( resid > ftol && resid < dtol && iters < maxiters+1 )
{
    w = A*p;
    alpha = kappa/Dot(w,p);
    x += alpha*p;
    r -= alpha*w;
    ApplyPreconditioner(r,z);
    beta = 1.0/kappa;
    kappa = Dot(r,z);
    beta *= kappa;
    resid = sqrt(fabs(kappa));
    p = z + beta*p;
    iters++;
}
```

PCG Main loop

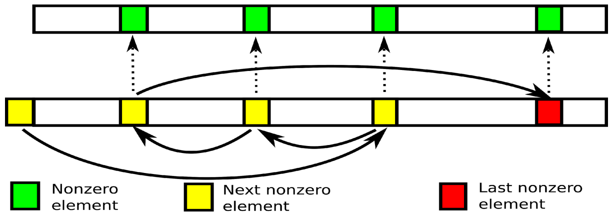
```
do
{
    beta = 1.0 / rho * alpha / omega;
    rho = Dot(r0,r);
    beta*= rho;
    p = r + beta*(p - omega*v);
    ApplyPreconditioner(p,y);
    v = A*y;
    alpha = rho / Dot(r0,v);
    s = r - alpha*v;
    ApplyPreconditioner(s,z);
    t = A*z;
    omega = Dot(t,s)/Dot(t,t);
    x += alpha*y + omega*z;
    r = s - omega*t;
    resid = Norm(r);
    iters++;
} while ( resid > ftol && resid < dtol && omega && iters < maxiters+1);
```

BiCGStab Main loop

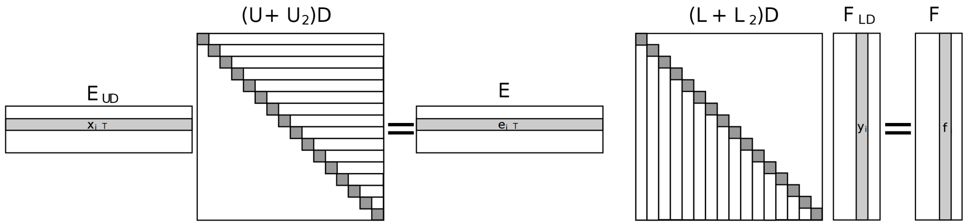
Поддержка операций

Data structures and algorithms

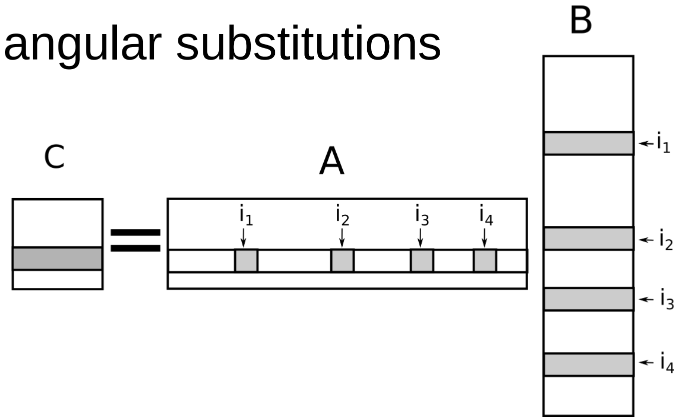
- C++ expression templates for vectors
- C++ operators for sparse matrices: add, subtract, multiply, transpose
- Dense linked lists for **O(1)** operations on sparse vectors (row accumulators)
- Triangular substitutions



Dense row accumulator



Triangular substitutions

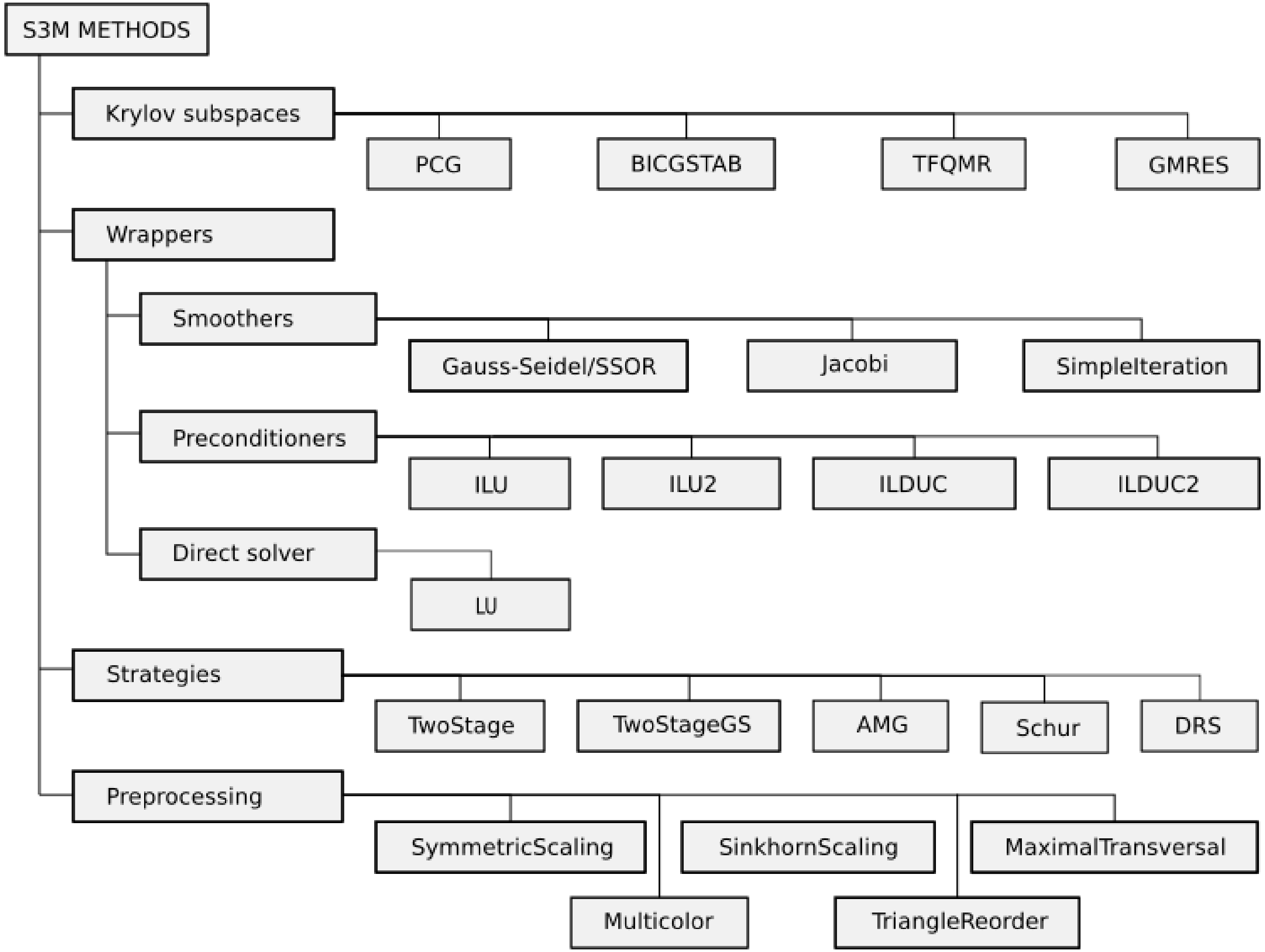


Sparse matrix multiplications (Gustavson)

Sparse System Solution Methods (S^3M)

Concept:

- Combine C++ templates for solver assembly from collections of
  - Iterative methods
  - Preconditioners and smoothers
  - Preprocessors and reorderings
  - Algebraic multigrid methods
  - **Multilevel methods** (new)
  - Multistage strategies



Sparse System Solution Methods (S^3M)

Concept:

- Combine C++ templates for solver assembly from collections of
  - Iterative methods
  - Preconditioners and smoothers
  - Preprocessors and reorderings
  - Algebraic multigrid methods
  - **Multilevel methods** **(new)**
  - Multistage strategies

- `BICGSTAB<ILU2> Solver;`
  - **Iterative solver**: stabilized bi-conjugate gradient method
  - **Preconditioner**: second-order incomplete factorization method
- `SymmetricScaling< PCG<GaussSeidel> > Solver;`
  - **Preprocessor**: symmetric norm equilibration (Sinkhorn scaling)
  - **Iterative solver**: preconditioned conjugate gradient methd
  - **Preconditioner**: Gauss-Seidel method

Sparse System Solution Methods (S^3M)

Concept:

- Combine C++ templates for solver assembly from collections of
  - Iterative methods
  - Preconditioners and smoothers
  - Preprocessors and reorderings
  - Algebraic multigrid methods
  - **Multilevel methods** (new)
  - Multistage strategies

- `PCG< AMG<GaussSeidel,LU> > Solver;`
  - **Iterative solver**: conjugate gradient method
  - **Preconditioner**: algebraic multigrid
  - **Smoother**: Gauss-Seidel
  - **Coarsest level solver**: direct factorization
- `BICGSTAB< CPR< AMG<GaussSeidel,LU>, ILU >, TwoStageGaussSeidel > Solver;`
  - **Multistage strategy:** constrained pressure residual scaling
  - **Pressure preconditioner**: Gauss-Seidel method
  - **Saturations preconditioner**: incomplete factorization

Наборы методов

Iterative methods:

- PCG, BICGSTAB, GMRES, TFQMR

Preconditioners and **smoothers**:

- **parallel:** ConjugateGradient, SimpleIteration, Chebyshev, Jacobi, MulticolorGaussSeidel, **ParILUT** (new)
- **sequential:** GaussSeidel, ILU, ILU2, LU

Complex **preconditioners**:

- AMG, AMGRugeStuben, **MLILU**, **MLILU2** (new), amg1r6

Preprocessors:

- SinkhornScaling, SymmetricScaling, MaximumTransversal (famous MC64), RCM, METIS

Multistage strategies:

- TwoStage, TwoStageGaussSeidel, CPR, DRS, SchurMFD, SchurSeries

Parallelization strategy:

- **BlockJacobi** (new)

Многоуровневые методы

Multistage strategies:

- TwoStage – a way to combine multiple preconditioners and solve $(AM^{-1})(Mx)=b$ with

$$M^{-1} = M_1^{-1} + \sum_{i=2}^{n_{st}} M_i^{-1} \prod_{j=1}^{i-1} (I - AM_j^{-1}),$$

- TwoStageGaussSeidel – use block Gauss-Seidel with individual preconditioner M_1 and M_2:

$$\begin{bmatrix} B & E \\ F & C \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \Longrightarrow \begin{matrix} \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{bmatrix} = \begin{bmatrix} B & \\ F & C \end{bmatrix}^{-1} \left(\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} - \begin{bmatrix} 0 & E \\ & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \right), \\ \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} B & E \\ & C \end{bmatrix}^{-1} \left(\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} - \begin{bmatrix} 0 & \\ F & 0 \end{bmatrix} \cdot \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{bmatrix} \right). \end{matrix}$$

$$\tilde{x}_1 = M_1^{-1}(b_1 - Ex_2), \quad x_2 = M_2^{-1}(b_2 - F\tilde{x}_1), \quad x_1 = M_1^{-1}(b_1 - Ex_2).$$

Многоуровневые методы

Multistage strategies:

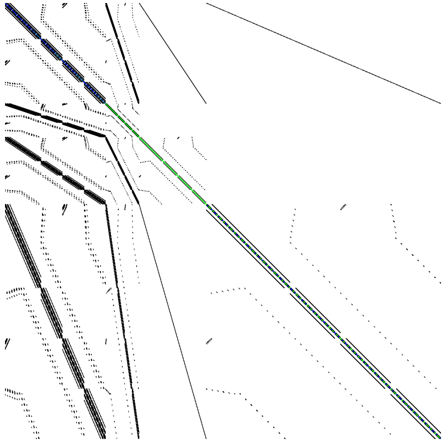
- CPR – constrained pressure residual, DRS – dynamic row scaling
  - Using black-oil problem structure, multiply from the left by a matrix to approximately decouple the pressure system:

$$\begin{bmatrix} A_{pp} & A_{ps} \\ A_{sp} & A_{ss} \end{bmatrix} \cdot \begin{bmatrix} p \\ s \end{bmatrix} = \begin{bmatrix} b_p \\ b_s \end{bmatrix} . \Longrightarrow \begin{bmatrix} B_{pp} & Z_{ps} \\ A_{sp} & A_{ss} \end{bmatrix} \cdot \begin{bmatrix} p \\ s \end{bmatrix} = \begin{bmatrix} b_p - D_{ps} D_{ss}^{-1} b_s \\ b_s \end{bmatrix} \quad \begin{matrix} B_{pp} \equiv A_{pp} - D_{ps} D_{ss}^{-1} A_{ps} \\ Z_{ps} \equiv A_{ps} - D_{ps} D_{ss}^{-1} A_{ss} \approx 0 \end{matrix}$$

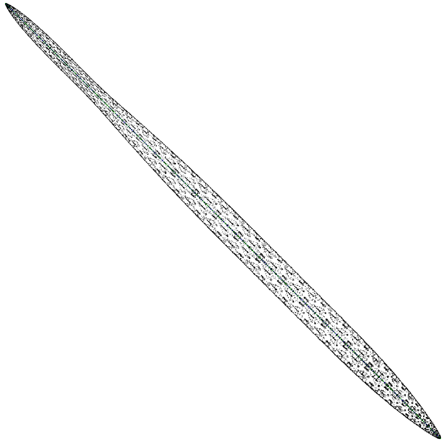
  - Use a two-stage method to solve the system.
  - M_1 - for pressure system, M_2 - for either complete system or saturations system **(block GS)**.
- SchurMFD – use mimetic finite difference method structure to compute the Schur.
- SchurSeries – use power series to invert Schur complement without assembly.

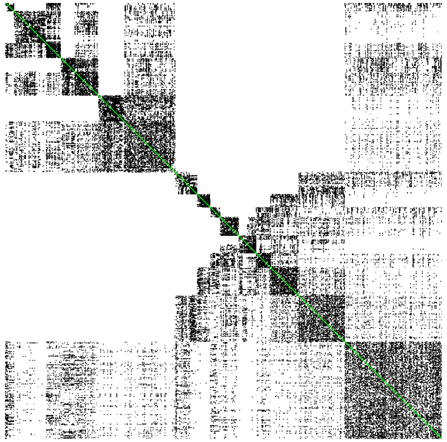
Предварительное упорядочивание

Initial System

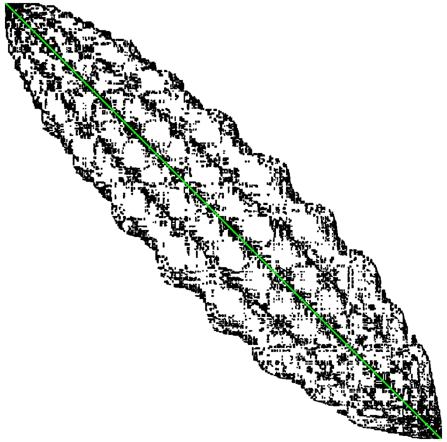


Bandwidth reduction





Fill-in reduction (METIS)



Bandwidth reduction

- Complete **blood coagulation** problem.
- 13 equations per cell.

- **Navier-Stokes** problem only.
- 4 equations per cell.
- Fill-in reduction (METIS) is 4 times slower than bandwidth reduction (RCM) algorithm.
- Quite dense matrix graph, big separator.

Контроль параметров

Flexible control of solver parameters

- Hierarchical embedding of each solver parameters
- Default parameter set, generation of files
- Common control parameters: verbosity, check, …

Input-output:

- MTX format
- Binary format for faster loading of large systems

```
Method:
    name = BICGSTAB
    dtol = 1e+50
    rtol = 1e-06
    tol = 1e-08
    maxiters = 5000
    true_residual = 0
    verbosity = 1
    Preconditioner:
        name = AMGRugeStuben
        check = 1
        cycle = V
        level = *
        operator_type = 2
        order = 0
        phi = 0.25
        refine_splitting = 0
        verbosity = 1
        write_matrix = 0
        CoarsestSolver:
            name = LU
            verbosity = 0
        /
        Smoother:
            name = Chebyshev
            maxiters = 2
            tol = 0
            verbosity = 1
        /
    /
/
```

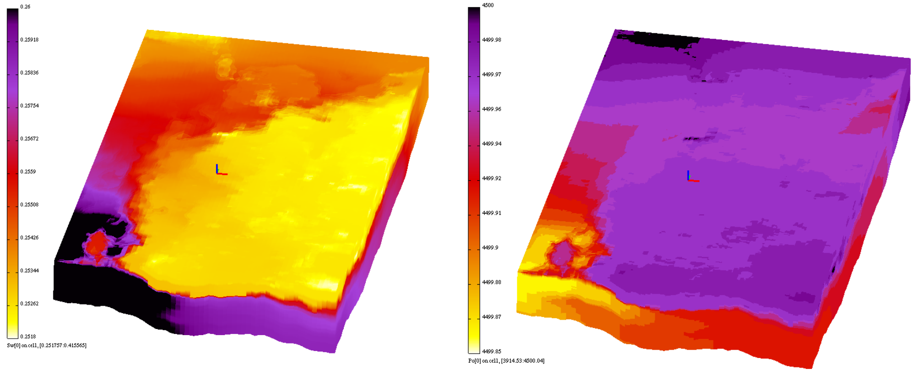
Пример применения (Black Oil)

Methods:

- MT-ILUC2 – maximum transversal with ILU
- CPR-MT-ILUC2 – CPR rescaling and then the preconditioner is applied to the full system
- CPR-TS – same as second, but multigrid is used to precondition pressure system
- CPR-TSGS – multigrid is used for pressure block and first preconditioner for saturations block

Results:

- Table does not account for setup-time memory
- Block Gauss-Seidel Requires the least memory, but looses to plain two-stage at extreme sizes



		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

Заключение и дальнейшие планы

- Implemented very flexible linear solvers framework
- Used as practical illustration in MIPT course
- Future directions:
  - **MPI** and **OpenMP target** parallelization
  - Integration of **domain-decomposition** methods
  - More **multistage** strategies:
    - Poromechanics – fixed-stress strategy
    - Navier-Stokes – pressure projection, Vanka smoother
    - Maxwell - …
  - integration into INMOST for linear system solution of multiphysics problems

Supported by RSF grant №21-71-20024

Спасибо за внимание

Contacts

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

Links

- WWW.INMOST.ORG
- WWW.INMOST.RU
- https://github.com/kirill-terekhov/mipt-solvers