Ani3D-extension of parallel platform INMOST and hydrodynamic applications

Vasily Kramarenko[1] Igor Konshin[1,3] Yuri Vassilevski[1,2]

[1]Institute of Numerical Mathematics, Russian Academy of Sciences, Moscow, Russia

[2]Moscow Institute of Physics and Technology, Dolgoprudny, Moscow Region, Russia

[3]Dorodnicyn Computing Centre, FRC CSC RAS, Moscow, Russia

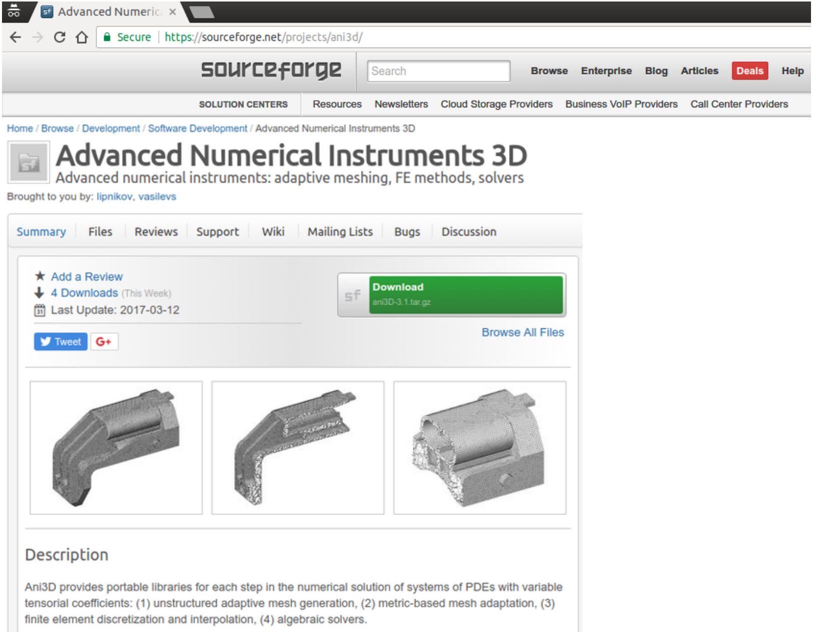
RuSCDays-2017, Moscow, 25 September, 2017

Contents

Motivation stages

- There are advanced serial FEM codes (e.g. Ani3D)
- Increasing the size of the problems to be solved
- Time and memory limitations for a serial computer
- Parallel mesh generation, discretization, linear system solution are required
- The direct code parallelization is too difficult
- There are software packages to operate with the distributed mesh data

Ani3D


Advanced Numeric
Secure
https://sourceforge.net/projects/ani3d/
sourceforge
Search
Browse
Enterprise
Blog
Articles
Deals
Help
SOLUTION CENTERS
Resources
Newsletters
Cloud Storage Providers
Business VoIP Providers
Call Center Providers
Home / Browse / Development / Software Development / Advanced Numerical Instruments 3D
Advanced Numerical Instruments 3D
Advanced numerical instruments: adaptive meshing, FE methods, solvers
Brought to you by: lipnikov, vasilevs
Summary
Files
Reviews
Support
Wiki
Mailing Lists
Bugs
Discussion
Add a Review
4 Downloads (This Week)
Last Update: 2017-03-12
Download
ani3D-3.1.tar.gz
Browse All Files
Tweet
G+
Description
Ani3D provides portable libraries for each step in the numerical solution of systems of PDEs with variable tensorial coefficients: (1) unstructured adaptive mesh generation, (2) metric-based mesh adaptation, (3) finite element discretization and interpolation, (4) algebraic solvers.

Ani3D software package

- Generation of tetrahedral meshes
- Mesh adaptation
- FEM discretization on tetrahedral meshes
- Solution of linear and nonlinear systems
- *Serial* code in Fortran and C

Open source code:

`http://sourceforge.net/projects/ani3d`

Ani3D libraries

Ani-MBA library

- generation of quasi-uniform meshes in a user-defined metric
- uniform refinement of tetrahedral meshes

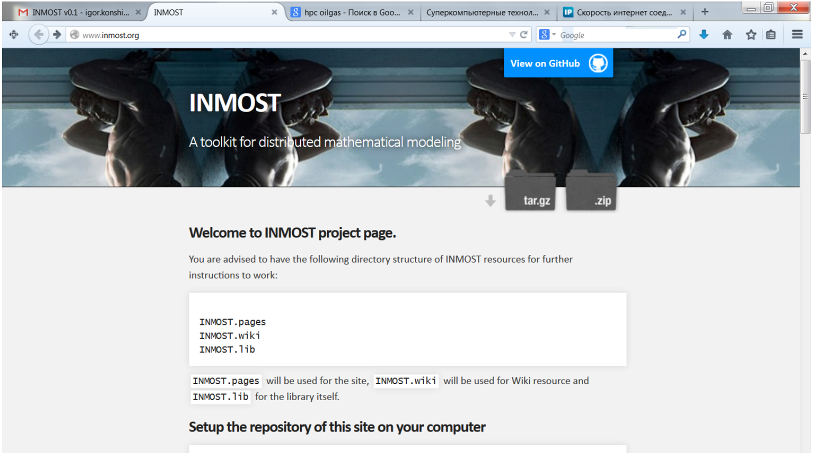
Ani-FEM library

- local finite element discretization on tetrahedron
- assembling the local discretizations into a global linear system

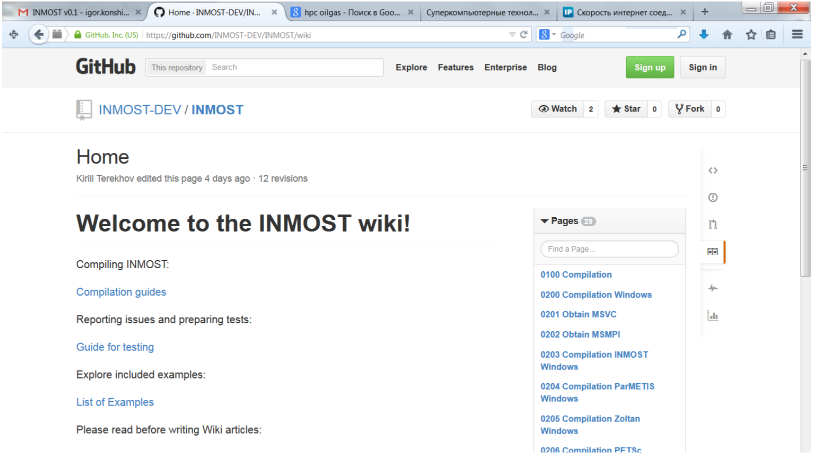
the rest of libraries...

- Ani-C2F, Ani-INB, Ani-LMR, Ani-PRJ, Ani-RCB
- Ani-ILU, Ani-LU

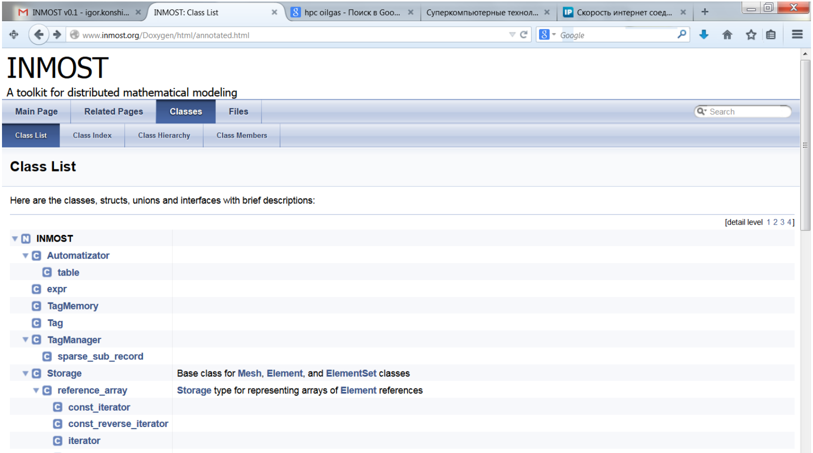
INMOST: www.inmost.org



INMOST: wiki.inmost.org



INMOST: doxygen.inmost.org



INMOST

- **I**ntegrated
- **N**umerical
- **M**odelling and
- **O**bject-oriented
- **S**upercomputing
- **T**echnologies

INMOST is the software platform for developing parallel numerical models on general meshes.

INMOST is a tool for supercomputer simulations characterized by a maximum generality of supported computational meshes, distributed data structure flexibility, cost-effectiveness, cross platform portability.

INMOST: Mesh

- Mesh data are distributed
- Conformal meshes with Tetrahedra, Hexahedra, Prisms, Pyramids, Polyhedra etc.
- Mesh elements hierarchy: Vertex, Edge, Face, Cell
- Mesh element can contain some data (tags)
- Specification of "ghost" elements ("hallo")
- Exchange tag data for ghost elements
- Save/Load mesh data in a parallel format file (.pmf, .pvtk, ...)

INMOST: Solver

- Assemble the distributed matrix and right-hand side of the linear system
- Parallel solution of the distributed linear system
- A set of internal linear solvers
- A set of external solvers: PETSc, Trilinos, SuperLU, ...

Parallelization Technology

Mesh

- read and partition
- refine on each processor *preserving conformity*
- merge

Init

- enumerate respective elements
- create tags for DOFs numbers
- synchronize tags

Assemble

- generate local matrix for each tetrahedron
- assemble them into a global matrix

Stokes problem

Formulation

$$
\begin{array}{ccccccc}
-\Delta \boldsymbol{u} & + & \nabla p & = & 0 & \text{in} & \Omega \\
\nabla \cdot \boldsymbol{u} & & & = & 0 & \text{in} & \Omega \\
 & & \boldsymbol{u} & = & \boldsymbol{u}_0 & \text{on} & \partial\Omega_1 \\
 & & \boldsymbol{u} & = & 0 & \text{on} & \partial\Omega_2 \\
\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{n}} & - & p & = & 0 & \text{on} & \partial\Omega_3
\end{array}
$$

$$\boldsymbol{u}_0 = (64 \cdot (y - 0.5) \cdot (1 - y) \cdot z \cdot (1 - z), 0, 0)$$

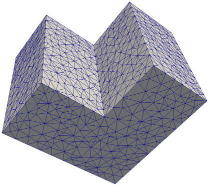


Figure: The coarsest mesh S0

Table: The problems parameters

Problem name	S0	S1	S2
Number of nodes	5187	36824	279903
Number of edges	31637	243079	1908542
Number of tetrahedra	25113	200904	1607232
Matrix size	115659	876533	6845238
Number of nonzeros	10751851	84374191	668849086

Results for S0 and S1 problem

Table: The solution of S0 problem on $p = 1, ..., 32$ processors

p	T_{ini}	T_{ass}	T_{prec}	T_{iter}	N_{iter}	Dens	PM	T_{sol}	S
1	0.06	8.77	3.30	4.26	102	0.81	0	7.56	1.00
2	0.04	5.72	2.27	3.27	132	0.96	0	5.54	1.36
4	0.03	3.95	1.57	2.34	152	1.18	0	3.91	1.93
8	0.02	2.25	1.19	1.81	172	1.53	5	3.00	2.52
16	0.02	1.63	1.20	1.40	182	2.01	3	2.83	2.67
32	0.02	1.49	1.50	1.33	182	2.80	10	2.83	2.67

Table: The solution of S1 problem on $p = 1, ..., 32$ processors

p	T_{ini}	T_{ass}	T_{prec}	T_{iter}	N_{iter}	Dens	PM	T_{sol}	S
1	0.38	69.07	31.41	97.20	242	0.85	0	128.61	1.00
2	0.28	40.97	20.30	76.88	322	0.93	0	97.18	1.32
4	0.21	25.31	12.76	47.88	332	1.03	0	60.64	2.12
8	0.14	13.76	7.83	28.68	332	1.15	1	36.51	3.52
16	0.10	8.37	4.77	18.46	362	1.37	3	23.23	5.53
32	0.06	5.09	4.12	12.70	402	1.65	8	16.82	7.64

Results for S2 problem

Table: The solution of S2 problem on $p = 4, ..., 32$ processors

p	T_{ini}	T_{ass}	T_{prec}	T_{iter}	N_{iter}	Dens	PM	T_{sol}	S
4	1.48	181.05	142.29	1484.02	722	0.97	0	1626.31	1.00
8	0.90	94.03	76.01	847.62	802	1.03	0	923.63	1.76
16	0.58	52.03	72.70	481.50	802	1.10	2	554.20	2.93
32	0.37	29.23	26.88	288.73	802	1.21	2	315.61	5.15

Unsteady convection-diffusion problem

Formulation

$$
\begin{aligned}
\frac{\partial c}{\partial t} - \nabla\left(D\nabla c\right) + \boldsymbol{v}\cdot\nabla c &= 0 \quad \text{in} \quad \Omega \\
c &= g \quad \text{on} \quad \partial\Omega
\end{aligned}
$$

$$
D = \begin{pmatrix} 0.001 & 0 & 0 \\ 0 & 0.001 & 0 \\ 0 & 0 & 0.001 \end{pmatrix}
$$

$$
\boldsymbol{v} = (1, 0, 0)
$$

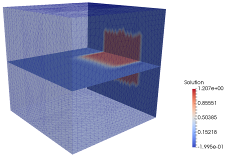


Figure: The concentration at time $t = 0.5$

Table: The problems parameters

Problem name	L0	L1	L2
Number of nodes	20417	155905	1218561
Number of tetrahedra	111616	892928	7143424
Matrix size	20417	155905	1218561
Number of nonzeros	291393	2281217	18053121

Results for L0 and L1 problem

Table: The solution of problem on mesh L0 on $p = 1, ..., 32$ processors

p	T_{ini}	T_{ass}	T_{sol}	T_{Σ}	S
1	2.87	62.68	3.27	68.82	1.00
2	1.93	40.45	2.05	44.43	1.54
4	1.29	26.94	1.24	29.47	2.33
8	0.82	16.76	0.76	18.34	3.75
16	0.58	11.91	0.53	13.02	5.28
32	0.45	8,89	0.55	9.89	6.95

Table: The solution of problem on mesh L1 on $p = 1, ..., 32$ processors

p	T_{ini}	T_{ass}	T_{sol}	T_{Σ}	S
1	26.16	552.87	33.86	612.89	1.00
2	16.48	327.93	21.31	365.72	1.67
4	9.83	197.89	12.19	219.91	2.78
8	5.75	114.81	7.28	127.84	4.79
16	3.56	73.57	4.03	81.16	7.55
32	2.31	47.28	2.45	52.04	11.77

Results for L2 problem

Table: The solution of problem on mesh L2 on $p = 1, ..., 32$ processors

p	T_{ini}	T_{ass}	T_{sol}	T_{Σ}	S
1	436.82	5692.31	723.23	6852.36	1.00
2	169.17	2628.09	258.71	3055.97	2.24
4	94.33	1461.24	147.30	1702.87	4.02
8	53.62	874.21	91.24	1019.07	6.72
16	31.52	522.99	53.28	607.79	11.27
32	17.6	308.84	29.61	356.05	19.24

Conclusions

- The Ani3D-extension of the parallel platform INMOST is presented.
- The extension widens the functionality of INMOST by the FE and meshing libraries of the Ani3D software package.
- Numerical experiments demonstrated the efficiency of the presented approach for the parallel solution of two model hydrodynamic problems.
- The examples can be downloaded at

https://github.com/INMOST-DEV/INMOST/tree/master/Examples/Ani_Inmost