

NUMERICAL MODELLING VIA INMOST SOFTWARE PLATFORM

I.N. KONSHIN^{1,2,3,4}, K.M. TEREKHOV^{1*}, AND YU.V. VASSILEVSKI^{1,2,3}

¹ Marchuk Institute of Numerical Mathematics, RAS. Moscow, Russia.

² Moscow Institute of Physics and Technology. Dolgoprudny, Moscow Region, Russia

³ Sechenov University. Moscow, Russia

⁴ Dorodnicyn Computing Centre, FRC CSC RAS. Moscow, Russia

*Corresponding author. E-mail: kirill.terehov@gmail.com

DOI: 10.20948/mathmontis-2020-47-7

Summary. INMOST is a software platform for the development of parallel numerical models on general polyhedral grids. In this paper we present the INMOST platform as the powerful tool for numerical modelling. The place of INMOST platform among other modern widespread libraries and numerical modelling packages is shown. A brief overview of tools that help in implementation of each stage of mathematical modelling is presented. Examples of INMOST application demonstrate appealing features of INMOST-based numerical modelling.

1 INTRODUCTION

The amount of software for unstructured mesh generation, numerical modelling and graphic visualization is huge. Along with the development of modern parallel computer systems, there is a need to use parallel algorithms with distributed mesh data. All these applications undoubtedly have a common set of needs for representing and manipulating distributed unstructured meshes. However, a large number of mesh representations are in use in the computational community each tailored to a specific application. Therefore, to gain widespread acceptance it is important to have a full mesh framework which allows applications to operate with all types of mesh data including the general polyhedral grids. At the same time, the infrastructure should be lightweight and efficient to have sufficient utilities for real-world numerical modelling applications. In addition, such an infrastructure should provide an opportunity for convenient assembling of systems of linear and nonlinear equations, their solving, as well as analysis and visualization of obtained solutions.

INMOST is a software platform for the development of parallel numerical models on general polyhedral grids [1, 2, 7, 9]. In this paper we present the INMOST platform as the powerful tool for numerical modelling. The examples of INMOST application demonstrate appealing features of INMOST-based numerical modelling.

In fact, the INMOST software platform does not include ready-made numerical models or even ready-made discretization schemes (such as finite volumes and finite elements). INMOST is just a software “platform” on the basis of which scientific researchers or developers of industrial codes can build their general-purpose grids distributed across processors, apply their discretization schemes and perform calculations for their numerical models. The presented examples of INMOST application demonstrate appealing features of INMOST-based numerical modelling. However, in view of the above reasoning, in this paper there is no direct comparison with the results of calculations using other packages.

2010 Mathematics Subject Classification: 86-08, 97N80, 65M50, 65N22, 65Y05.

Key words and Phrases: INMOST, Numerical Modelling, Computational Fluid Dynamics.

It is worth noting that at present there is a development (see, for example, [6]) presenting the extension of parallel platform INMOST using the Ani3D package [16], which allows the construction of adapted tetrahedral meshes along with the use of a large number of finite element discretizations (see INMOST Ani_Inmost examples [6]).

The present paper is organized as follows. Section 2 contains a brief description of INMOST platform functionality and shows the place of INMOST among other modern widespread libraries and numerical modelling packages. A detailed review of specific approaches to construction of numerical models is given in Section 3. In Section 4 we present several INMOST-based numerical models and their computational performance. The conclusion summarize the research.

2 INMOST SOFTWARE PLATFORM

The main purpose of INMOST platform is to provide to the user all the necessary tools for development and exploration of various numerical models. This involves wide functionality, i.e. operations for general distributed mesh data, convenient interface for assembling and solving systems of nonlinear and linear equations, with built-in or plugged-in linear algebra packages. INMOST software platform was developed to meet the following criteria:

- wide functionality;
- efficiency;
- reliability;
- universality;
- ease of use;
- portability;
- open source code.

At present, it is very difficult to find software packages that satisfy all of the above requirements. There are many alternative solutions, such as the FMDB library (Flexible distributed Mesh DataBase) [36], the MOAB library (Mesh-Oriented datABase) [37], the MStk library (MeSh ToolKit) [38], the STK library (Sierra ToolKit) [39], Salome package [40], OpenFOAM package (Open Source Field Operation And Manipulation CFD ToolBox) [41] and others that do not fully meet the stated criteria. Existing solutions do not always have easy portability between different platforms (Windows, Linux), existing implementations are not always reliable, it is impossible (in some packages) or difficult to implement user's discretization schemes. However most of publicly available packages are the best choice for the solution of a particular problem. For instance, packages ParMETIS [42] and Zoltan [43] distribute and redistribute a general mesh and graph data across processors, while libraries PETSc [21] and Trilinos [22] solve distributed systems of linear equations.

These considerations motivated a group from INM RAS to develop a comprehensive set of software tools that provides the necessary functional capabilities and allow the use of ParMETIS, Zoltan, PETSc, and Trilinos packages. These tools form the INMOST (Integrated Numerical Modelling and Object-oriented Supercomputing Technologies) software platform [9].

The main modules of the platform are:

- mesh operations module (INMOST Mesh);
- mesh data balancing module (INMOST Partitioner);

- automatic differentiation module (INMOST Automatizator);
- module for assembling and solving linear systems (INMOST Solver).

In the near future, visualization (INMOST DrawGrid) and parameter optimization tools (INMOST TTSP (Tool for Tuning Solver Parameters), INMOST OptimizerSolve example) will also be added as separate modules.

The main objective of INMOST is to support data structure and distributed mesh operations (INMOST Mesh) demanded by grid generators and numerical implementations of physical models. Consistent computational grids may contain cells with arbitrary numbers of faces, each face may be formed by an arbitrary number of edges. Thus, INMOST supports polyhedral cells of arbitrary configuration.

The grid balancing module (INMOST Partitioner) is responsible for automatic and efficient distribution of a computational grid among processors. External ParMETIS and Zoltan partitioners can be exploited as well as internal parallel partitioner based on K-means clustering. Upon grid distribution, the user can determine ghost cells along the interfaces of mesh subdomains where data of neighboring processors are synchronized. The number of layers of ghost cells is defined by the user who can also assign ghost cells explicitly. The last option is useful for complex discretization stencils. Importantly, the MPI exchange library is hidden from the user and compilation of the sequential single-processor version of INMOST is easy.

The automatic differentiation module (INMOST Automatizator) is developed to help in implementation of nonlinear numerical models, automatic generation and assembling of Jacobian matrices and nonlinear residuals appearing in Newton type linearizations. This module simplifies discretizations of new mathematical models.

The module (INMOST Solver) for assembling and solving systems of linear equations has the following appealing features:

- convenient and efficient assembling of a linear system matrix via addressing by the global row and column indices (INMOST MatSolve example);
- variety of built-in linear solvers based on threshold incomplete triangular factorizations as well as linear solvers from external packages, such as PETSc and Trilinos;
- the user can switch between linear solvers independently of the matrix assembling procedure.

3 NUMERICAL MODELLING STAGES AND ASSOCIATED TECHNOLOGIES

Numerical modelling is used both in academic and industrial purposes. The design of a numerical model consists of the following stages:

- Physical model;
- Mathematical model;
- Discretization;
- Solution methods;
- Computer program;
- Post-processing and analysis of the results.

Physical model. Understanding of underlying physical phenomena is crucial: advection, diffusion, reaction or decay processes; heat or density transfer; elasticity or rheology effects,

etc. Chosen physics implies a list of primary physical quantities to be measured and/or simulated: pressure, velocity, concentration, temperature, probabilities, etc.

Mathematical model is represented by a system of differential or integral equations describing the chosen phenomena and involving the chosen physical quantities. Often, formulation of boundary and initial conditions is required. Apart of problem formulation, analysis of solution existence and uniqueness is important. In some cases the analysis is not available though this does not inhibit the numerical solution.

Mathematical description involves governing equations and domain geometry. For academic models in parallelepipedal domains it is sufficient to specify the length-width-height parameters only. In more realistic cases the domain geometry can be presented as a set of unions or intersections of primitives: spheres, cylinders, cones, half spaces, parallelepipeds, etc. The most complex geometries for industrial applications can be constructed with one of the following tools: AutoCAD [10], 3DS Max [11], Sketchup [12], OpenCascade [13], etc.

Discretization of a mathematical model includes discretization of the domain and the equations. Discretization of the domain implies generation of a computational mesh. *A priori* or runtime local mesh refinement near solution gradients crucially affects the solution accuracy. Accurate approximation of domain boundaries is also important. A lot of mesh generation and refinement tools (commercial or open source) are available: Tetgen [14], GMSH [15], ANI3D [16], INMOST OctreeCutcell [9], etc.

The type of the computational grid correlates with the method for discretization of differential equations: finite differences (FD) imply structured grids, finite elements (FE) imply tetrahedral or hexahedral meshes, while finite volumes (FV) fit to general polyhedral meshes.

Discretization results in a system of linear or nonlinear algebraic equations. The total number of equations is proportional to the number of mesh elements (cells, faces, edges, nodes) and ranges from thousands to billions.

Solution methods. For the solution of systems of nonlinear algebraic equations, the following approaches can be used: Newton method, Picard method, line search method, trust interval method and others. Examples of nonlinear solver packages are: SUNDIALS [17], Trilinos NOX [18]. Automatic differentiation capabilities of INMOST allow for easy assembly and solution of the nonlinear system with the Newton method.

The solution of a nonlinear system is based on its linearization which results in a large system of sparse linear equations. For solution of the system the following methods are applicable: direct factorization, Krylov's iterative methods with different preconditioners such as incomplete factorization, algebraic multigrid methods, domain decomposition methods, etc. A variety of advanced software packages can be used for this purpose: SuperLU [19], MUMPS [20], PETSc [21], Trilinos [22], Hypre [23], ILUPACK [24], INMOST Solve [9], etc.

Computer program. To design the computer program and connect the required software packages, one exploits computer languages C, C++, Fortran, Python, etc. To utilize parallelism of modern computers including various accelerators, one may use computer libraries and languages: OpenMP [25], MPI [26], OpenCL [27], CUDA [28], etc. In some cases the number of computer languages within a numerical model may reach 4 or 5.

For computer program debugging the following tools are available: gdb [29], valgrind [30], drmemory [31], etc.

Post-processing and analysis of results. Computation of required metrics, plotting of

graphs and diagrams, generation of pictures and videos facilitate analysis of the computed results. Useful visualization tools are ParaView [32], VisIt [33], General Mesh Viewer [34], gnuplot [35], INMOST DrawGrid [9], etc.

4 INMOST-BASED NUMERICAL MODELS

In this section we review several applications of INMOST platform in numerical modelling.

4.1 Incompressible fluid flow

We first consider the incompressible fluid flow. The problem is described by the Navier–Stokes equations:

$$\begin{aligned} \rho \frac{\partial \mathbf{u}}{\partial t} + \operatorname{div}(\rho \mathbf{u} \mathbf{u}^T - \mu \nabla \mathbf{u} + p \mathbf{I}) &= \mathbf{0}, \\ \operatorname{div}(\mathbf{u}) &= 0. \end{aligned} \quad (1)$$

Here $\mathbf{u} = \{u, v, w\}^T$ is the unknown fluid flow velocity, p is the unknown pressure, ρ is the constant density and μ is the dynamic viscosity. The system of Navier–Stokes equations is augmented by the boundary conditions. On the boundary we can impose no-slip or slip condition, Maxwell–Navier friction, do-nothing condition, condition for the normal stress (fixed pressure).

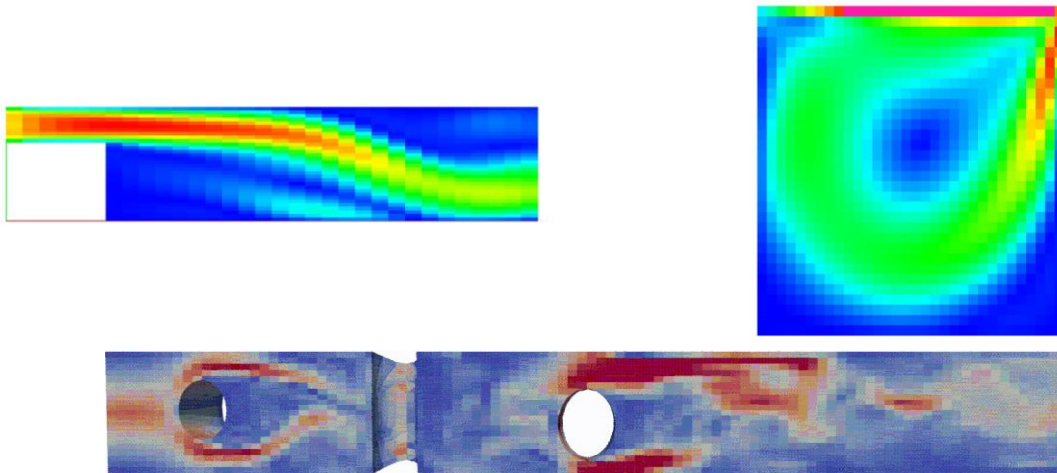


Figure 1. FV solutions: flow over the backward step (top-left), flow in the lid-driven cavity (top-right), flow in a channel with obstacles (bottom).

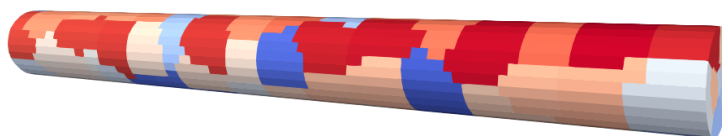


Figure 2. Decomposition of a computational domain for 92 processors

Both the momentum and the continuity equations are discretized by the finite volume method. The unknowns are collocated at the cell centers. To avoid checkerboard patterns in the discrete pressure due to the Ladyzhenskaya–Babushka–Brezzi instability, we stabilize the discretization by eigen-splitting of the part of the flux that corresponds to the pressure and the incompressibility condition, by analogy with the approach [44]. The FV system is nonlinear due to the convective term and the numerical scheme is stable even for large time steps.

In Figure 1 (top row of pictures) visualization of the flow for the solutions to two benchmark problems are shown: the steady flow over the backward step and the steady flow in the lid-driven cavity. In the bottom picture of Figure 1 we present the unsteady flow in a channel with cylindrical obstacles in the case of high Reynolds number ($Re \sim 1000$). Figure 1 demonstrates physically adequate absolute velocity solution for a relatively small number of cells using finite volume discretization scheme. The presented examples make the full use of the INMOST programming platform, i.e. they exploit the general mesh data structure, the sparse matrix structure and automatic differentiation for the Jacobian system assembly and built-in sparse parallel linear solvers. Thanks to INMOST the flow model may be run in parallel. Decomposition of a computational mesh in a channel for 92 processors is demonstrated in Figure 2. The solution times of steady Pousielle flow problem for 36 and 92 MPI processes were 2.51 and 1.25 seconds, respectively, which implies feasible parallel efficiency.

The presented incompressible fluid flow model was complemented with the blood coagulation model [7] using INMOST multiphysics extension. This extension is currently under active development. It allows to couple the flow model with a reaction-advection-diffusion model triggering coagulation into a joint nonlinear system which is solved on each time step.

4.2 Free surface fluid flow

We solve numerically the problem of fluid flow with a free surface on dynamically adapted octree grids. The problem is guided by the coupled solution of the system of incompressible Navier–Stokes equations and the level set equation:

$$\begin{aligned} \rho \frac{\partial \mathbf{u}}{\partial t} + \operatorname{div}(\rho \mathbf{u} \mathbf{u}^T - \mu \nabla \mathbf{u} + p \mathbf{I}) &= \rho g \nabla z, \\ \operatorname{div}(\mathbf{u}) &= 0, \\ \frac{\partial \varphi}{\partial t} + \operatorname{div}(\varphi \mathbf{u}) &= 0, \\ |\nabla \varphi| &= 1, \end{aligned} \tag{2}$$

where $\mathbf{u} = \{u, v, w\}^T$ and p are unknown velocity and pressure, ρ is the density, μ is the dynamic viscosity, g is the gravitational constant. In this problem the velocity is staggered on cell faces and the pressure is collocated at cell centers. The Navier–Stokes equations are augmented with the boundary conditions listed in Section 4.1. The unknown level-set function φ , satisfying the Neumann boundary condition, is passively advected with fluid and reinitialized by the solution of the Eikonal equation. The level-set function is prescribed at mesh nodes, its sign separates the domain into two parts: the fluid domain and the empty domain, see Figure 3 (left). The boundary condition at the interface accounts the surface

tension:

$$-\mu \frac{\nabla \mathbf{u} + (\nabla \mathbf{u})^T}{2} \mathbf{n} + p \mathbf{n} = \sigma \kappa \mathbf{n}. \quad (3)$$

Here σ is the surface tension coefficient, κ is the surface curvature computed from the level-set function, \mathbf{n} is the surface normal. More implementation details can be found in [45] and references therein. Figure 3 (right) shows the surface tension induced water crown as a splash from falling drops into a pool filled with the fluid.

Currently, the numerical model relies on the sparse matrix structure, automatic differentiation and linear solvers from INMOST tackling the diffusion problem and pressure projection problem. However, the octree mesh data structure of the model is not based on INMOST tools limiting parallelization of the model to OpenMP technology. The parallel general mesh adaptivity functionality is already developed [8, 46], and the transition of the code to the INMOST mesh data structure is underway.

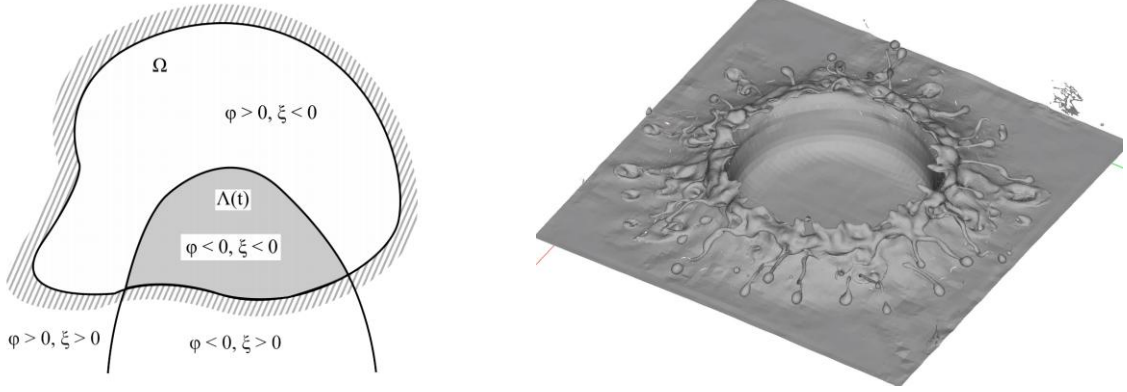


Figure 3. Computational domain (left) and the computed water crown (right)

4.3 Oil and gas modelling

The flow of the mixture of fluids subject to the Darcy law is used for simulation of primary or secondary oil and gas recovery from a heterogeneous anisotropic fractured reservoir. The black oil model equations for unknown pressure and saturations are [50]:

$$\begin{aligned} \frac{\partial \rho_w \theta S_w}{\partial t} - \operatorname{div}(\lambda_w \mathbf{K}(\nabla p - \rho_w g \nabla z)) &= q_w, \\ \frac{\partial \rho_o \theta S_o}{\partial t} - \operatorname{div}(\lambda_o \mathbf{K}(\nabla p - \nabla P c_o - \rho_o g \nabla z)) &= q_o, \\ \frac{\partial \rho_g \theta (R S_o + S_g)}{\partial t} - \operatorname{div}(\lambda_g \mathbf{K}(\nabla p - \nabla P c_g - \rho_g g \nabla z)) \\ - \operatorname{div}(\lambda_{go} \mathbf{K}(\nabla p - \nabla P c_o - \rho_o g \nabla z)) &= q_g. \end{aligned} \quad (4)$$

Here \mathbf{K} is the permeability of the rock, R is the gas solubility, θ is the porosity, ρ is the density dependent on pressure, $\lambda = \rho k_r / \mu$ are phase mobilities dependent on pressure and saturations for water, oil and gas, q are sources and sinks representing the wells guided by the

Peaceman formula [49]. The no-flow boundary condition is imposed on all the boundaries. Various FV discretization methods for the Darcy problem may be used for the numerical solution of system (4) [51]. In Figure 4 (left) we present an example of water saturation field in a reservoir with complex geology defined through a general mesh adapted to geological structures. In Figure 4 (right) we demonstrate water saturation in a network of fractures.

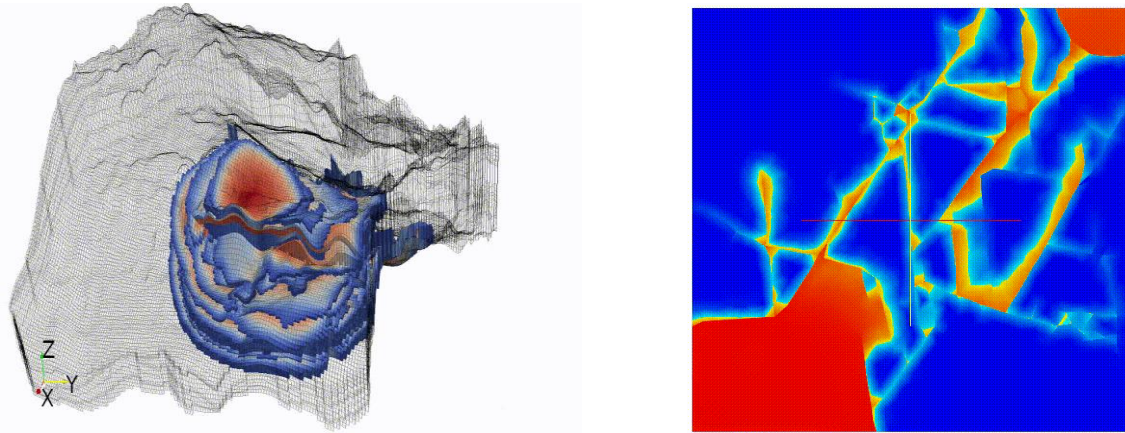


Figure 4. Water saturation field and general mesh in a reservoir (left); water saturation in a fracture network (right)

The numerical model uses the mesh data structure, sparse matrix structure, automatic differentiation and linear solvers from INMOST.

4.4 Mechanics of deformable bodies

The model describes the elastic deformation of bodies from heterogeneous anisotropic material. The model is given by the elasticity equations $\sigma = \mathbf{C}:(\nabla \mathbf{u} + (\nabla \mathbf{u})^T)/2$, $\text{div}(\sigma) = 0$. Here $\mathbf{u} = \{u, v, w\}^T$ is the displacement field, σ is the 3x3 stress tensor, \mathbf{C} is the 4-th rank material stiffness tensor, “:” is the contraction operator. The system is augmented by Diriclet, Neumann or roller type boundary conditions.

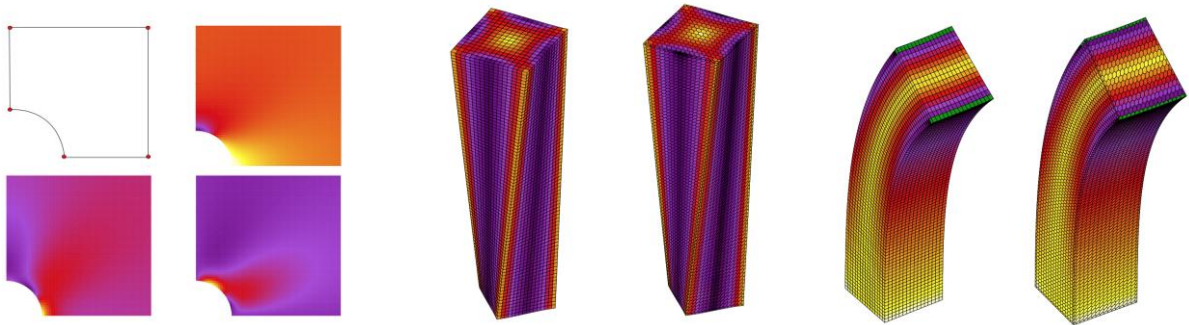


Figure 5. Stress components under load (left), stress magnitude under twisting (middle), and bending (right)

The numerical implementation is based on the FV method described in [47]. Figure 5 demonstrates the application of the model to benchmark problems: three components of stress in perforated infinite strip under load (left), magnitude of stress in a beam under twisting (middle) and bending (right) on structured and unstructured grids.

The numerical model uses the mesh data structure, sparse matrix structure, automatic differentiation and linear solvers from INMOST.

4.5 Poromechanics

The last example describes a solid body saturated with the fluid. The model is used for analysis of land subsidence and Earth fissuring due to water pumping into or out of the ground, as well as bed failure under constructions. It couples the solid mechanics problem (Section 4.4) with the fluid filtration problem guided by the Darcy problem (Section 4.3), the interaction between solid and fluid being described by Biot coupling terms [52]. For the single phase flow with unknowns u , v , w , and p it reads as:

$$\begin{aligned} \frac{1}{M} \frac{\partial p}{\partial t} - \operatorname{div} \left(\mathbf{K} (\nabla p - \rho g \nabla z) - \mathbf{B} \frac{\partial \mathbf{u}}{\partial t} \right) &= q, \\ -\operatorname{div} \left(\mathbf{S} : \frac{\nabla \mathbf{u} + (\nabla \mathbf{u})^T}{2} - \mathbf{B} p \right) &= \rho g \nabla z. \end{aligned} \quad (5)$$

Here p is the fluid pressure, $\mathbf{u} = \{u, v, w\}^T$ is the displacement of solid, \mathbf{S} is the material compliance tensor, inverse of the stiffness tensor, ρ is the density, \mathbf{B} is the Biot coefficient tensor, M is the Biot modulus, $1/M$ is the specific storage coefficient. The boundary conditions are similar to those discussed in Sections 4.3 and 4.4.

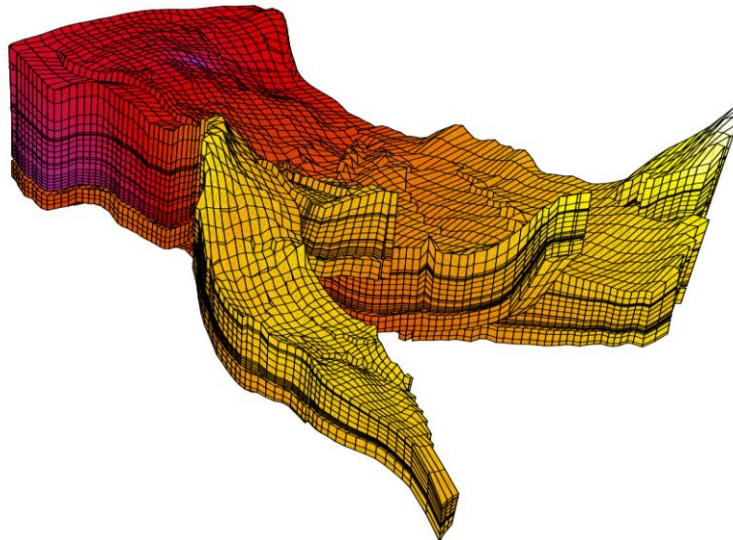


Figure 6. Computational mesh and the pressure field for poromechanics problem

Figure 6 demonstrates the pressure field for the poromechanics problem due to oil recovery from the Norne oil field, the elastic properties are synthetic. The grid is unstructured with faults, the system (5) is discretized by the FV method [48].

The numerical model uses the mesh data structure, sparse matrix structure, automatic differentiation and linear solvers from INMOST.

CONCLUSION

The present paper presents the INMOST platform as a powerful tool for numerical modelling. INMOST provide a wide functionality of operations for general distributed mesh data, convenient interface for assembling and solving systems of linear and nonlinear equations, as well as analysis and visualization of obtained solutions. The description of the most important modules of INMOST is given. An overview of numerical modelling stages along with the tools that help in their implementation is presented. The place of INMOST platform among other modern widespread libraries and numerical modelling packages is shown.

For a specific application in numerical modeling, it may turn out to be the most optimal choice of a special grid generator, as well as the sampling method for this type of grid. This is the main reason that these modules are not included in the INMOST software platform. For the same reason, it is difficult to make a full comparison of INMOST with other popular numerical modelling packages, which mainly use the simplest semi-regular types of grids, allowing the direct use of certain types of discretization, for example, two-point finite-volume ones. Conversely, INMOST software platform focuses on support for operation with general type grids. Despite this, the presented examples of INMOST application demonstrate appealing features of INMOST-based numerical modelling. Especially, this includes a wide range of its application for solving problems from gas hydrodynamics and problems with a free surface to problems of mechanics of a solid and deformable bodies, as well as problems of subsurface flow, including problems of poromechanics.

Acknowledgements: The work was supported in part by the world-class research center “Moscow Center for Fundamental and Applied Mathematics” and by the RAS Research program No. 26 “Basics of algorithms and software for high performance computing”.

REFERENCES

- [1] Yu. Vassilevski, I. Konshin, G. Kopytov, K. Terekhov, *INMOST – a software platform and graphical environment for development of parallel numerical models on general meshes*. Lomonosov Moscow State Univ. Publ., Moscow, (2013), 144 p. (in Russian)
- [2] A.A. Danilov, K.M. Terekhov, I.N. Konshin, Yu.V. Vassilevski, “INMOST parallel platform: framework for numerical modeling”, *Supercomp. Frontiers and Innovations*, **2**(4), 55–66 (2015).
- [3] I. Konshin, I. Kapyrin, K. Nikitin, K. Terekhov, “Application of the parallel INMOST platform to subsurface flow and transport modelling”, *Lecture Notes in Computer Science*, **9574**, 277–286 (2016).
- [4] F.V. Grigoriev, I.V. Kapyrin, I.N. Konshin, “Software platform INMOST in the GeRa code to operate with the distributed mesh data”. *Mathematica Montisnigri*, **XXXVI**, 27–44 (2016) (in Russian)
- [5] I. Konshin, I. Kapyrin, “Scalable computations of GeRa code on the base of software platform INMOST”, *Lecture Notes in Computer Science*, **10421**, 433–445 (2017).
- [6] V. Kramarenko, I. Konshin, Y. Vassilevski, “Ani3D-extension of parallel platform INMOST and hydrodynamic applications”, *Communications in Computer and Information Science*, **793**, 219–228 (2017).
- [7] K. Terekhov, Yu. Vassilevski, INMOST parallel platform for mathematical modeling and applications. *Communications in Computer and Information Science*, **965**, 230–241 (2018).
- [8] K. Terekhov, Yu. Vassilevski, Mesh modification and adaptation within INMOST programming

- platform. *Lecture Notes in Computer Science and Engineering*, **131**, 243–255 (2019).
- [9] INMOST – a toolkit for distributed mathematical modelling. <http://www.inmost.org> (Accessed November 29, 2019).
 - [10] AutoCAD – a computer-aided design (CAD) software. <https://www.autodesk.com/products/autocad/overview> (Accessed November 29, 2019).
 - [11] 3DS Max – 3D modeling and rendering software for design visualization. <http://www.autodesk.com/products/3ds-max/overview> (Accessed November 29, 2019).
 - [12] Sketchup – 3D modeling for professionals. <https://www.sketchup.com/> (Accessed November 29, 2019).
 - [13] OpenCascade – Advanced 3D Technology. <https://www.opencascade.com/> (Accessed November 29, 2019).
 - [14] Tetgen – a quality tetrahedral mesh generator and a 3D Delaunay triangulator. <http://wias-berlin.de/software/index.jsp?id=TetGen> (Accessed November 29, 2019).
 - [15] GMSH – a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. <http://gmsh.info/> (Accessed November 29, 2019).
 - [16] ANI3D – Advanced Numerical Instruments 3D. <https://sourceforge.net/projects/ani3d/> (Accessed November 29, 2019).
 - [17] SUNDIALS: SUite of Nonlinear and Differential/ALgebraic equation Solvers. URL: <https://computing.llnl.gov/projects/sundials> (Accessed November 29, 2019).
 - [18] Trilinos NOX – nonlinear analysis tools. https://trilinos.github.io/nox_and_loca.html (Accessed November 29, 2019).
 - [19] SuperLU – Supernodal sparse direct solver. <https://portal.nersc.gov/project/sparse/superlu/> (Accessed November 29, 2019).
 - [20] MUMPS: MUltifrontal Massively Parallel sparse direct Solver. <http://mumps.enseeiht.fr/> (Accessed November 29, 2019).
 - [21] PETSc – Portable Extensible Toolkit for Scientific Computation. <https://www.mcs.anl.gov/petsc/> (Accessed November 29, 2019).
 - [22] Trilinos linear solver. https://trilinos.github.io/linear_solver.html (Accessed November 29, 2019).
 - [23] HyPre – scalable linear solvers and multigrid methods. <https://computing.llnl.gov/projects/hypre-scalable-linear-solvers-multigrid-methods> (Accessed November 29, 2019).
 - [24] ILUPACK – multilevel ILU preconditioners for general real and complex matrices. <http://ilupack.tu-bs.de/> (Accessed November 29, 2019).
 - [25] OpenMP – multi-platform shared-memory parallel programming in C/C++ and Fortran. <https://www.openmp.org/> (Accessed November 29, 2019).
 - [26] MPI – Message Passing Interface. <https://www.mpi-forum.org/> (Accessed November 29, 2019).
 - [27] OpenCL (Open Computing Language) – The open standard for parallel programming of heterogeneous systems. <https://www.khronos.org/opencl/> (Accessed November 29, 2019).
 - [28] CUDA (Compute Unified Device Architecture) – a parallel computing platform and programming model for general computing on GPUs. <https://developer.nvidia.com/cuda-zone> (Accessed November 29, 2019).
 - [29] gdb – The GNU project debugger. <https://www.gnu.org/software/gdb/> (Accessed November 29, 2019).
 - [30] valgrind – instrumentation framework for building dynamic analysis tools. <http://valgrind.org/> (Accessed November 29, 2019).
 - [31] drmemory – Memory Debugger for Windows, Linux, and Mac. <https://drmemory.org/> (Accessed November 29, 2019).
 - [32] ParaView – an open-source, multi-platform data analysis and visualization application. <https://www.paraview.org/> (Accessed November 29, 2019).
 - [33] VisIt – an open-source interactive parallel visualization and graphical analysis tool for viewing scientific data. <https://visit.llnl.gov/> (Accessed November 29, 2019).

- [34] GMV – The General Mesh Viewer. <http://www.gmv-barracuda.com/index.html> (Accessed November 29, 2019).
- [35] gnuplot – a portable command-line driven graphing utility. <http://www.gnuplot.info/> (Accessed November 29, 2019).
- [36] E. S. Seol, FMDB: Flexible Distributed Mesh DataBase for parallel automated adaptive analysis. *Ph.D. Thesis*, Rensselaer Polytechnic Institute, New York, 2005, 151 p. Available at URL: <https://www.scorec.rpi.edu/REPORTS/2005-9.pdf> (Accessed November 29, 2019).
- [37] MOAB – a Mesh-Oriented datABase. URL: <https://press3.mcs.anl.gov/sigma/moab-library/> (Accessed November 29, 2019).
- [38] MSTK (MeSh ToolKit) – Infrastructure library for reading, writing, querying and manipulating distributed, general unstructured polyhedral meshes. <https://github.com/MeshToolkit/MSTK> (Accessed November 29, 2019).
- [39] STK (Sierra ToolKit mesh) – an unstructured mesh database. <https://trilinos.github.io/stk.html> (Accessed November 29, 2019).
- [40] Salome – an open-source software that provides a generic pre- and post-processing platform for numerical simulation. URL: <https://www.salome-platform.org/> (Accessed November 29, 2019).
- [41] OpenFOAM – Open Source Field Operation And Manipulation CFD ToolBox. URL: <https://www.openfoam.com/> (Accessed November 29, 2019).
- [42] ParMETIS: parallel graph partitioning and fill-reducing matrix ordering. <http://glaros.dtc.umn.edu/gkhome/metis/parmetis/overview> (Accessed November 29, 2019).
- [43] Zoltan – a toolkit of parallel services for dynamic, unstructured, and/or adaptive simulations. <https://trilinos.github.io/zoltan.html> (Accessed November 29, 2019).
- [44] K.M. Terekhov, Yu.V. Vassilevski, “Finite volume method for coupled subsurface flow problems, I: Darcy problem”, *J. of Computational Physics*, **395**, 298–306 (2019).
- [45] K.D. Nikitin, K.M. Terekhov, Yu.V. Vassilevski, “Two methods of surface tension treatment in free surface flow simulations”, *Applied Mathematics Letters*, **86**, 236–242 (2018).
- [46] K.M. Terekhov, “Parallel dynamic mesh adaptation within INMOST platform”, *Communications in Computer and Information Science*, **1129**, 313–326 (2019).
- [47] K.M. Terekhov, H.A. Tchelepi, “Cell-centered finite-volume method for elastic deformation of heterogeneous media with full-tensor properties”, *J. of Computational and Applied Mathematics*, **364**, 112331, (2020).
- [48] K.M. Terekhov, “Cell-centered finite-volume method for heterogeneous anisotropic poromechanics problem”, *J. of Computational and Applied Mathematics*, **365**, 112357 (2020).
- [49] D.W. Peaceman, “Representation of a horizontal well in numerical reservoir simulation”, *SPE Advanced Technology Series*, **1**(1), 7–16 (1993).
- [50] Zh. Chen, “Reservoir simulation: mathematical techniques in oil recovery”, *SIAM Series in Applied Mathematics*, Vol. 77, 2007.
- [51] K.M. Terekhov, B.T. Mallison, H.A. Tchelepi, “Cell-centered nonlinear finite-volume methods for the heterogeneous anisotropic diffusion problem”, *J. Comput. Physics*, **330**, 245–267 (2017).
- [52] M.A. Biot, “General theory of three-dimensional consolidation”, *J. of Applied Physics*, **12**(2), 155–164 (1941).

Received November 20, 2019