Implementation and Testing of the Fast Numerical Algorithm for Simulation of 3D Gravity Creeping Flow of Incompressible Newtonian Fluid

Timofey V. Abramov
Trofimuk Institute of Petroleum Geology and Geophysics of Siberian Branch of Russian Academy of Sciences
Koptug ave. 3, 630090
Novosibirsk, Russia
AbramovTV@ipgg.sbras.ru

Mikhail M. Lavrentiev
Novosibirsk State University
Pirogova Str. 2, 630090
Novosibirsk, Russia
mmlavr@nsu.ru

Boris V. Lunev
Trofimuk Institute of Petroleum Geology and Geophysics of Siberian Branch of Russian Academy of Sciences
Koptug ave. 3, 630090
Novosibirsk, Russia
LunevBV@ipgg.sbras.ru

ABSTRACT

A method for fast numerical simulation of 3D gravitational creeping flow of uniform viscous incompressible Newtonian fluid with piecewise constant density in a half-space bounded above by a free surface is presented. The method is based on the known explicit form of Green's function of corresponding boundary value problem.

As it is known, the solution to this problem may be found just as a convolution of two known functions — Green's one and right hand side of the equation, instead of solving a difference equation. Direct algorithm requires $O(N^2)$ operations to calculate velocity filed in the entire half-space, where N is the number of nodes of the computational grid. Obviously, this procedure is extremely inefficient when N is large enough (about 10^6 and more), and parallel computing does not resolve the problem.

In this paper the modification of the calculation algorithm is proposed, implemented and compared with the direct one. The modification consists in the conversion of the desired convolution-like sum (solution to the problem) to the form of cyclic convolution, which can be calculated in $O(N \log N)$ operations by a fast algorithm.

Categories and Subject Descriptors

F.2.1 [**Analysis of algorithms and problem complexity**]: Numerical Algorithms and Problems—*Computation of transforms*; G.1.0 [**Mathematics of Computing**]: Numerical analysis—*Numerical algorithms, Parallel algorithms*; G.1.8 [**Mathematics of Computing**]: Partial Differential Equations—*Elliptic equations*

Keywords

Rayleigh-Taylor instability, Newtonian fluid, boundary value problem, Green's function, cyclic convolution, fast Fourier transform

ICAIT '16, Oct. 6 – 8, 2016, Aizu-Wakamatsu, Japan.

1. INTRODUCTION

Gravitational creeping flow of inhomogeneous (primarily, with different densities) incompressible Newtonian fluid is widely used as a mathematical model for many geological processes in the Earth. In such way evolution of sedimentary basins, thermal convection in the mantle, continents' influence on the heat flows in the mantle and many other processes are simulated. Perhaps, the most practical importance has salt tectogenesis (salt diapirism) — solid rocks upwelling through more dense sediments because of low density of salt, which creates high-amplitude mushroom-shaped structures (salt diapirs or domes, see Fig. 1).

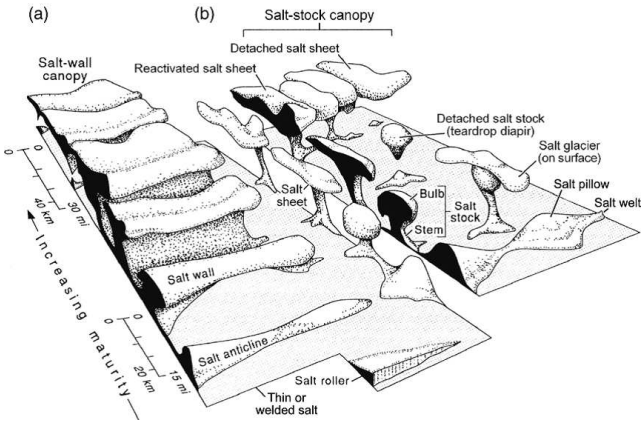

Figure 1: Diagram observed various salt structures. Structural maturity and size increase toward the background. (a) Structures rising from line sources. (b) Structures rising from point sources [3].

The practical interest of geologists and geophysicists is in the fact that salt tectogenesis appears in almost all major oil and gas provinces, it complicates the structure of the sedimentary basin, so it determines the distribution of hydrocarbon deposits [9].

According to modern understanding, physically it is a spe-

cial case of Rayleigh-Taylor instability and it is adequately described by gravitational creeping flow of incompressible Newtonian fluid[1].

To study a physical process via numerical simulation it is necessary to make a great number of computational experiments, varying the model parameters. Therefore, fast simulation tools are on demand. Fortunately, there is an analytical solution of the corresponding boundary value problem for gravitational creeping flow of Newtonian fluid with inhomogeneous density and continuous viscosity, which is bounded by a free surface [5, 6]. The corresponding formulae make it possible to create several efficient algorithms for calculation of such flows.

In this paper we describe 3D fast numerical algorithm for solving the considered problem. Precise mathematical statement of the problem, as well as the corresponding discrete version, are given in Section 2. Section 3 describes the proposed fast numerical algorithm, while the results of numerical tests are presented in Section 4.

2. MATHEMATICAL STATEMENT

Consider a half-space bounded above by a free surface, occupied by several incompressible and immiscible Newtonian fluids with very high uniform viscosity ($\sim 10^{20}$ Pa $\cdot$ s) and different densities, as shown in Fig. 2. The problem is to determine the gravity creeping flow of these fluids with $\mathbf{g} = (-g, 0, 0)$.

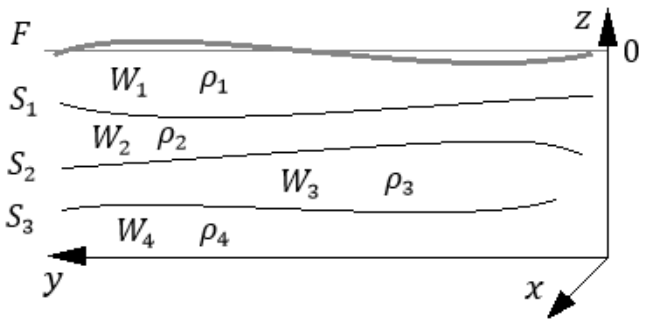

Figure 2: A half-space in Cartesian coordinate system $\mathbf{x} = (x, y, z)$ occupied by liquids $W_1, \ldots, W_4$ with densities $\rho_1, \ldots, \rho_4$, respectively, and uniform viscosity μ. They are separated by the surfaces $S_1, \ldots, S_3$, and bounded above by the free surface F.

Each time moment t the density distribution in the half-space is uniquely determined by the actual configuration of surfaces S_i: $\rho(\mathbf{x}, t) = \rho_i$ for $\mathbf{x} \in W_i$. Density ρ, stress tensor $\mathbf{T}$ and pressure P are represented as:

$$\begin{aligned} \rho(\mathbf{x}, t) &= \rho^0(z, t) + \sigma(\mathbf{x}, t) \\ \mathbf{T}(\mathbf{x}, t) &= \mathbf{T}^0(z, t) + \boldsymbol{\tau}(\mathbf{x}, t) \\ P(\mathbf{x}, t) &= P^0(z, t) + p(\mathbf{x}, t), \end{aligned} \tag{1}$$

where $\rho^0(z, t)$, $\mathbf{T}^0(z, t)$, $P^0(z, t)$ characterize hydrostatic condition ($\mathbf{T}^0(z, t) = -\delta_{ij} P^0(z, t) = -\delta_{ij} g \int_0^z \rho^0(z, t) dz$, where δ_{ij} — Kronecker symbols), and $\sigma(\mathbf{x}, t)$, $\boldsymbol{\tau}(\mathbf{x}, t)$ $p(\mathbf{x}, t)$ — small deviations.

As it is shown in [5, 6], the problem of determining a creeping flow may be formulated with respect to the deviations and splitting into quasi-stationary part and evolution part. The quasi-stationary part with linearized boundary conditions has the form:

$$\begin{aligned} &\mu \boldsymbol{\nabla}^2 \mathbf{v} - \boldsymbol{\nabla} p = -\sigma \mathbf{g}, \\ &\boldsymbol{\nabla} \cdot \mathbf{v} = 0, \\ &(v_z = \tau_{zx} = \tau_{zy} = 0) \mid_{z=0}, \end{aligned} \tag{2}$$

with the additional condition for determining $\zeta(x, y)$ — perturbations of free surface $F = z - \zeta(x, y) = 0$:

$$(\tau_{zz}) \mid_{z=0} = -\rho^0 |\mathbf{g}| \zeta. \tag{3}$$

The evolution part is as follows

$$\frac{\partial S_i}{\partial t} + \mathbf{v} \cdot \boldsymbol{\nabla} S_i = 0. \tag{4}$$

Creeping flow at every time moment $t^{(k)}$ is evaluated in two stages. Firstly, we obtain velocity field $\mathbf{v}^{(k)}$, which is determined by $\rho(\mathbf{x}, t_k)$, i. e. by the configuration of surfaces $S_i^{(k)}$. Then the evolution equation (4) is solved, i. e. surfaces $S_i^{(k)}$ are moved by $\mathbf{v}^{(k)}$ to the next state $S_i{}^{(k+1)}$. New configuration of surfaces $S_i^{(k+1)}$ determines new density distribution $\rho(\mathbf{x}, t_{k+1})$, which provides new velocity field $\mathbf{v}^{(k+1)}$. Thus, the evolution of all layers are described by a sequence of steady states, each subsequent element of which is connected with the previous one through the evolution of surfaces S_i.

For equation (2) the analytical form of Green's function was found in [5], that allows to find $\mathbf{v}$ just as a convolution of this function with density distribution, without using any finite difference methods [1, 6]:

$$\mathbf{v}(\mathbf{x}) = \iiint \mathbf{V}(\mathbf{x}, \boldsymbol{\xi}) \sigma(\boldsymbol{\xi}) d\xi_x d\xi_y d\xi_z, \tag{5}$$

where $\mathbf{V} = (V_x, V_y, V_z)$ — Green's function, $\mathbf{x}$, $\boldsymbol{\xi}$ — Cartesian coordinates. Discrete version of (5) is nothing but:

$$\mathbf{v}(\mathbf{n}) = h_x h_y h_z \sum_{m_x=0}^{N_x-1} \sum_{m_y=0}^{N_y-1} \sum_{m_z=0}^{N_z-1} \mathbf{V}(\mathbf{n}, \mathbf{m}) \sigma(\mathbf{m}) \tag{6}$$

where $\mathbf{n} = (n_x, n_y, n_z)$ and $\mathbf{m} = (m_x, m_y, m_z)$ are discrete coordinates; $n_x = 0, \ldots, N_x - 1$, $n_y = 0, \ldots, N_y - 1$, $n_z = 0, \ldots, N_z - 1$; N_x, N_y, N_z — the size of uniform grid of density distribution; and h_x, h_y, h_z are the grid steps in different directions.

Evaluation of the velocity field $\mathbf{v}$ by (6) has very high computational cost. Direct algorithm requires $O(NM)$ operations, where N — the number of nodes of the density grid and M — the number of points, in which velocity vector is evaluated. To solve equation (4) we need to determine $\mathbf{v}$ field only at surfaces S_i, so, $M \ll N$. However, sometimes it is necessary to get $\mathbf{v}$ on the entire regular grid and $M = O(N)$. In both cases, using the analytical solution of this problem far more preferable than using finite difference methods, which demand $O(N^3)$ operations [4].

However, $O(N^2)$ and even $O(NM)$ is quite high complexity, that makes it impossible to increase significantly the size of the problem, which is important for applications. So, the fast algorithm has been used for evaluation boundary value

[1]This work is focused on salt diapirism and it is a continuation of a series of earlier authors' studies on the subject [1, 6, 7, 2]. However, the presented results can be generalized to a wider class of problems

problems with known explicit form of Green's function (i. e. evaluation expressions like (6)), proposed in [2].

3. FAST NUMERICAL ALGORITHM

One of possible optimization methods consists in using fast convolution algorithms [8], which allow to calculate *discrete cyclic convolution* by $O(N \log N)$ operations instead of $O(N^2)$ operations for direct algorithm. In 3D space it is defined as follows:

$$y(\mathbf{n}) = f(\mathbf{n}) * g(\mathbf{n}) = \\ = \frac{1}{N_x}\frac{1}{N_y}\frac{1}{N_z}\sum_{m_x=0}^{N_x-1}\sum_{m_y=0}^{N_y-1}\sum_{m_z=0}^{N_z-1} f(\mathbf{n}-\mathbf{m})g(\mathbf{m}) \quad (7)$$

where $f(\mathbf{n})$ and $g(\mathbf{n})$ — convolved functions of discrete variables; $n_x = 0, \ldots, N_x - 1$, $n_y = 0, \ldots, N_y - 1$, $n_z = 0, \ldots, N_z - 1$. Functions $f(\mathbf{n})$ and $g(\mathbf{n})$ are *assumed to be periodic* with period $\mathbf{N}$, i. e. $f(\mathbf{n}) = f(k\mathbf{N} + \mathbf{n})$ and $g(\mathbf{n}) = g(k\mathbf{N} + \mathbf{n})$.

Such function can be evaluated fast by well-known algorithms. However, expression (6) is not equal to (7), so we are not able to use these algorithms directly.

The way to overstep this problem was proposed in [2]. First, in our case Green's function of six variables $\mathbf{V}(\mathbf{x}, \boldsymbol{\xi})$ is regarded as the function of three ones: $\mathbf{V}(\mathbf{x} - \boldsymbol{\xi})$. Assume that it is the function of discrete variables: $\mathbf{V}(\mathbf{n} - \mathbf{m})$, where $n_x, m_x = 0, \ldots, N_x - 1$, $n_y, m_y = 0, \ldots, N_y - 1$, $n_z, m_z = 0, \ldots, N_z - 1$. Let us consider the new function $\hat{\mathbf{V}}(n_x, n_y, n_z)$ according to:

$$\hat{\mathbf{V}} = \begin{cases} \mathbf{V}(n_x, n_y, n_z), \\ \qquad n_x < N_x, n_y < N_y, n_z < N_z; \\ \mathbf{V}(n_x - 2N_x + 1, n_y, n_z), \\ \qquad n_x \geq N_x, n_y < N_y, n_z < N_z; \\ \mathbf{V}(n_x, n_y - 2N_y + 1, n_z), \\ \qquad n_x < N_x, n_y \geq N_y, n_z < N_z; \\ \mathbf{V}(n_x, n_y, n_z - 2N_z + 1), \\ \qquad n_x < N_x, n_y < N_y, n_z \geq N_z; \\ \mathbf{V}(n_x - 2N_x + 1, n_y - 2N_y + 1, n_z), \\ \qquad n_x \geq N_x, n_y \geq N_y, n_z < N_z; \\ \mathbf{V}(n_x - 2N_x + 1, n_y, n_z - 2N_z + 1), \\ \qquad n_x \geq N_x, n_y < N_y, n_z \geq N_z; \\ \mathbf{V}(n_x, n_y - 2N_y + 1, n_z - 2N_z + 1), \\ \qquad n_x < N_x, n_y \geq N_y, n_z \geq N_z; \\ \mathbf{V}(n_x - 2N_x + 1, n_y - 2N_y + 1, n_z - 2N_z + 1), \\ \qquad n_x \geq N_x, n_y \geq N_y, n_z \geq N_z; \end{cases} \quad (8)$$

where $\hat{\mathbf{V}}(n_x, n_y, n_z)$ is defined on 3D uniform grid of $(2N_x - 1) \times (2N_y - 1) \times (2N_z - 1)$ nodes. Let us also define a new function $\hat{\sigma}(n_x, n_y, n_z)$ on a grid of the same size:

$$\hat{\sigma} = \begin{cases} \sigma(n_x, n_y, n_z) & , n_x < N_x, n_y < N_y, n_z < N_z; \\ 0 & , otherwise. \end{cases} \quad (9)$$

As it was shown in [2], cyclic convolution of $\hat{\mathbf{V}}(\mathbf{n})$ ($\hat{V}_x, \hat{V}_y, \hat{V}_z$ are treated separately) and $\hat{\sigma}(\mathbf{n})$ provides exact (up to a constant factor) solution to equation (6):

$$\begin{aligned} &\hat{V}_x(\mathbf{n}) * \hat{\sigma}(\mathbf{n}) = v_x(\mathbf{n}), \\ &\hat{V}_y(\mathbf{n}) * \hat{\sigma}(\mathbf{n}) = v_y(\mathbf{n}), \\ &\hat{V}_z(\mathbf{n}) * \hat{\sigma}(\mathbf{n}) = v_z(\mathbf{n}), \quad \text{where :} \\ &n_x = 0, \ldots, N_x - 1; \\ &n_y = 0, \ldots, N_y - 1; \\ &n_z = 0, \ldots, N_z - 1. \end{aligned} \quad (10)$$

These define the fast $O(N \log N)$ algorithm for calculating stationary velocity field, so the overall computational complexity of creeping flow simulation is $O(N \log N)$, too.

4. RESULTS

The fast algorithm, described in this paper, was implemented and tested. For fast calculation of discrete cyclic convolution the classical algorithm, based on the convolution theorem [8], was used. Fast Fourier transforms were calculated via cuFFT program library on GPU and via multithreaded FFTW library on CPU.

The direct algorithm for (6) using GPU computing via NVIDIA® CUDA® technology has been also implemented by the author [1]. The efficiency of this implementation has been verified by the NVIDIA's profiler `nvprof`, and it has shown compute utilization about 80–95 %[2], i. e. the achieved real performance is very close to the peak one.

Figure 3 shows a stage of Rayleigh-Taylor instability evolution at the same model time, "grown up" from the same initial state obtained by the "fast" and direct methods.

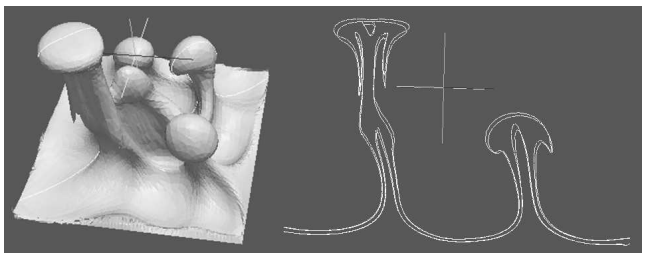

Figure 3: A stage of the model evolution. Left side shows the surface of unstable layer, obtained by the fast method. Right side shows two identical cuts of unstable layer, one was obtained by the direct method, another — by the fast algorithm. Computational grid size is $100 \times 100 \times 100$ **nodes.**

The direct algorithm is more precise to calculate velocity field directly on the surfaces. The fast algorithm provides $\mathbf{v}$ on the uniform grid, so it is necessary to interpolate (linear interpolation was used) velocity field on surfaces to solve (4). Due to this we have some difference on the cuts (see figure 3 right)[3]. In table 1 the performance gain, achieved by the fast algorithm, is shown.

Since the direct algorithm calculates velocity filed $\mathbf{v}$ only on the boundaries (surfaces S_i), the execution time does not quadratically depend on the grid size. Sometimes it is necessary to get $\mathbf{v}$ on the regular grid (the same as density

[2]Depending on the graphics accelerator

[3]This difference is quite acceptable for desired application — simulation of salt diapirism

Table 1: Calculation time on Tesla® M2090 (all evolution)

Grid size	Time (min)		Speedup
	direct (CUDA)	fast (cuFFT)	
75 × 75 × 75	11.4	1.05	**10.8x**
100 × 100 × 100	27.0	2.84	**9.5x**
150 × 150 × 150	144	12.5	**11.5x**

one), and in this case fast algorithm is much more efficient. In the table 2 the times of velocity field **v** calculation on uniform grid are shown.

Table 2: Calculation time on Tesla M2090 (v calculating on uniform grid)

Grid size	Time (sec)		Speedup
	direct (CUDA)	fast (cuFFT)	
64 × 64 × 64	18	1.49	**12.0x**
128 × 128 × 128	1133	2.61	**434.1x**
256 × 256 × 128	18157	29.04	**625.2x**

We also did some tests with famous multithreaded FFTW library on single CPU Intel® Core™ i5-4440, and compared it with direct implementation on single GPU Tesla M2090. Figure 4 clearly shows the advantages of the proposed fast method for calculating **v**. Note that the fast method works on single CPU using FFTW library much faster than the direct algorithm on GPU.

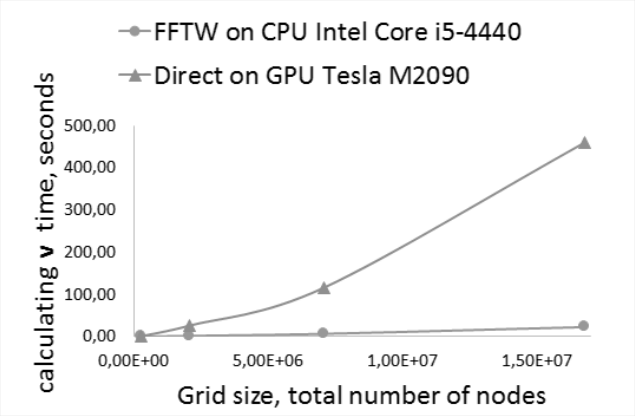

Figure 4: Comparison of calculation speed of fast implementation on CPU and direct one on single GPU depending on the grid size.

5. CONCLUSION

The method for fast numerical simulation of 3D gravitational creeping flow of uniform viscous incompressible Newtonian fluid with piecewise constant density in a half-space bounded by a free surface have been proposed, implemented and tested.

This method is extremely efficient on grids with a large number of nodes (10^6 and more) due to low computational complexity of the algorithm. As the result, valuable acceleration of numerical solution of the given mathematical physics problem, is achieved. The considered problem has great practical importance as oil and gas deposits are often located at salt domes.

Note that evaluating (6) via cyclic convolution (7) is exact, i. e. direct and fast algorithms provide precisely the same results neglecting rounding errors.

6. ACKNOWLEDGMENTS

The work was conducted as a part of the Program of Basic Scientific Research VIII.73.2 SB RAS and supported by the Russian Foundation for Basic Research (Grant no. 16-35-00443 mol-a).

The authors also wish to thank Valery N. Martynov[4] for the idea of applying the convolution theorem to optimize the direct algorithm.

7. REFERENCES

[1] T. V. Abramov. Massively parallel rayleigh-taylor instability simulation using analytical expression of greens function of the corresponding boundary value problem. *Computational technologies*, 20(4):3–16, 2015.

[2] T. V. Abramov. Fast numerical solution of boundary value problems with known greens function using cyclic convolution. *Computational technologies*, 21(2):3–11, 2016.

[3] M. R. Hudec and M. P. A. Jackson. Terra infirma: Understanding salt tectonics. *Earth Science Reviews*, 82(1):1–28, 2007.

[4] A. T. Ismail-Zadeh, I. A. Tsepelev, C. Talbot, and P. Oster. Three-dimensional modeling of salt diapirism: A numerical approach and algorithm of parallel calculations. *Comput. Seism. Geodyn. Amer. Geophys. Union*, (6):33–41, 2004.

[5] B. V. Lunev. Isostasy as the dynamic equilibrium of a viscous fluid. *Doklady AN SSSR*, 290(6):72–76, 1986.

[6] B. V. Lunev and V. V. Lapkovskii. Fast numerical simulation of salt tectonics: possibility of the operational application in geological practice. *Fiz. mezomekh.*, 12(1):63–74, 2009.

[7] B. V. Lunev and V. V. Lapkovskii. Mechanism of development of inversion folding in the subsalt. *Izvestiya, Physics of the Solid Earth*, 50(1):57–63, 2014.

[8] H. Nussbaumer. *Fast Fourier Transform and Convolution Algorithms. Second Corrected and Updated Edition.* Springer-Verlag Berlin and Heidelberg GmbH & Co. KG, Berlin, 1990.

[9] C. J. Talbot and M. P. A. Jackson. Salt tectonics. *Scientific American*, 257(2):70–79, 1987.

[4]Institute of Computational Mathematics and Mathematical Geophysics SB RAS