GPU-accelerated discrete element modeling of geological faults

V. Lisitsa¹,², V. Tchebverda¹, V. Volianskaia³

¹ Institute of Petroleum Geology and Geophysics SB RAS, Novosibirsk, Russia
² Sobolev Institute of Mathematics SB RAS, Novosibirsk, Russia
³ PAO "Rosneft", Moscow, Russia

E-mail: lisitsavv@ipgg.sbras.ru

Abstract. In this paper, we present an original algorithm for the numerical simulation of tectonic movements and the related formation of geological faults. The approach is based on the use of Discrete Element Method, where the geological media is represented as an agglomeration of discrete particles which interact as elastic, visco-elastic, or elasto-plastic bodies. This approach naturally allows accounting for large deformations and discontinuities in the geological media; thus, allows simulation of faults formation. Implementation of the algorithm is based on the use of Graphical Processor Units. The Discrete Element Method requires a high number of floating point operations and logical operations per a single particle (degree of freedom) per time step; whereas the number of memory access operations is relatively low. Thus, the use of GPUs decreases the computational time substantially.

1. Introduction

In seismic processing and interpretation, geological faults are considered and represented as discontinuities in seismic signal phase and further discontinuities in sedimentary, metamorphic or magmatic rock bodies. Thus, no physical properties are assigned to a fault. However, real geological faults have a complex structure which includes the main fault body (“fault core”) and fractured or damage zones around [1], [2]. Damage zone may be highly fractured, thus, permeable especially for carbonates [3], or it can be an impermeable due to the presence of deformation bonds which is typical for the sandstones [4]. Such differences of the local permeability near faults may strongly affect the reservoir performance [5]. Thus an accurate representation of the fault and damage zone is required for efficient oil and gas exploration.

Direct observation and analysis of fault structures are troublesome due to some natural reasons. Thus, numerical simulation is a reliable and efficient way to investigate the peculiarities of the structure’s forming and tectonic movement process. There are several approaches to simulate faults and fractures formation. These includes grid-based methods [6], [7], [8] and meshless approaches also known as discrete elements method (DEM) [9], [10]. The letter is preferred because no predefined crack or fault geometry is needed for simulation. However, particle-based methods are more computationally intense [10]. Despite this, the particle-based methods are incredibly flexible and can be used to generate multiple statistical realizations of the fault zones and study statistical features of the strongly deformed and highly-distorted zones. Meshless methods of geological faults formation simulations can be used to generate faults geometries in realistic environments. After that simulated faults can be introduced in
geological models which are used for seismic modeling and imaging [11], [12], moreover use advanced simulation techniques such as local mesh refinement [13], [14] allow studying seismic responses of the fine structure of near-fault damage zones.

2. Discrete Elements Method

Following [15] and [16], to simulate the tectonic movements causing finite deformations and geological fault formation in the Earth’s crust, we use the discrete element method. In this approach, the media is represented as an assembly of individual particles with a particular geometry and physical properties. Each particle is characterized by the coordinate of its center \( \vec{x} \) and radii \( R_i \) and \( R_j \) respectively. Particle \( i \) acts on particle \( j \) with the normal elastic forces:

\[
F_{n}^{ji} = \begin{cases} 
K_r (R_i + R_j - \|\vec{X}_{ji}\|)\vec{n}_{ji}, & R_i + R_j - \|\vec{X}_{ji}\| > 0, \text{ repulsion}, \\
\sigma_{ji}^r (R_i + R_j - \|\vec{X}_{ji}\|)\vec{n}_{ji}, & 0 \leq R_i + R_j - \|\vec{X}_{ji}\| \leq r_0, \text{ active bond}, \\
0, & R_i + R_j - \|\vec{X}_{ji}\| > r_0, \text{ no bond},
\end{cases}
\]

where \( r_0 \) is the bond length, vector \( \vec{X}_{ji} = \vec{x}_i - \vec{x}_j \) connects the centers of the particles and directed from particle \( j \) to particle \( i \), vector \( \vec{n}_{ji} = \vec{X}_{ji} / \|\vec{X}_{ji}\| \) is the unit vector directed from the centers of particle \( j \) to the center of particle \( i \) or normal vector.

Additionally frictional forces are taken into account if two particles are in a contact [16]:

\[
F_{t}^{ji} = \begin{cases} 
-K_s \delta_t \vec{\upsilon}_t, & K_s \delta_t \leq \mu_s \|F_{n}^{ji}\|, \text{ static friction}, \\
-\mu_i \|F_{n}^{ji}\|\vec{\upsilon}_t, & K_s \delta_t > \mu_s \|F_{n}^{ji}\|, \text{ dynamic friction},
\end{cases}
\]

where \( K_s \) is the tangential sliding stiffness, usually considered to be equal to bulk modulus; i.e., \( K_s = K_r \), vector \( \vec{\upsilon}_{ji} \) is the unitary tangential vector directed along the projection of the relative velocity onto the contact plane of two particles; i.e.,

\[
\vec{\upsilon}_{ji} = \frac{\delta_{ji} - (\vec{\upsilon}_t, \vec{n}_{ji})\vec{n}_{ji}}{\|\delta_{ji} - (\vec{\upsilon}_t, \vec{n}_{ji})\vec{n}_{ji}\|}, \quad \vec{\upsilon}_{ji} = \vec{\upsilon}_i - \vec{\upsilon}_j.
\]

In this notations \( \vec{\upsilon}_{ji} \) is the relative velocity of the particle \( i \) with respect to particle \( j \). Parameter \( \delta_{ji} \) denotes the tangential displacement of the contact point from its initial position. Tangential forces provided by formula (2) satisfy the Coulombs law; i.e., the static friction governs the particles interaction if the forces as below a critical value. If the tangential forces exceed the critical dynamical friction proportional to normal force is applied. Typically the static friction is much higher than the dynamical one.

Additionally, potential external forces are considered, and an artificial dissipation is introduced in the system to prevent elastic waves from propagating through the model and ensuring the media to remain stable at infinite instants.

Having computed all external forces acting at \( j \)-th particle one may recompute its position using classical mechanics principles:

\[
M_i \frac{d^2 \vec{x}_i}{dt^2} = \vec{F}_i \left( t, \vec{x}_i, \vec{\upsilon}_i, \frac{d\vec{x}_i}{dt}, \frac{d\vec{\upsilon}_i}{dt} \right),
\]

(4)
where dissipative $\vec{F}_d^i$ and frictional forces $\vec{F}_t^{ji}$ explicitly depend on the particles velocities $\vec{v}^i = \frac{d\vec{x}^i}{dt}$.

To numerically resolve the system of equations (4), we use the Verlet-like scheme with the velocity half-step [17].

### 2.2. Boundary conditions

Proper implementation of the boundary conditions is a challenging task for the particles-based methods. In our research, we deal with two types of boundary conditions. First, we impose the rigid boundary condition; i.e., the surface $\Gamma_s$ is fixed, or its movement is prescribed. Moreover, it is stiff; thus, the particles cannot penetrate through it. Formally, this type of boundary condition can be stated as follows. Assume a boundary $\Gamma_s = \{ \vec{x}|x_2 = x_{2r}^j \}$. If a particle is close enough to the boundary; i.e., if for the j-th particle $|x^j_2 - x_{2r}^j| \leq R^j$, then $F_j^B = K_r(R^j - |x^j_2 - x_{2r}^j|)$.

However, the numerical implementation of this condition requires extra conditional operators. Thus it is worth implementing stiff-boundary as a series of particles, to make the simulation uniform either in the interior of the domain or near the boundary. To do so, we introduced the ”boundary” particles with the same physical properties as those of the interior particles. However, we do not compute the forces acting on the ”boundary” particles but allow the ”boundary” particles to move according to a prescribed law.

The second type of boundary conditions is $P_{over} = const$. This condition ensures constant overburden pressure. Note that, condition $P_{over}$ assumes that external forces act at the upper boundary of the domain $\Gamma_p(t)$ along the normal direction to the boundary. This boundary is flexible, and it evolves in time; thus, to impose the boundary condition, we need to follow the elements which form the upper boundary. This can be done, for example, by computing Voronoi diagrams for upper elements. However, such procedures are computationally intense. To overcome this difficulty, we suggest using the flexible membrane at the upper boundary [18], [19]. The idea of the approach is to introduce a layer of discrete elements so that the membrane elements are affected only by the normal forces.

If two adjoin membrane elements are interacting

$$F_{nm,m^\pm 1} = K_r(R_{m^\pm1} + R^m - ||\vec{X}_{m,m^\pm1}||)\vec{n}_{m,m^\pm1},$$

(5)

if membrane element interacts with other elements

$$F_{ni} = K_r(R^i + R^m - ||\vec{X}^{mi}||)\vec{n}^{mi}, \quad R^i + R^m - ||\vec{X}^{mi}|| > 0.$$  

(6)

It means that the adjoin membrane elements are bonded, and these bonds never break, however no bonds of friction are considered when membrane elements interact with elements of other types. The membrane elements are ordered; thus, it is easy to approximate constant pressure condition. If a membrane element with number $m$ is considered then additional force, due to the pressure is

$$F_{p} = 2PR^m\vec{n},$$

(7)

where $\vec{n}$ is the vector normal to the boundary, which can be computed as:

$$\vec{n} = (x_2^{m-1} - x_2^{m+1}, x_1^{m-1} - x_1^{m+1})^T,$$

the direction of the normal vector is defined uniquely due to the ordering of the membrane elements.
2.3. Output parameters
Numerous parameters can be obtained as a result of discrete element simulations. If rock properties are studied using uniaxial and triaxial stress tests, then the primary attention is paid to the distribution of the braked bonds [20], [21], stresses, and normal forces distribution [22]. However, at the scale of the geological bodies a reliable parameter to determine fault zones is the strains distribution [23], [15], [17], [24], [25]. These strains can be further translated to the changes of physical parameters of rocks using the experimental laboratory measurements.

To estimate the strains distribution, one may compute the relative displacements element-wise, after that the components of the strain tensor components can be computed and interpolated to a regular grid. A detailed discussion of the estimation of the strain can be found in [23].

3. Implementation of the Algorithm
According to the general formulation of the particle-based methods, one has to compute the forces affecting each particle due to the interaction with all other particles. However, in geomechanical modeling by the discrete element method, for each particle, only a small number of neighboring particles directly contact the considered one. The adjacency matrix is sparse, but it can evolve. Thus, two related problems should be solved. First, organizing the process of adjacency matrix construction (approximation). Second, computing forces and applying time stepping.

To construct the adjacency matrix, we suggest using the lattice method. As it follows from the equations (1) and (2), only directly contacting particles affect each other; thus, for each particle, the domain of dependence does not exceed $2R_{\text{max}} + r_0$, where $R_{\text{max}}$ is the maximal radius of the particles. Also, due to the stability criterion of the Verlet scheme, a single particle cannot move more than $0.1 R_{\text{min}}$ per a single time step, where $R_{\text{min}}$ is the minimum radius over all particles. Thus, we can introduce a grid with the lattice size equal to $2R_{\text{max}} + r_0$, so that each particle and all its neighbors belong to the same lattice of directly adjoint lattice. Now we can state the rule of adjacency matrix approximation - for each particle, all the particles belonging to the same or directly adjoint lattices are neighbors. In this case, we overestimate the number of connected particles but strictly simplify the process of the matrix construction.

The initial assignment of the particles to the lattices is performed by a sequential code by CPU. It is implemented particle-by-particle so that we determine the lattice number for considered particle and add the particle number to the list of particles for this lattice. This procedure is inapplicable under OMP of CUDA parallelization. Thus, the GPU implementation of the reassignment of the particles to the lattices is done lattice-by-lattice. The lattices are large enough, so that after one time step a particle may either stay in the same lattice or move to a directly adjacent lattice. Thus, to update the list of particles for each lattice, we need to check the particles which previously belonged to this lattice or the directly adjusted one. Similar ideas are used in the molecular dynamics and lattice Boltzmann methods but with different principles of lattices construction [26].

Computation of the forces and the numerical solution of the equation of motion is implemented on GPU. The parallelization is applied particle-by-particle so that a GPU core compute forces for one particle at a time.

4. Numerical experiments
We use the algorithm to simulate geological fault position for different directions of tectonic displacements in 2D. The computational domain was 4000 m by 500 m. The boundary conditions introduced the displacements. We consider five scenarios with the displacement direction equal to $0^\circ$ (vertical displacement), $15^\circ$, $30^\circ$, $45^\circ$, and $60^\circ$ from vertical direction pointing downward. So, in all cases, these are normal dip-slip faults. The vertical displacement was 100 m. Radii
of the elements were homogeneously distributed from 0.75 to 1.5 m. The dynamic friction coefficient $\mu_d = 0.3$. We provide the results of simulation - distribution of the horizontal strains $\varepsilon_{xx}$ in figure 1. Note, that the sub-vertical displacements lead to the formation of a wide damage zone, which has a form of a wedge. The closer the displacements to the horizontal direction, the narrower is the wedge with higher deformations in the fault body. Moreover, the main direction of the fault; i.e., the dip angle tends to about 30°, which is governed by the properties of the media, rather than displacement direction.

![Image of horizontal strains distribution](image-url)

**Figure 1.** Distribution of horizontal strains $\varepsilon_{xx}$ for the displacement directions 0° (a), 30° (b), and 60° (c)

5. Conclusions
We presented an algorithm for numerical simulation of the geological fault geometry and position. The approach is based on the Discrete Elements Method, which allows modeling of finite deformations and structural discontinuities in the Earth’s crust. We implement the algorithm
using CUDA technology to simulate a single statistical realization of the model, whereas MPI is used to parallelize for different statistical realizations. The presented numerical experiments illustrate the formation of the V-shaped fault damage zone. The obtained strains distributions can be used further to estimate the mechanical and transport properties of the fault damage zone.

Acknowledgments
V. Lisitsa did development of the mathematical basis of the algorithm in IPGG SB RAS under financial support of the Russian Precedent agency grant MD-20.2019.5. Parallel implementation of the algorithm was done by V. Lisitsa in IM SB RAS under the support of the Russian Science Foundation grant no. 19-77-20004. Numerical experiments were carried out by V. Tcheverda under the support of the Russian Science Foundation grant no. 17-17-01128. V. Volianskaia made the geological interpretation. The research is carried out using the equipment of the shared research facilities of HPC computing resources at Lomonosov Moscow State University and cluster NKS-30T+GPU of the Siberian supercomputer center.

References
[1] Choi J H, Edwards P, Ko K and Kim Y S 2016 Earth-Science Reviews 152 70–87
[2] Peacock D C P, Dimmen V, Rotevatn A and Sanderson D J 2017 Journal of Structural Geology 102 179–192
[3] Hausegger S, Kurz W, Rabitsch R, Kiechl E and Brosch F J 2010 Journal of Structural Geology 32 1349–1362
[4] Fossen H, Schultz R A, Shipton Z K and Mair K 2007 Journal of the Geological Society 164 755–769
[5] Bense V F, Gleeson T, Loveless S E, Bour O and Scibek J 2013 Earth-Science Reviews 127 171–192
[6] González G, Gerbault M, Martinod J, Cembrano J, Carrizo D, Allmendinger R and Espina J 2008 Journal of Structural Geology 30 791–808
[7] Guitton M L E, Sassi W, Leroy Y M and Gauthier B D M 2003 Journal of Structural Geology 25 1317–1330
[8] Resor P G and Pollard D D 2012 Journal of Structural Geology 41 98–109
[9] Gray G G, Morgan J K and Sanz P F 2014 Journal of Structural Geology 59 19–36
[10] Lisjak A and Grasselli G 2014 Journal of Rock Mechanics and Geotechnical Engineering 6 301–314
[11] Kolyukhin D R, Lisitsa V V, Protasov M I, Qu D, Reshetova G V, Tveranger J, Tcheverda V A and Vishnevsky D M 2017 Interpretation 5 SP71–SP82
[12] Vishnevsky D M, Kolyukhin D R, Lisitsa V V, Protasov M I, Reshetova G V, Tcheverda V A, Qu D and Tcheverda J 2017 Doklady Earth Sciences 473 477–481
[13] Kostin V, Lisitsa V, Reshetova G and Tcheverda V 2015 Journal of Computational Physics 281 669–689
[14] Lisitsa V, Reshetova G and Tcheverda V 2012 Computational Geosciences 16 39–54
[15] Hardy S and Finch E 2005 Basin Research 17 507–520
[16] Ludwig S 2008 European Journal of Environmental and Civil Engineering 12 785–826
[17] Hardy S, McClay C K and Munoz J A 2009 Marine and Petroleum Geology 26 232–248
[18] Wang Y and Tonon F 2009 Journal of Engineering Mechanics 135 1029–1037
[19] Bono J D, Mcdowell G and Wanatowski D 2012 Géotechnique Letters 2 199–203
[20] Cundall P A and Strack O D L 1979 Géotechnique 29 47–65
[21] Li Z, Wang Y H, Ma C H and Mok C M B 2017 Acta Geotechnica 12 987–1002
[22] Duan K, Kwok C Y and Ma X 2017 Acta Geotechnica 12 495–510
[23] O’Sullivan C, D Bray J and Li S 2003 International Journal for Numerical and Analytical Methods in Geomechanics 27 859–877
[24] Abe S, Gent H v and Urai J L 2011 Tectonophysics 512 12–21
[25] Botter C, Cardozo N, Hardy S, Lecomte I and Escalona A 2014 Marine and Petroleum Geology 57 187–207
[26] Alpak F O, Gray F, Saxena N, Dieterich J, Hofmann R and Berg S 2018 Computational Geosciences 22 815–832
[27] Calinski T and Harabasz J 1974 Communications in Statistics 3 1–27
[28] Rousseuw P J 1987 Journal of Computational and Applied Mathematics 20 53–65