



GPU-Based Discrete Element Modeling of Geological Faults

Vadim Lisitsa¹(✉), Dmitriy Kolyukhin², Vladimir Tcheverda²,
Victoria Volianskaia³, and Viatcheslav Priimenko⁴

¹ Institute of Petroleum Geology and Geophysics SB RAS,
Sobolev Institute of Mathematics SB RAS, Novosibirsk, Russia
lisitsavv@ipgg.sbras.ru

² Institute of Petroleum Geology and Geophysics SB RAS, Novosibirsk, Russia
{kolyukhindr, cheverdava}@ipgg.sbras.ru

³ PAO RosNeft, Moscow, Russia
v.volyanskaya@rosneft.ru

⁴ North Fluminense State University Darcy Ribeiro,
Campos dos Goytacazes, Brazil
slava@cana.uenf.br

Abstract. In this paper, we present an algorithm for numerical simulation of the tectonic movements leading to the formation of geological faults. We use the discrete element method, so that the media are presented as an agglomeration of elastic, visco-elastic, or elasto-plastic interacting particles. This approach can naturally handle finite deformations and can account for the structural discontinuities in the Earth crust. We implement the algorithm using CUDA technology to simulate single statistical realization of the model, whereas MPI is used to parallelize with respect to different statistical realizations. Obtained numerical results show that for low dip angles of the tectonic displacements relatively narrow faults form, whereas high dip angles of the tectonic displacements lead to a wide V-shaped deformation zones.

Keywords: Geomechanics · Discrete elements method · CUDA · MPI

1 Introduction

A classical definition of the geological faults is that they are discontinuities of 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 main fault body (“fault core”) and fractured or damage zones around [7, 25]. In particular, damage zone may be highly fractured, thus, permeable especially for carbonates [16], or it can be an impermeable due to the presence of deformation bonds which is typical for the sandstones [10]. Such differences of the local permeability near faults may strongly affect the reservoir performance [3]. Thus a detailed representation of the fault and damage zone is required for efficient oil and gas exploration.

Very often, to do field observations or laboratory studies of the real fault is difficult or impossible 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 numerous techniques to simulate finite deformations in geological formations either grid-based methods such as finite differences [11], finite elements [13], boundary elements [26] or by meshless approaches also known as discrete elements method (DEM) [12, 22]. The latter is preferred because no predefined crack or fault geometry is needed for simulation. However, particle-based methods are more computationally intense and require calibration of the particle properties to match the mechanics of the whole body [22]. 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. In our opinion, 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 [17, 28], moreover use advanced simulation techniques such as local mesh refinement [18, 21] allow studying seismic responses of the fine structure of near-fault damage zones.

In this paper, we present an algorithm based on the Discrete Elements Method (DEM). This approach is based on the media representation by a set of discrete particles. These particles interact as stiff elastic bodies according to the mechanical rules; i.e., elastic and frictional forces affect each particle, that leads to the particle movement according to the Newton mechanics [14]. Computation of the forces affecting a particle includes a high number of floating point and logical operations; thus it is computationally intense and hard to implement on CPU, using vectorization, etc. As a result, the efficiency of the CPU based realizations of DEM is low, and computation time to solve even a 2D problem may be as long as several thousand node-hours. On the contrary, GPU architecture is more appropriate for DEM implementation, because it can efficiently handle a big number of flops with a small amount of memory involved in computations.

2 Discrete Elements Method

To simulate the tectonic movements causing finite deformations and geological fault formation in the Earth's crust we use the discrete element method, following [14, 23]. 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 \mathbf{x}^j , radius R_j , repulsion and attraction bulk moduli K_r^+ and K_r^- respectively, tangential sliding stiffness K_s , and two friction coefficients μ_s is the static one and μ_d is the dynamic friction coefficient. Having set these parameters, one may define the interaction forces between two adjoint particles.

2.1 Forces Computation

Consider two particles with the numbers i and j , with the coordinates \mathbf{x}^i and \mathbf{x}^j and radii R^i and R^j respectively. Particle j acts on particle i with the normal elastic forces:

$$\mathbf{F}_n^{ji} = \begin{cases} K_r^-(R^i + R^j - \|\mathbf{X}^{ji}\|)\mathbf{n}^{ji}, & R^i + R^j - \|\mathbf{X}^{ji}\| > 0, & \text{repulsion,} \\ K_r^+(R^i + R^j - \|\mathbf{X}^{ji}\|)\mathbf{n}^{ji}, & 0 \leq R^i + R^j - \|\mathbf{X}^{ji}\| \leq r_0, & \text{active bond,} \\ 0, & R^i + R^j - \|\mathbf{X}^{ji}\| > r_0, & \text{no bond,} \end{cases} \quad (1)$$

where r_0 is the bond length, typically chosen equal to $0.05(R_i + R_j)$, vector $\mathbf{X}^{ji} = \mathbf{x}^i - \mathbf{x}^j$ connects the centers of the particles and directed from particle j to particle i , vector $\mathbf{n}^{ji} = \mathbf{X}^{ji}/\|\mathbf{X}^{ji}\|$ is the unit vector directed from the centers of particle j to the center of particle i or normal vector, because it is normal to the contact plane. Note, that we use the model of linear elastic particles interaction and assume that the repulsion and attraction bulk moduli coincide, which is mainly valid for geomaterials across a wide range of scales.

Additionally frictional forces are taken into account if two particles are in a contact [23]:

$$\mathbf{F}_t^{ji} = \begin{cases} -K_s\delta_t\mathbf{t}^{ji}, & K_s\delta_t \leq \mu^s\|\mathbf{F}_n^{ji}\|, & \text{static friction,} \\ -\mu^d\|\mathbf{F}_n^{ji}\|\mathbf{t}^{ji}, & K_s\delta_t > \mu^s\|\mathbf{F}_n^{ji}\|, & \text{dynamic friction,} \end{cases} \quad (2)$$

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

$$\mathbf{t}^{ji} = \frac{\mathbf{v}^{ji} - (\mathbf{v}^{ji} \cdot \mathbf{n}^{ji})\mathbf{n}^{ji}}{\|\mathbf{v}^{ji} - (\mathbf{v}^{ji} \cdot \mathbf{n}^{ji})\mathbf{n}^{ji}\|}, \quad \mathbf{v}^{ji} = \mathbf{v}^i - \mathbf{v}^j. \quad (3)$$

In this notations \mathbf{v}^{ji} is the relative velocity of the particle i with respect to particle j . Parameter δ^{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, 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:

$$\mathbf{F}_d^i = -\nu\mathbf{v}^i, \quad (4)$$

where ν is an artificial viscosity.

The Earth's crust also remains under gravitational forces which are accounted as

$$\mathbf{F}_g^i = M_i g \mathbf{e}_3, \quad (5)$$

where $g = 9.8$ m/s is the gravitational constant, $\mathbf{e}_3 = (0, 0, 1)^T$, and M_i is the mass of the considered particle.

To compute the total forces acting at a particle one needs to account the forces due to interactions with all the neighbors, plus artificial dissipation, plus gravitational forces, as a result, one gets:

$$\mathbf{F}^i = \sum_{j \in J(i)} \left[\mathbf{F}_n^{ji} + \mathbf{F}_t^{ji} \right] + \mathbf{F}_d^i + \mathbf{F}_g^i, \quad (6)$$

where $J(i)$ is the set of indexes of the neighbors of i -th particle.

2.2 Time Integration

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

$$M^i \frac{d^2 \mathbf{x}^i}{dt^2} = \mathbf{F}^i \left(t, \mathbf{x}^i, \mathbf{x}^j, \frac{d\mathbf{x}^i}{dt}, \frac{d\mathbf{x}^j}{dt} \right), \quad (7)$$

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

To numerically resolve system of Eq. (7) we use the Verlet-like scheme with the velocity half-step [15]. Assume coordinates, velocities, and thus forces of all particles are known at instant $t = t^n = \tau \cdot n$, then they can be updated to the instant t^{n+1} by the rule:

$$\begin{aligned} \frac{(\mathbf{v}^i)^{n+1/2} - (\mathbf{v}^i)^n}{\tau/2} &= \frac{1}{M^i} \mathbf{F}^i \left(t^n, (\mathbf{x}^i)^n, (\mathbf{x}^j)^n, (\mathbf{v}^i)^n, (\mathbf{v}^j)^n \right), \\ \frac{(\mathbf{x}^i)^{n+1} - (\mathbf{x}^i)^n}{\tau} &= (\mathbf{v}^i)^{n+1/2}, \\ \frac{(\mathbf{v}^i)^{n+1} - (\mathbf{v}^i)^{n+1/2}}{\tau/2} &= \frac{1}{M^i} \mathbf{F}^i \left(t^{n+1}, (\mathbf{x}^i)^{n+1}, (\mathbf{x}^j)^{n+1}, (\mathbf{v}^i)^{n+1/2}, (\mathbf{v}^j)^{n+1/2} \right), \\ j &\in J(i). \end{aligned} \quad (8)$$

In case of no explicit dependence of forces on the velocities the scheme is the second order accurate, however if applied to the equation of motion for DEM, this scheme possesses only the first order of approximation.

To ensure the stability of the finite-difference scheme we use the time step as suggested in [15]

$$\tau \leq 0.2 \frac{D_{min}}{V_{max}}, \quad (9)$$

where R_{min} is the minimum diameter of the particles, and V_{max} is the maximal velocity of perturbation propagation in the system.

2.3 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 Γ_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 $\Gamma_s = \{\mathbf{x} | x_2 = x_2^B\}$. If a particle is close enough to the boundary; i.e., if for the j -th particle $|x_2^j - x_2^b| \leq R^j$, then $F_2^{jB} = K_r^-(R^j - |x_2^j - x_2^b|)$.

However, 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. We specify the particular movement laws in the Sect. 4.

The second type of the boundary conditions is $P_{over} = const$. This condition ensures the 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 [4, 29]. 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 adjoint membrane elements are interacting

$$\mathbf{F}_n^{m,m\pm 1} = K_r(R^{m\pm 1} + R^m - \|\mathbf{X}^{m,m\pm 1}\|)\mathbf{n}^{m,m\pm 1}, \quad (10)$$

if membrane element interacts with other elements

$$\mathbf{F}_n^{mi} = K_r(R^i + R^m - \|\mathbf{X}^{mi}\|)\mathbf{n}^{mi}, R^i + R^m - \|\mathbf{X}^{mi}\| > 0. \quad (11)$$

It means that the adjoint membrane elements are bonded, and these bonds never bake, 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

$$\mathbf{F}_p^m = 2PR^m\mathbf{n}, \quad (12)$$

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

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

the direction of the normal vector is defined uniquely due to the ordering of the membrane elements.

2.4 Output Parameters

Numerous parameters can be obtained as a result of discrete elements 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 [8, 19], stresses, and normal forces distribution [9]. However, at the scale of the geological bodies a reliable parameter to determine fault zones is the strains distribution [1, 5, 14, 15, 24]. These strains can be further translated to the changes of physical parameters of rocks using the experimental laboratory measurements.

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 [20]. As it follows from the Eqs. (1) and (2), only directly contacting particles affect each other; thus, for each particle, the domain of dependence does not exceed $2R_{max} + r_0$, where R_{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 $0.1R_{min}$ per a single time step, where R_{min} is the minimal radius over all particles. Thus, we can introduce a grid with the lattice size equal to $2R_{max} + r_0$, 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 adjoint 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 [2].

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

In this paper, we focus our attention on the effect of the direction and amplitude of tectonic movement on the geometry of the fault and damage zone. DEM-based simulations include uncertainties due to the particle's positions and radii distributions. It means that for each scenario of the tectonic movements we need to perform a series of numerical simulations for different statistical realizations of the particles geometry distribution.

In all the experiments presented below, we use the following set of parameters. The size of the computational domain is 4000 m in horizontal and 500 m in the vertical direction. The repulsion/attraction modulus is 16 GPa, and same value is used for the tangential sliding stiffness. The coefficient of static friction is 0.8, which is typical for the majority of geomaterials, whereas the dynamic friction coefficient is 0.3, which is close to that of sandstone and limestone. We consider the bonds length proportional to the radii of the adjoint particles; i.e., $r_0 = 0.05(R_j + R_i)$. We assume that the formation is buried at 3000 m; thus the overburden pressure of 10^7 Pa is applied at the top boundary of the model. The particles radii are homogeneously distributed from 1.25 to 2.5 m. So, the total number of elements is 390000.

We consider several scenarios of dipping normal tectonic movements with the dip angles equal to 90° , 75° , 60° , 45° , 30° . Maximal vertical displacement is 100 m.

For each tectonic movement we simulate ten statistical realizations of the particles distributions; thus, 10 simulations are performed for each scenario. Also, we computed extra 20 realizations for the most common movement scenario with the dip angle equal to 60° . Each simulation consists of two stages. First, the elements should be compacted under the overburden pressure and gravitational forces. This step takes about 60 % of the computational time. Second, the tectonic movements are applied. The total simulation time for one experiment (one realization) is about 8.7 h by a single GPU (NVIDIA Tesla M 2090).

We provide the strains distribution for each movement scenario in Figs. 1, 2 and 3. The main trend observed from the presented figures is that for big dip angles; i.e., for nearly vertical displacements no narrow fault cores are formed. When the dip angle gets smaller fault cores are formed (Fig. 1) and they are located within a narrow zone. Moreover, for low dip angles the form of the fault and its inclination is similar, thus might depend mainly on the medium properties rather than on the direction of tectonic movements. To verify this assumption, we perform clustering of the results and their statistical analysis in the following section.

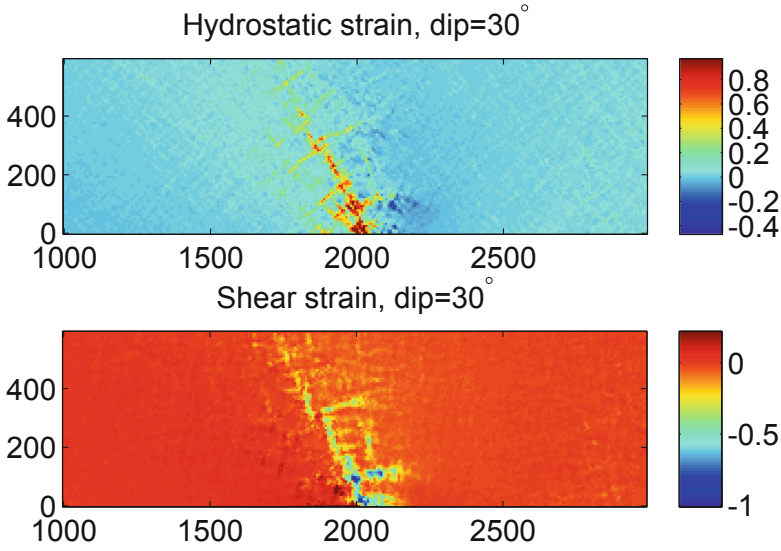


Fig. 1. A single realization of hydrostatic (top) and shear (bottom) strains distribution in the fault zone for the displacement dip equal to 30° .

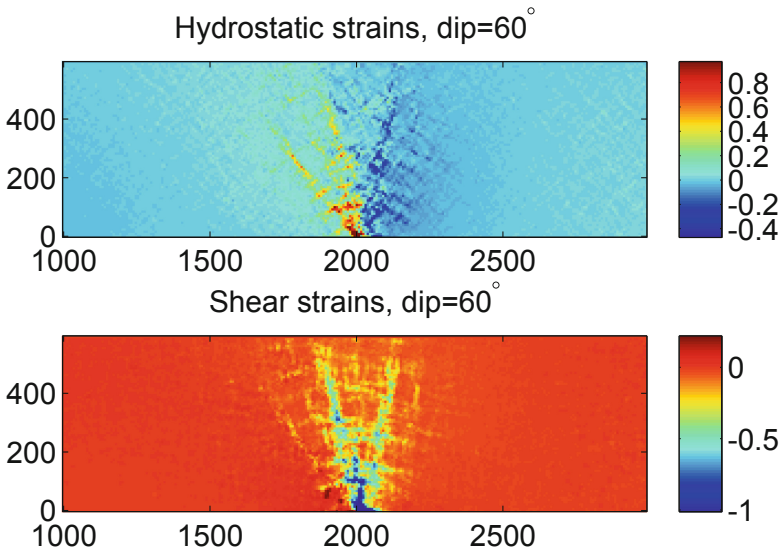


Fig. 2. A single realization of hydrostatic (top) and shear (bottom) strains distribution in the fault zone for the displacement dip equal to 60° .

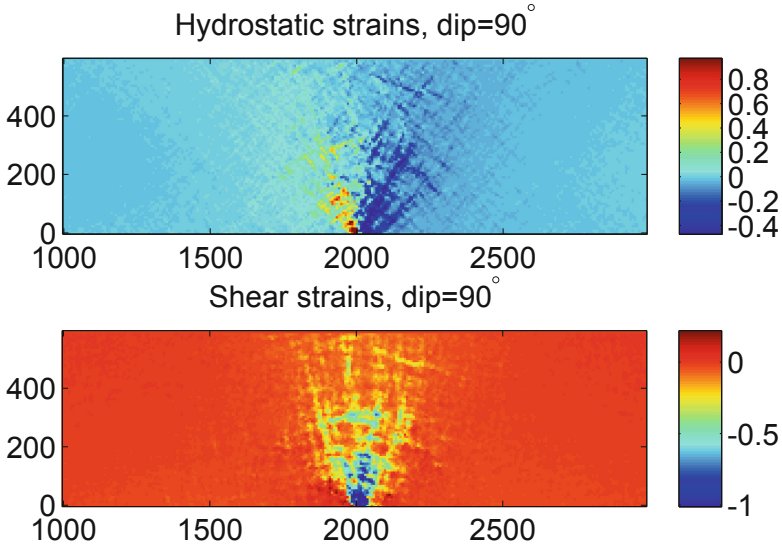


Fig. 3. A single realization of hydrostatic (top) and shear (bottom) strains distribution in the fault zone for the displacement dip equal to 90° .

4.1 Clustering of the Results

Results of the numerical simulation tend to form narrow inclined faults if the dipping angle is small, whereas high dipping angles cause a wide V-shape damage zone. To quantify this observation, we applied k-means clustering of the computed strains distribution. Before processing to the formal analysis, we need to point out, that we performed two additional series of simulations (9 realizations in each series) corresponding to the tectonic movement dip angle equal to 60° . In total we have 27 statistical realizations corresponding to this scenario; however, we will still consider them as three independent series in our statistical analysis.

According to the Calinski-Harabasz Index [6] and the silhouette criterion [27] the optimal number of clusters of the considered data is two. We applied the k-means clustering technique to our data. We constructed clusters for each component of the strain tensor separately, as well as for all of them together. The panels in Fig. 4 represent the clustering results in two clusters applied to all components of the strain tensor. One may note that the displacement scenarios with dip angles equal to 75° and 90° form one cluster. This confirms the assumption that direction of the tectonic movement strictly affects the structure of the fault and near fault damage zone.

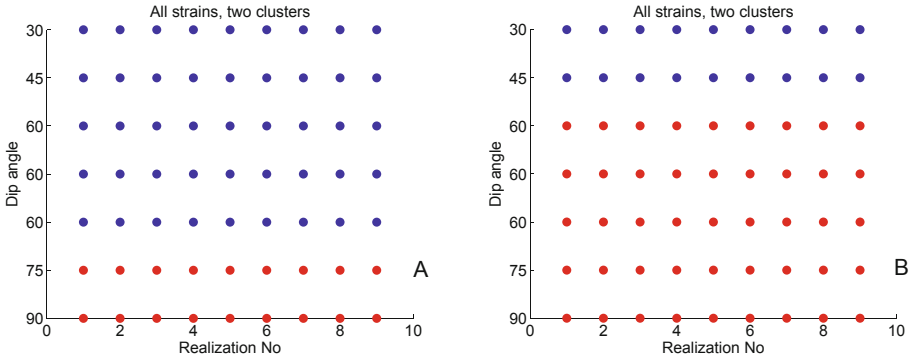


Fig. 4. Panels representing data clustering (two clusters) for all components of strain tensor. Left panel (A) corresponds to the optimal clustering with minimal distance, right panel (B) represents a case of local minimum of k-means functional. Different colors correspond to different clusters.

5 Conclusions

We presented an algorithm for simulation of the Earth's crust tectonic movements and formation of the geological faults and near-fault damage zones. The algorithms are based on the Discrete Elements Method, and it is implemented using CUDA technology. We used to simulate faults formation due to different scenarios of tectonic movements. We considered the displacements with dipping angles varied from 30 to 90 degrees; i.e., up to vertical throw. For each scenario, we performed simulations for some statistical realizations. According to clustering analysis shows that displacements with high (75° and 90°) and low (30° and 45°) dip angles form completely different geological structures. Nearly vertical displacements, high dip angles, form wide V-shaped deformation zones, whereas the flat displacements cause narrow fault-cores with rapidly decreasing strains apart from the fault core. Results of the presented simulations can be used to estimate mechanical and seismic properties of rocks in the vicinity of the faults and applied further to construct models for seismic modeling and interpretation, hydrodynamical simulations, history of matching simulation, etc.

Acknowledgements. Development of mathematical basis of the algorithm was done by V. Lisitsa in IPGG SB RAS under financial support of the grant of the Precedent of the Russian Federation for young researchers MD-20.2019.5. Parallel implementation of the algorithm was done by V. Lisitsa in IM SB RAS under support of Russian Science Foundation grant no. 19-77-20004. D. Kolyukhin performed the statistical analysis of the results under support of Russian Foundation for Basic Research grant no. 18-05-00031. Numerical experiments were carried out by V. Tcheverda under support of the Russian Science Foundation grant no. 17-17-01128. V. Priimenko applied the clustering analysis of the results. V. Volianskaia did 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. Abe, S., Gent, H., Urai, J.: DEM simulation of normal faults in cohesive materials. *Tectonophysics* **512**, 12–21 (2011). <https://doi.org/10.1016/j.tecto.2011.09.008>
2. Alpak, F.O., Gray, F., Saxena, N., Dietderich, J., Hofmann, R., Berg, S.: A distributed parallel multiple-relaxation-time lattice-Boltzmann method on general-purpose graphics processing units for the rapid and scalable computation of absolute permeability from high-resolution 3D micro-CT images. *Comput. Geosci.* **22**(3), 815–832 (2018). <https://doi.org/10.1007/s10596-018-9727-7>
3. Bense, V.F., Gleeson, T., Loveless, S.E., Bour, O., Scibek, J.: Fault zone hydrogeology. *Earth Sci. Rev.* **127**, 171–192 (2013). <https://doi.org/10.1016/j.earscirev.2013.09.008>
4. de Bono, J., McDowell, G., Wanatowski, D.: Discrete element modelling of a flexible membrane for triaxial testing of granular material at high pressures. *Géotechnique Lett.* **2**(4), 199–203 (2012). <https://doi.org/10.1680/geolett.12.00040>
5. Botter, C., Cardozo, N., Hardy, S., Lecomte, I., Escalona, A.: From mechanical modeling to seismic imaging of faults: a synthetic workflow to study the impact of faults on seismic. *Mar. Pet. Geol.* **57**, 187–207 (2014). <https://doi.org/10.1016/j.marpetgeo.2014.05.013>
6. Calinski, T., Harabasz, J.: A dendrite method for cluster analysis. *Commun. Stat.* **3**(1), 1–27 (1974). <https://doi.org/10.1080/03610927408827101>
7. Choi, J.-H., Edwards, P., Ko, K., Kim, Y.-S.: Definition and classification of fault damage zones: a review and a new methodological approach. *Earth Sci. Rev.* **152**, 70–87 (2016). <https://doi.org/10.1016/j.earscirev.2015.11.006>
8. Cundall, P.A., Strack, O.D.L.: A discrete numerical model for granular assemblies. *Géotechnique* **29**(1), 47–65 (1979). <https://doi.org/10.1680/geot.1979.29.1.47>
9. Duan, K., Kwok, C.Y., Ma, X.: DEM simulations of sandstone under true triaxial compressive tests. *Acta Geotechnica* **12**(3), 495–510 (2017). <https://doi.org/10.1007/s11440-016-0480-6>
10. Fossen, H., Schultz, R.A., Shipton, Z.K., Mair, K.: Deformation bands in sandstone: a review. *J. Geol. Soc.* **164**(4), 755–769 (2007). <https://doi.org/10.1144/0016-76492006-036>
11. González, G., et al.: Crack formation on top of propagating reverse faults of the Chuculay fault system, northern Chile: Insights from field data and numerical modelling. *J. Struct. Geol.* **30**(6), 791–808 (2008). <https://doi.org/10.1016/j.jsg.2008.02.008>
12. Gray, G., Morgan, J., Sanz, P.: Overview of continuum and particle dynamics methods for mechanical modeling of contractional geologic structures. *J. Struct. Geol.* **59**(Suppl. C), 19–36 (2014). <https://doi.org/10.1016/j.jsg.2013.11.009>
13. Guiton, M.L.E., Sassi, W., Leroy, Y.M., Gauthier, B.: Mechanical constraints on the chronology of fracture activation in folded Devonian sandstone of the western Moroccan anti-atlas. *J. Struct. Geol.* **25**(8), 1317–1330 (2003). [https://doi.org/10.1016/S0191-8141\(02\)00155-4](https://doi.org/10.1016/S0191-8141(02)00155-4)
14. Hardy, S., Finch, E.: Discrete-element modelling of detachment folding. *Basin Res.* **17**(4), 507–520 (2005). <https://doi.org/10.1111/j.1365-2117.2005.00280.x>

15. Hardy, S., McClay, K., Munoz, J.A.: Deformation and fault activity in space and time in high-resolution numerical models of doubly vergent thrust wedges. *Mar. Pet. Geol.* **26**, 232–248 (2009). <https://doi.org/10.1016/j.marpetgeo.2007.12.003>
16. Hausegger, S., Kurz, W., Rabitsch, R., Kiechl, E., Brosch, F.J.: Analysis of the internal structure of a carbonate damage zone: implications for the mechanisms of fault Breccia formation and fluid flow. *J. Struct. Geol.* **32**(9), 1349–1362 (2010). <https://doi.org/10.1016/j.jsg.2009.04.014>
17. Kolyukhin, D., et al.: Seismic imaging and statistical analysis of fault facies models. *Interpretation* **5**(4), SP71–SP82 (2017). <https://doi.org/10.1190/int-2016-0202.1>
18. Kostin, V., Lisitsa, V., Reshetova, G., Tcheverda, V.: Local time-space mesh refinement for simulation of elastic wave propagation in multi-scale media. *J. Comput. Phys.* **281**, 669–689 (2015). <https://doi.org/10.1016/j.jcp.2014.10.047>
19. Li, Z., Wang, Y.H., Ma, C.H., Mok, C.M.B.: Experimental characterization and 3D DEM simulation of bond breakages in artificially cemented sands with different bond strengths when subjected to triaxial shearing. *Acta Geotechnica* **12**(5), 987–1002 (2017). <https://doi.org/10.1007/s11440-017-0593-6>
20. Lisitsa, V., Tcheverda, V., Volianskaia, V.: GPU-based implementation of discrete element method for simulation of the geological fault geometry and position. *Supercomput. Front. Innov.* **5**(3), 46–50 (2018). <https://doi.org/10.14529/jsfi180307>
21. Lisitsa, V., Reshetova, G., Tcheverda, V.: Finite-difference algorithm with local time-space grid refinement for simulation of waves. *Comput. Geosci.* **16**(1), 39–54 (2012). <https://doi.org/10.1007/s10596-011-9247-1>
22. Lisjak, A., Grasselli, G.: A review of discrete modeling techniques for fracturing processes in discontinuous rock masses. *J. Rock Mech. Geotech. Eng.* **6**(4), 301–314 (2014). <https://doi.org/10.1016/j.jrmge.2013.12.007>
23. Luding, S.: Introduction to discrete element methods. *Eur. J. Environ. Civil Eng.* **12**(7–8), 785–826 (2008). <https://doi.org/10.1080/19648189.2008.9693050>
24. O’Sullivan, C., Bray, J.D., Li, S.: A new approach for calculating strain for particulate media. *Int. J. Numer. Anal. Meth. Geomech.* **27**(10), 859–877 (2003). <https://doi.org/10.1002/nag.304>
25. Peacock, D.C.P., Dimmen, V., Rotevatn, A., Sanderson, D.J.: A broader classification of damage zones. *J. Struct. Geol.* **102**, 179–192 (2017). <https://doi.org/10.1016/j.jsg.2017.08.004>
26. Resor, P.G., Pollard, D.D.: Reverse drag revisited: why footwall deformation may be the key to inferring Listric fault geometry. *J. Struct. Geol.* **41**, 98–109 (2012). <https://doi.org/10.1016/j.jsg.2011.10.012>
27. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **20**, 53–65 (1987). [https://doi.org/10.1016/0377-0427\(87\)90125-7](https://doi.org/10.1016/0377-0427(87)90125-7)
28. Vishnevsky, D., et al.: Correlation analysis of statistical facies fault models. *Dokl. Earth Sci.* **473**(2), 477–481 (2017). <https://doi.org/10.1134/s1028334x17040249>
29. Wang, Y., Tonon, F.: Modeling triaxial test on intact rock using discrete element method with membrane boundary. *J. Eng. Mech.* **135**(9), 1029–1037 (2009). [https://doi.org/10.1061/\(ASCE\)EM.1943-7889.0000017](https://doi.org/10.1061/(ASCE)EM.1943-7889.0000017)