



Research paper

A constrained Delaunay discretization method for adaptively meshing highly discontinuous geological media



Yang Wang^a, Guowei Ma^{a,b,*}, Feng Ren^a, Tuo Li^a

^a School of Civil, Environmental and Mining Engineering, The University of Western Australia, Perth, WA, 6009, Australia

^b School of Civil and Transportation Engineering, Hebei University of Technology, Tianjin 300401 China

ARTICLE INFO

Keywords:

Highly discontinuous geological media
Discrete fracture networks
Doubly adaptive meshes
Constrained Delaunay discretization
Persson's algorithm

ABSTRACT

A constrained Delaunay discretization method is developed to generate high-quality doubly adaptive meshes of highly discontinuous geological media. Complex features such as three-dimensional discrete fracture networks (DFNs), tunnels, shafts, slopes, boreholes, water curtains, and drainage systems are taken into account in the mesh generation. The constrained Delaunay triangulation method is used to create adaptive triangular elements on planar fractures. Persson's algorithm (Persson, 2005), based on an analogy between triangular elements and spring networks, is enriched to automatically discretize a planar fracture into mesh points with varying density and smooth-quality gradient. The triangulated planar fractures are treated as planar straight-line graphs (PSLGs) to construct piecewise-linear complex (PLC) for constrained Delaunay tetrahedralization. This guarantees the doubly adaptive characteristic of the resulted mesh: the mesh is adaptive not only along fractures but also in space. The quality of elements is compared with the results from an existing method. It is verified that the present method can generate smoother elements and a better distribution of element aspect ratios. Two numerical simulations are implemented to demonstrate that the present method can be applied to various simulations of complex geological media that contain a large number of discontinuities.

1. Introduction

Discrete fracture networks (DFNs) play an important role in simulations and analysis of fluid flow and transport in discontinuous geological media (Jing, 2003). They have rather complex configurations that are mainly caused by fracture interconnections and variations in fractures' geometries such as orientation, size, and position. The geometrical complexities of DFNs significantly increase the difficulties of keeping geometric consistency in the mesh generation. Developing an effective discretization method is highly demanded for numerical analysis of fluid flow and transport in complex discontinuous media.

1.1. Triangular and tetrahedral meshes in DFNs

Typically, there are two types of mesh typically used in simulations of DFN problems. One is the planar mesh method using triangular elements on fractures only. It has been widely used to simulate hydraulic flow in fractured rocks in which the matrix has a low permeability (De Dreuzy et al., 2012). The planar mesh requires that the elements of a fracture conform to those of other fractures at their intersections to ensure the

continuity of unknowns such as displacement, pressure, and fluid flux across the intersection lines between fractures. Some researchers tried to solve mesh issues in flow simulations by enforcing the continuity of heads and flux at nonmatching grids (Berrone et al., 2014; Pichot et al., 2010). However, more computations are required to implement the calculation at fracture intersections. Conforming elements are still preferred for accurate and efficient simulations using the finite element method and the control volume method. The other is the matrix mesh method, which needs to discretize the discontinuous geological media when the influences of the matrix on the system are not negligible, such as the simulations of coupling thermal-hydro-mechanical systems (Zhuang et al., 2014) and fluid flow in fractured porous media (Vu et al., 2013). The matrix elements are required to conform to the intersections among fractures as well as to recover all planar fractures.

1.2. Developed methods of mesh generation for DFNs

Various approaches have been proposed to generate numerical models for a rock mass with 3-D DFNs. For example, Maryška et al. (2005) developed a method to geometrically simplify the fracture

* Corresponding author. School of Civil, Environmental and Mining Engineering, The University of Western Australia, Perth, WA, 6009, Australia.
E-mail address: guowei.ma@uwa.edu.au (G. Ma).

networks by removing or replacing the close parallel fractures; moving, stretching, and merging intersections in the fracture planes; and then discretizing the modified geometry using an advancing front method. As a consequence of the geometrical simplification, the 3-D geometrical correspondence disappears. Mustapha and Mustapha (2007) proposed another method to reduce the highly complex geometry of fractures by projecting the boundaries of the fractures and the intersections between the fractures on a 3-D regular grid (3DRG). The obtained geometry is made very regular by removing the ill-conditioned elements in the triangulated mesh. However, Erhel et al. (2009) pointed out that each projected node must have only two adjacent edges, and each projected edge must have one or two neighbors based on Mustapha and Mustapha's method (2007), otherwise the mesh generation would fail. Therefore, a local correction to modify the surface of fractures to eliminate these abnormal configurations was introduced. The above methods based on geometrical simplification and 3DRG changed the original geometrical characteristics of fractures, and the generated mesh is not adaptive. Hyman et al. (2014) introduced a feature rejection algorithm for meshing (FRAM) to reject pathological fractures in the realization of DFNs for avoiding undesirable configurations. In their work, adaptive and conforming triangular meshes with good quality were generated for the modified DFNs based on a conforming Delaunay triangulation algorithm (Murphy et al., 2001).

The above studies are restricted in meshes on fractures only. Many researchers also attempted to discretize the fractures and the matrix together. Blesent et al. (2009) developed a modeling method for fractured porous media, in which fine tetrahedra are generated around the possible position of a fracture, and some tetrahedron faces are chosen to approximate the fracture. This method is efficient in approximating an isolated fracture, but it becomes rather difficult in highly discrete fracture networks. Mustapha et al. (2011) proposed a local transformation method to significantly improve the quality of meshes for 3-D DFNs by scanning the elements through a small cube and merging the vertices of involved tetrahedra. The shape of fractures is distorted slightly after the local transformation. It is verified that the distortion of the mesh does not affect the fluid simulation in simple discrete fracture networks, but it may affect the results of some simulations such as fracture propagation and coupling systems. In recent years, based on open-source mesh software, some methods have been developed to generate tetrahedral meshes on more complex geological structures (Cacace and Blocher, 2015; Zehner et al., 2015).

A constrained Delaunay discretization method for adaptively meshing highly discontinuous geological media is developed in the present study. This method adopts Persson's algorithm (Persson and Strang, 2004; Persson, 2005) and an explicit size function to adaptively distribute mesh points on fractures. The size function based on a distance function ensures that the mesh is refined in the important and sensitive parts such as fracture intersections and gradually coarse away from them. Then, a constrained Delaunay triangulation (Chew, 1989) and constrained Delaunay tetrahedralization (Si, 2015) are introduced to construct the conforming connectivity of elements. As a result, the triangular elements and the tetrahedral elements can exactly conform to the fracture intersections and the pre-positioned fractures, respectively. In addition, the doubly adaptive nature of the tetrahedral elements reduces the number of elements significantly and improves efficiency in numerical simulations. During the entire process, there is no local and global geometrical transformation of DFNs for keeping the original features of DFNs. The proposed method is applicable to a rock mass medium featured with tunnels, shafts, slopes, boreholes, water curtains and drainage systems, etc.

2. Constrained Delaunay triangulation for DFNs

There are typically two ways to reconstruct the geological model of fractured rocks with a complex configuration of highly discrete fracture networks (DFNs), namely deterministic description and statistical

realization. The deterministic description is used for the large-scale fractures from the field surveys (Tavakkoli et al., 2009). Because of the limitations of cost and survey techniques, not all fractures are necessary or could be surveyed individually. Therefore, small-scale fractures under seismic resolutions are generated stochastically based on some probability distribution functions (PDFs) relating to the properties of DFNs (Xu and Dowd, 2010). In this study, 3-D discrete fracture networks are generated stochastically to represent the discontinuities in a general form.

2.1. Projecting each fracture on a 2-D plane

Fractures in three-dimensional DFNs are usually equivalent to convex polygons. But if the artificial structures are involved, such as caverns or tunnels as shown in Fig. 1(a), they can be possibly cut into any shapes. For simplifying the triangulation of fractures, the mesh process is implemented on each planar fracture individually rather than meshing them all in space. During mesh generation, the intersecting segments and points on each fracture are pre-positioned without translation during the convergence iterations.

Taking Fig. 1(a) as an example, two rock caverns and two water curtain tunnels are intersected with fractures and boreholes. The geometry of a fracture shown in Fig. 1(b) becomes very irregular because of the intersection with a cavern (Ω^1), a tunnel (Ω^2), two fractures (Γ^1 , Γ^2 , and Γ^3), and boreholes (Γ^4 and Γ^5). The cavern and tunnel boundaries and intersections between fractures create discontinuous segments, and the vertices of the polygons, endpoints of the fracture intersections, and the crossover points of the boreholes create discontinuous points. Both discontinuous segments and points are regarded as pre-positioned features during the adaptive-mesh generation process.

2.2. Implicit geometry of the polygonal fracture

The complex geometry of objects can be described implicitly by using signed distance functions and Boolean operations in the mesh process (Persson and Strang, 2004; Persson, 2005). A signed distance function is an implicit function to determine whether or not a point lies outside of a specified domain. They also return the shortest distance from a point to the closest boundary. In this study, each fracture is described as a planar straight-line graph (PSLG), which includes polygons, polygons with holes, complexes, segments, and isolated vertices. The intact-fracture polygon, holes, and segments/vertices are represented by Ω^i , Ω^j , and Γ^j , respectively. The signed distance function of a domain can be written as

$$d_{\Omega^i}(p) = \pm \min p - p_{\partial\Omega^i}, \quad (1)$$

where $d_{\Omega^i}(p)$ is the signed distance from a point p to the boundary of the domain Ω^i . Positive and negative signs mean that the point is outside and inside the domain, respectively. $p_{\partial\Omega^i}$ is the point on the boundary of the domain Ω^i . The general implicit description of a polygonal fracture, including some holes, can be represented using the Boolean operation

$$d_{\Omega}(p) = \max(d_{\Omega^0}(p), -d_{\Omega^1}(p), \dots, -d_{\Omega^r}(p)). \quad (2)$$

2.3. Local refinement

In a wide range of numerical simulations, an adaptive mesh is required to adapt the accuracy of solutions around the important and sensitive parts in the domain and save computing time. In the system of DFNs, the intersections among fractures and intersections between fractures and other structures are usually regarded as sensitive parts in which a fine mesh is needed. Many mesh refinement methods have been developed to generate adaptive meshes, based on implicit- (Ruppert, 1995; Shewchuk, 2002; Erten and Üngör, 2007; Üngör, 2009) or explicit-size function (Peraire et al., 1987; Blacker and Stephenson, 1991; Persson

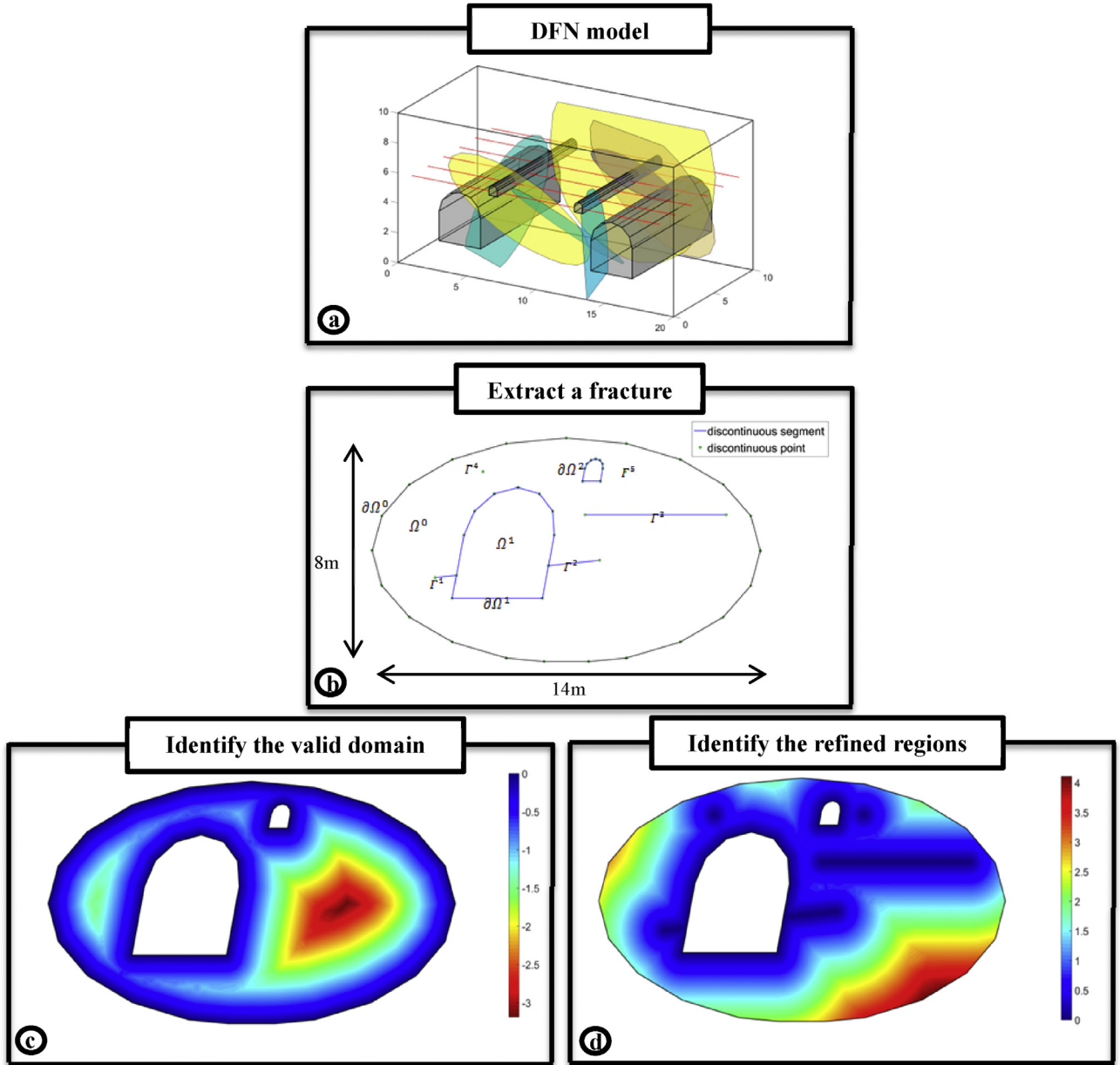


Fig. 1. (a) A DFN model in an artificial underground structure; (b) Planar straight line graph (PLSG) of an extracted fracture; (c) Signed distance function: the boundary of the valid domain is given by the zero level set of $d_\Omega(p)$; (d) Unsigned distance function: the sensitive parts are given by the zero level sets of $d_r(p)$.

and Strang, 2004; Persson, 2005; Li et al., 2014). The explicit method is where it is easier to control the element size and size grading in different regions than the implicit method. In the present study, an explicit function of element size is implemented to control the element size and size grading in different regions,

$$h(p) = \min(l_{\min} + g \cdot d_r(p), l_{\max}), \quad (3)$$

where $h(p)$ is the desired value of the element size at point p , l_{\min} is the lower limit of element size, l_{\max} is the upper limit of element size, g is the grading of the element size, and $d_r(p)$ is the unsigned distance from point p to the closest sensitive part. The unsigned distance function for a segment or a vertex Γ can be written as

$$d_{\Gamma^i}(p) = \min_p - p_{\Gamma^i}. \quad (4)$$

For the case of Fig. 1(b), if mesh refinement is required around the

discontinuous segments and points, $d_r(p)$ for arbitrary point p within the valid domain is

$$d_r(p) = \min(d_{\Gamma^1}(p), \dots, d_{\Gamma^5}(p), d_{\Omega^1}(p), d_{\Omega^2}(p)). \quad (5)$$

Fig. 1(c) and (d) illustrate the signed distance function $d_\Omega(p)$ to define the valid domain and the unsigned distance function $d_r(p)$ from Eq. (5) to define the background of the adaptive mesh over the valid domain. The desired value of element size at each point is directly proportional to $d_r(p)$ and is also dominated by l_{\min} and l_{\max} . As shown in Fig. 2, if l_{\max} is too large to work, the element size distribution will be directly proportional to Eq. (5) so that they have the same trend as the distance function $d_r(p)$. Otherwise, l_{\max} will limit the mesh size in the far field of the sensitive regions. Note that the desired element length l_0 between two points takes the average of $h(p)$,

$$l_0(p_1, p_2) = (h(p_1) + h(p_2))/2. \quad (6)$$

2.4. Division of the pre-positioned segments

According to Eq. (3) and Eq. (6), the desired lengths of pre-positioned segments are l_{\min} because $d_r(p)$ is zero. Therefore, segments that are longer than l_{\min} have to be divided into smaller elements to satisfy the element size function. However, in multi-intersected fractures, the intersection segments on a fracture may intersect with other intersection segments. The intersection points between intersection segments should be shared by multiple fractures. If these segments are broken into desired elements straightly, these intersecting points may be missed, and the mesh would not conform to these points. Therefore, the tangling segments in the model must be banished. The segments will be divided into subsegments with regard to the intersection points. Afterwards, each subsegment is equally partitioned into N_f elements,

$$N_f = \frac{L_i}{l_{\min}} \geq 1, \quad (7)$$

where N_f is the number of the elements on the subsegment, and L_i is the length of the subsegment. The subsegment is not discretized further if the length L_i is smaller than the lower limit of element size.

2.5. Scattering mesh points

The algorithm of spreading mesh points is based on the analogy between triangular elements and spring networks (Persson, 2005). Like the particle models, the forces to drive a node are assumed to be along the springs between such node and the immediate neighbor points. The topology of the lattice spring structure is created by the Delaunay triangulation (Edelsbrunner, 2001). The force vector of each node is

$$\mathbf{F}(\mathbf{p}) = \begin{cases} 0 & p \text{ on } \Gamma \\ \mathbf{F}_{\text{spring}}(\mathbf{p}) + \mathbf{F}_{\text{ext}} & \text{otherwise} \end{cases} \quad (8)$$

Forces applied to fixed points such as those on discontinuous segments are always zero. For other flexible points, the driving force contains the internal force $\mathbf{F}_{\text{spring}}(\mathbf{p})$ from springs and external forces \mathbf{F}_{ext} applied normally to the boundary. The external forces just pull outside points back to the boundary and constrain them to move along the boundary only. The internal force is assumed to be a linear function, which is analogous to ordinary linear springs as argued by Persson (2005):

$$f_i(l_i, l_{0,i}) = \begin{cases} k_i(l_{0,i} - l_i) & \text{if } l_i < l_{0,i} \\ 0 & \text{if } l_i \geq l_{0,i} \end{cases} \quad (9)$$

where $f_i(l_i, l_{0,i})$ is the internal force in the i -th spring, l_i is the actual length of the i -th element, $l_{0,i}$ is the desired length, and k_i is the stiffness of the i -th pseudo-spring. To drive the nodes moving to the equilibrium positions efficiently, the pseudo-springs are changed to be nonlinear:

$$k_i = 1.2 \times \frac{1}{l_{0,i} - l_i} \left(l_{0,i} \times \frac{\sum l_{0,i}^2}{\sum l_i^2} - l_i \right). \quad (10)$$

Eq. (9) indicates that the force between mesh points is only repulsive. Therefore, the gaps between mesh points will only increase or remain unchanged during scattering mesh points. To ensure the meshes are refined in specified regions, mesh points initially are distributed uniformly with a gap of the lower limit of mesh size. Then, a rejection strategy (Persson, 2005) is used to reject the redundant points.

The equilibrium position of mesh points is difficult to be solved straightly from the system $\mathbf{F}(\mathbf{p}) = 0$ because of the inconsistency in the topology and the external force. By introducing an artificial time (Persson, 2005), the position of points is assumed to be time-dependent and its first derivative with respect to the time is

$$\frac{\partial \mathbf{p}(t)}{\partial t} = \mathbf{F}(\mathbf{p}), \quad t \geq 0. \quad (11)$$

This ordinary differential equation (ODE) can be solved using a forward Euler method,

$$\mathbf{p}(t_n) = \mathbf{p}(t_{n-1}) + \Delta t \Delta \mathbf{F}(\mathbf{p}(t_{n-1})), \quad t_n = n \Delta t \quad n = 1, 2, 3, \dots, \quad (12)$$

where Δt is the artificial time step, and n is the iteration step. The iteration terminates when the position of points does not change further or when the iteration reaches the maximum value.

2.6. Constrained Delaunay triangulation

After the equilibrium position of mesh points is confirmed, the final topology must contain all preexisting points and segments. The topology is created by the Delaunay triangulation (DT) during scattering mesh points. The Delaunay triangulation can give well-shaped triangles by using an empty-circle criterion to maximize the internal minimum angles (Edelsbrunner, 2001), as shown in Fig. 3(a). However, the Delaunay triangulation cannot guarantee that the preexisting segments are preserved. To recover all preexisting segments in the model, the constrained Delaunay triangulation (CDT) (Chew, 1989) is adopted, which can keep the properties of Delaunay triangles as many as possible but relax the empty-circle criterion around restrained edges. This relaxation rule allows the circumcircles of triangles containing the preexisting edge to include other points as shown in Fig. 3(b), but it would sacrifice the mesh quality to some extent.

Fig. 4 displays the final distribution of mesh points and the corresponding mesh generated with the constrained Delaunay triangulation. For a discrete fracture network model, each polygonal fracture is meshed individually in the same way, and then they are assembled. The mesh result for the example in Fig. 1(a) is shown in Fig. 5. It also highlights that triangle edges and vertices from different fractures coincide at the

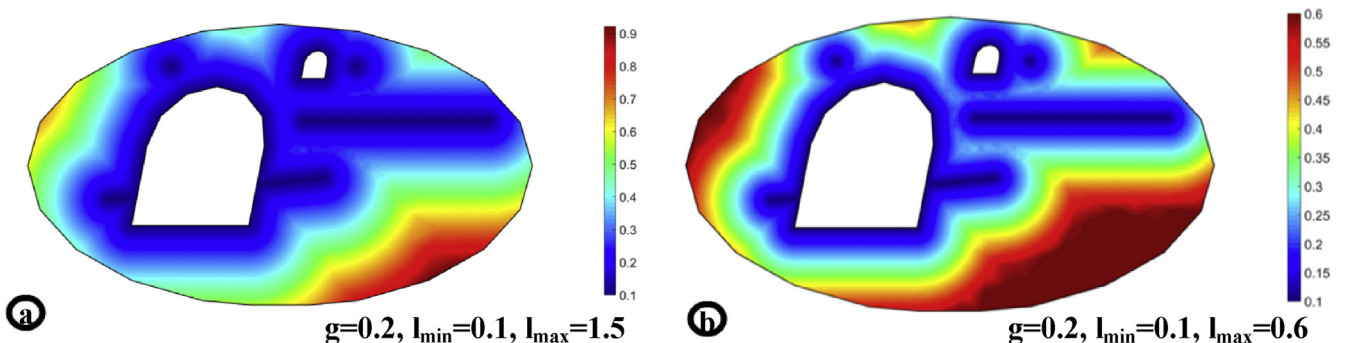


Fig. 2. Distributions of element size from the function of element size: (a) l_{\max} does not work: $\max(h(p)) < l_{\max}$; (b) l_{\max} works: $\max(h(p)) = l_{\max}$.

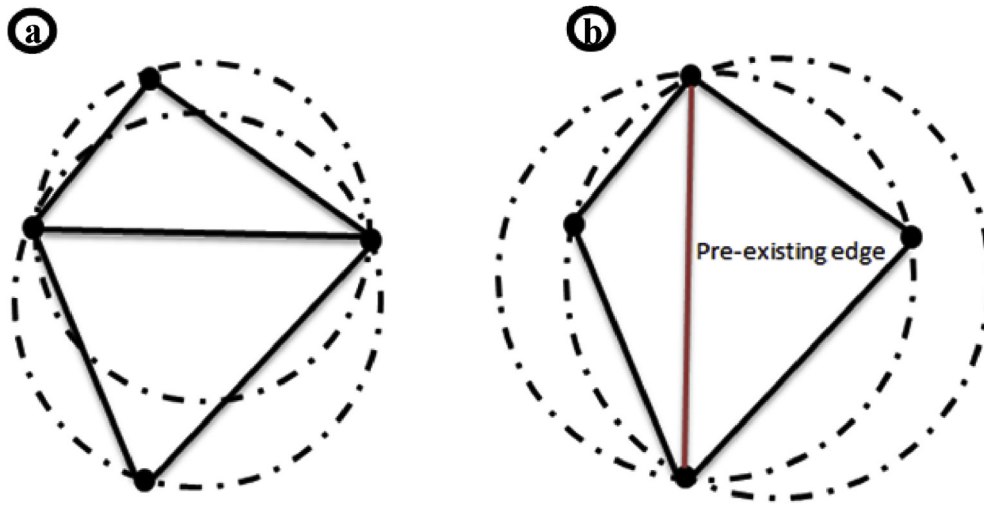


Fig. 3. (a) Circle empty criterion in Delaunay triangulation (DT), (b) The circle empty criterion is relaxed in triangles around the pre-existing edge in constrained Delaunay triangulation (CDT).

intersections so that there is no tangling element in the mesh. The capacity of the proposed method to mesh large-scale and complex DFNs and to build models for practical engineering projects will be demonstrated in section 5.

3. Constrained Delaunay tetrahedralization for fractured porous media

Besides the discrete fracture networks, the discretization of the rock matrix is also necessary in many numerical simulations, such as the multiphase flow in fractured porous medium, the interaction between fractures and rock matrix, coupling hydro-thermo-mechanical modeling in fractured rocks, and others. A major challenge to the mesh generation of 3-D fractured porous media is the recovery of the fracture polygons. Si (2015) proposed an algorithm of facet recovery to construct constrained Delaunay tetrahedralization (CDT) for 3-D polyhedral domains and developed an effective and efficient software—TetGen. In this study, TetGen is adopted to carry out the tetrahedralization of the triangulated DFNs.

3.1. Building PLC from triangulated DFN

TetGen (Si, 2015) uses a general input geometry called piecewise linear complex (PLC), which includes a set of facets, segments, and isolated vertices. By definition, any two parts in a PLC must be completely separated or intersected only at shared facets, segments, or vertices. And any points set which define a facet must be coplanar (Verbree, 2006). Each facet in a PLC must be closed and is represented by a planar straight-line graph (PSLG), which may also contain some segments, vertices, and holes. Similar to PLC, tangling segments are not allowed,

and the points set must be coplanar in each PSLG. If each fracture is regarded a PSLG in the input PLC, the boundary and the massive intersections have to be handled carefully so the program does not crash. With the increase of fractures, the geometrical complexity of the DFN becomes higher so that the PSLGs of fractures are more complicated.

Adopting the triangular elements of fractures as shown in Fig. 6(a) to construct the PSLGs over fracture polygons will simplify the construction of the input PLC. The input geometry is classified into two parts as shown in Fig. 6(b), that is, internal facets and boundary facets. The internal facets are many triangular elements from the planar mesh generation. As the vertices of any triangle must be coplanar and every triangular facet on fractures is nontangled, the problems of tangling segments and coplanar points are eliminated. As some fractures may intersect with boundary facets, the PSLGs of the boundary facets must contain these intersections and the mesh points on the boundary.

3.2. Constrained Delaunay tetrahedralization

Once the PLC of the fractured media is built properly, constrained Delaunay tetrahedralization of the PLC is implemented totally relying on TetGen (Si, 2015). Similar to constrained Delaunay triangulation, constrained Delaunay tetrahedralization relaxes the empty-sphere rule around the prespecified constraints such as points, edges, and faces. Therefore, the circumspheres of some tetrahedra are allowed to enclose other vertices in the process of tetrahedralization to preserve the constraints. TetGen tetrahedralizes the domain by inserting Steiner points and refines the mesh using a new constrained Delaunay refinement method. The user can specify the quality parameters to control the quality, grading, and size of the tetrahedra. In this study, version 1.5 of TetGen¹ is used. Only a maximum allowable radius-edge ratio and a

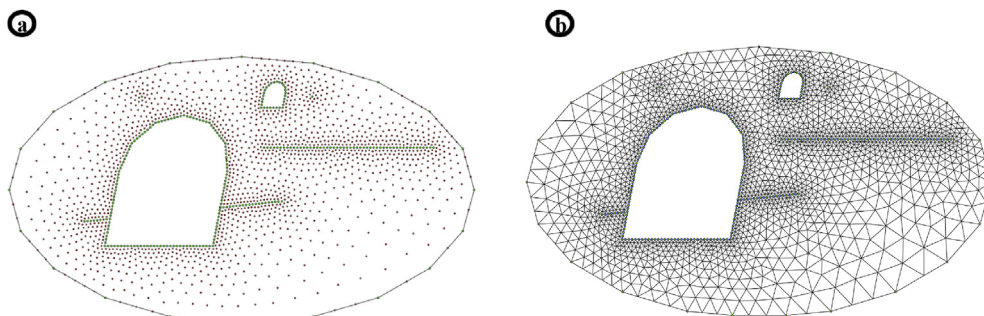


Fig. 4. (a) Final mesh points; (b) Connecting the mesh points based on Constrained Delaunay Triangulation (CDT).

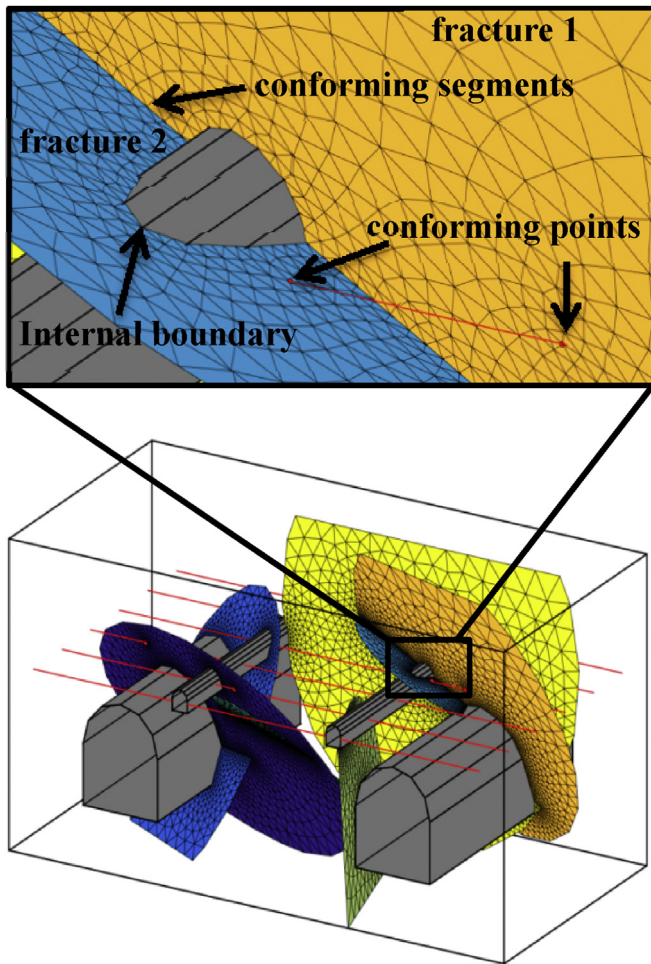


Fig. 5. Final planar mesh for the discrete fracture network in an artificial underground structure.

maximum volume constraint on all tetrahedra are specified in the command line for controlling the density and size of elements. For example, the command `-pq1.4a0.01` indicates that the PLC is tetrahedralized with the maximum radius-edge ratio of 1.4 and the maximum tetrahedral volume of 0.01 m^3 .

The mesh points from the planar mesh are dense around the sensitive parts such as intersections and become coarse away from them as shown in Fig. 6(c). The region with more initial points will be inserted with more Steiner points incrementally for locally optimizing the quality of elements during the tetrahedralizing of the domain (Si and Shewchuk, 2014). Therefore, the tetrahedral elements will be adaptive automatically and conform to the trend of size variation of the planar mesh along the fracture polygons and in space. This features the mesh with the doubly adaptive capacity—along fractures and in space. For cases where the incidence angles between fractures are very small or fractures are very close, even if there are only sparse mesh points on the fractures, the initial input points in that region are still many in space. As a result, unexpected fine elements will be generated there. The quarter cutoff graph in Fig. 6(c) demonstrates that the vertical fracture is recovered, and the elements are conforming and doubly adaptive.

4. Performance of the method

4.1. Triangle quality

The quality of the elements is of great importance to the accuracy and efficiency in computational analysis. Unacceptable quality of elements in

the numerical models will result in poor performance of numerical simulations. The element quality parameter is an important indicator to decide whether the element is acceptable or not. Many studies have been conducted to investigate the measures to evaluate the element quality (e.g., Parthasarathy et al., 1994; Field, 2000). In this work, the quality of elements in the planar mesh of DFN models is compared with that in Hyman et al. (2014), and the aspect ratio and edge ratio of triangles are

$$q_a = ((l_2 + l_3 - l_1)(l_1 + l_3 - l_2)(l_1 + l_2 - l_3))/l_1 l_2 l_3, \quad (13)$$

$$q_e = l_{\min}/l_{\max}, \quad (14)$$

where l_1, l_2, l_3 are the length of triangle edges, q_a is the aspect ratio, l_{\min} is the minimum edge length, and l_{\max} is the maximum edge length. Equilateral triangles have $q_a = 1$ and $q_e = 1$, and degenerated triangles have $q_a = 0$ and $q_e = 1$.

Both Hyman et al.'s method and the present method can generate the adaptive triangular mesh, and the mesh can gradually be coarsened away from the intersections. Fig. 7 displays the mesh of an elliptical fracture with multiple parallel intersecting segments. Both methods generated adaptive and conforming meshes. In Hyman et al.'s result, the mesh is refined around intersecting segments and rapidly becomes coarse in the far field of intersecting segments. By comparison, the elements produced from the present method have a gradual and smooth variation in size. A quantitative comparison about mesh quality based on Eq. (13) will be discussed below.

Hyman et al. (2014) investigated the statistics of triangle quality from 10 independent DFN realizations. Each realization included two fracture sets and a total of 100 elliptic fractures in a 10-m cubic domain. In each sample, the pathological fractures were removed to decrease the difficulty of meshing and improve the quality of elements. After mesh generation, the algorithm provided by Murphy and Gable (1998) was used to modify the boundary elements with the incident angle on the boundary greater than 90° . As the fractures are generated stochastically, it is almost impossible to reproduce the exact same DFNs as Hyman et al.'s. Therefore, in our samples, the same number of realizations and fractures in a cube with a 10-m length are generated as well. But the size, position, and orientation of each fracture is produced based on a logarithmic distribution, a uniform distribution, and a Fisher distribution, respectively (Xu and Dowd, 2010). In addition, no fracture is modified or removed during sampling. Almost the same number of elements is generated to compare with the results from Hyman et al. (2014).

By studying the statistics data in Fig. 8, all the means and medians of aspect ratio, edge ratio, minimum angle, and maximum angle of triangles are improved significantly by the proposed method. The mean and median aspect ratios are 0.9547 and 0.9731, respectively. And it shows that most of the triangles are nearly equilateral triangles. Moreover, the decrease of coefficients of variation in all quality parameters indicates that the distributions of the quality parameters have lower variances. According to the histograms, it is obvious that the histograms of all parameters in our results are sharper and narrower than Hyman et al.'s results. More triangles have the quality values around the average because of lower variances. Of the triangles, 87.83% have an aspect ratio greater than 0.9, while 89.74% have an edge ratio greater than 0.7. Both the means of minimum angle distribution and maximum angle distribution lie at almost 60° , which is the ideal angle of an equilateral triangle.

4.2. Tetrahedron quality

Numerous methods have been proposed to evaluate the quality of tetrahedral elements (Bern and Plassmann, 1999) with their own merits and drawbacks (Parthasarathy et al., 1994). One formula for a normalized aspect ratio is used to evaluate the tetrahedron,

$$q = 2\sqrt{6}R_{\min}/S_{\max}, \quad (15)$$

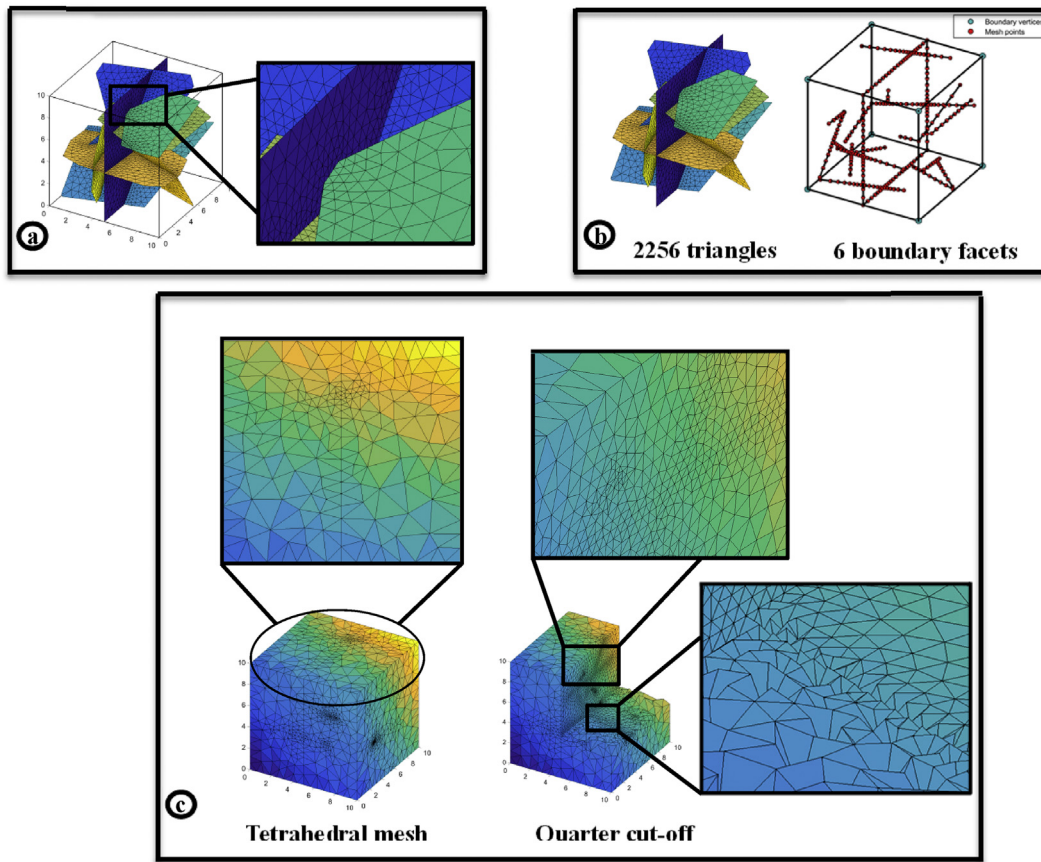


Fig. 6. Illustration of the steps to tetrahedralize a DFN model: (a) Triangulation of DFN; (b) Building the PLC; (c) Constrained Delaunay tetrahedralization using TetGen (Si, 2015).

where R_{in} is the radius of the inscribed sphere and S_{max} is the length of the longest edge. The q value changes from 0 to 1. A tetrahedron without volume is 0, and a regular tetrahedron is 1. In addition, the dihedral angles between the faces in each tetrahedron are also used as the shape measure of the mesh. The angle varies from 0 to 180° . All dihedral angles in a regular tetrahedron are 70.53° .

It is difficult to find the element quality investigation for the tetrahedralization of DFN models from current literature. Therefore, six cases are given to investigate the quality of tetrahedral elements in this section. The first case is a cubic continuum, and the others are cubic blocks with

different DFNs. They are all tetrahedralized according to the same input parameters. All the input parameters in these cases are summarized in [Supplementary Table 1](#). From cases 1 to 4, fractures are generated by the same probabilistic distributions with the same parameters, but the numbers of fractures are 50, 100, 150, and 200 respectively. The Fisher constant k is very small for making the orientation of fractures more random in space. In contrast, case 5 has well-orientated fractures by using a larger Fisher constant. The maximum radius-edge ratio of 1.3 and the maximum element volume of 1 m^3 are applied in all cases.

[Table 1](#) lists the statistics of the numbers of intersections and

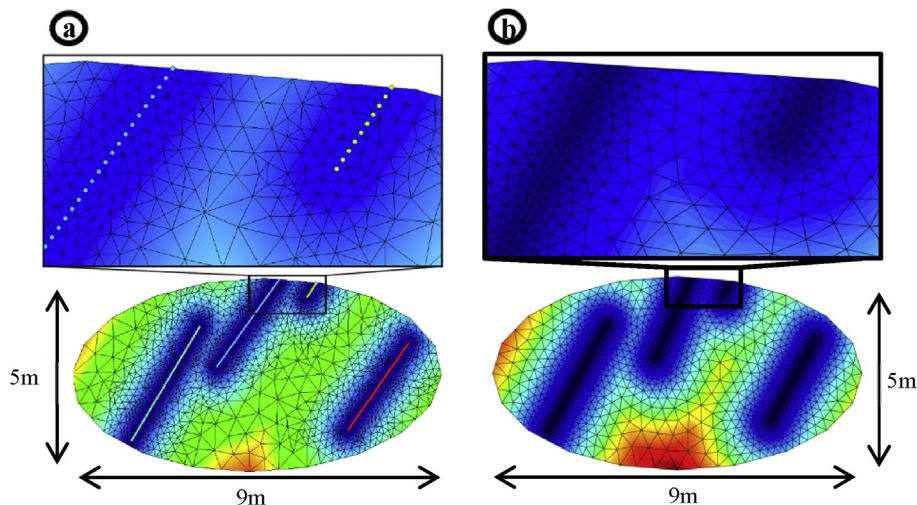


Fig. 7. Triangular mesh of an elliptical fracture with multiple parallel intersection segments: (a) Hyman et al.'s result [12]; (b) The pr.

	Aspect Ratio	Edge Ratio	Min. angle	Max. angle
Mean	0.8820	0.7213	44.15°	79.11°
Median	0.9132	0.7211	44.58°	76.68°
C_v	0.1195	0.1678	0.1492	0.1759

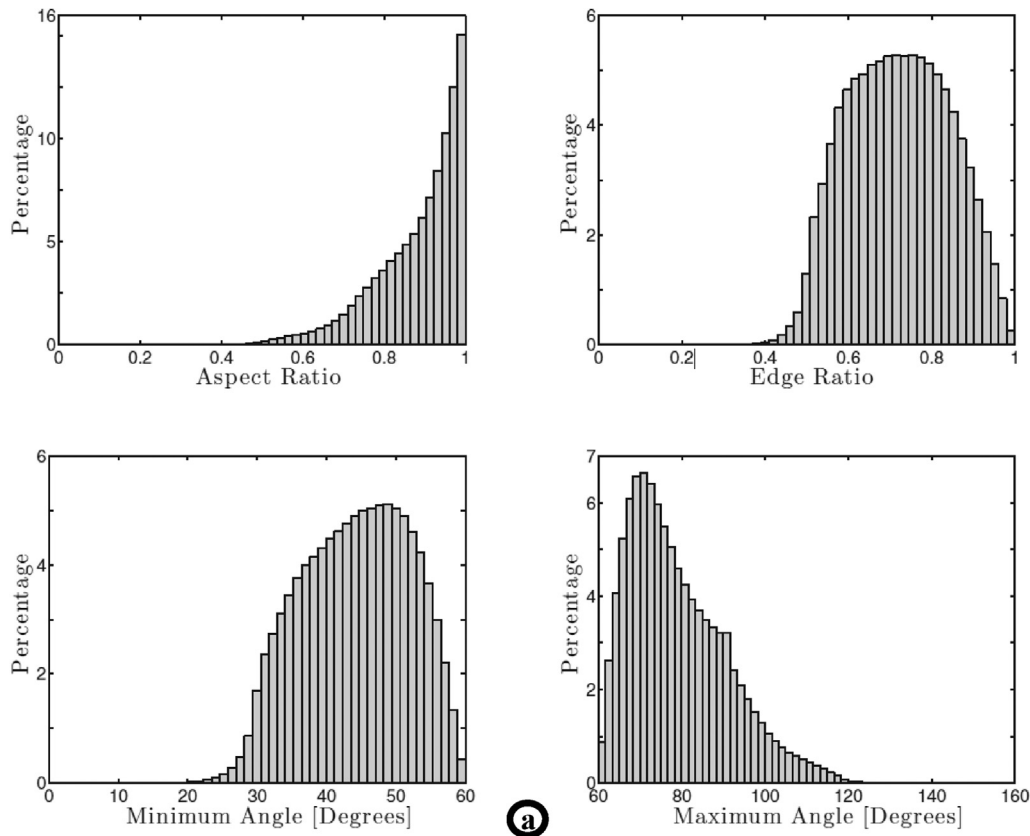


Fig. 8. Statistics and histograms of the aspect ratios, the edge ratios, the minimum angles and the maximum angles: (a) Hyman et al.'s result [12]: 5000355 triangles; (b) The present method: 4926725 triangles.

triangles, mean aspect ratio, mean dihedral angles, mean minimum, and maximum dihedral angles of tetrahedra for all cases. By comparing case 1 to 5 with the continuum, it is demonstrated that the insertion of fractures influences the mesh quality. The average aspect ratio and the mean minimum dihedral angle of the continuum are slightly better than other cases. But the mean maximum dihedral angles of the tetrahedra are almost unchanged. Overall, the quality of tetrahedral elements is good, and the small sacrifice of the average quality is also acceptable in the numerical simulations. The existence of fractures dramatically decreases the maximum and minimum volumes of the elements. As a result of smaller elements, the number of elements increases rapidly. It is also observed that the element quality decreases very slowly with the increase of fractures. But in case 4 and 5, although they have the same number of fractures, case 5 has slightly worse quality. An interesting phenomenon is found in the number of elements shown in Table 1. Case 5 has fewer points in the PLC but more output elements than case 4. It happens because some unexpected fine tetrahedra are generated around the subparallel fractures.

4.3. CPU time-cost

In this section, the computational times of the six examples in section 4.2 are presented. The techniques including the geometric treatment,

triangulation, and building PLC had been written into the Matlab environment. And the open-source software TetGen (Si, 2015) was developed in C++. All tests were performed on a computer having Windows 7 64-bit system, Intel Core i7-4770 CPU (3.4 GHz), and 8 GB RAM. These six runs were carried out with the same parameters as listed in Supplementary Table 2. The computing time in each step has been listed in Table 2, and the more details of the geometry and elements can be checked in Table 1.

TetGen took only 0.015 s to create a CDT for the continuum. However, three extra steps are required for DFN models before running TetGen. The first step (geometric treatment) is to truncate the parts of the fractures that are outside the domain and calculate the intersections among fractures. The time-cost in geometric treatment is directly proportional to the number of fractures. However, the running time of the second step (triangulation) not only depends on the number of fractures but also on the complexity of the fracture network. As shown in Tables 1 and 2, although case 5 has the same number of fractures as case 4, the larger number of intersections in case 4 took more time in the triangulation. The third step is to build the input PLC using the generated triangles. Therefore, it spends more time with more generated triangles. Moreover, the computing time of TetGen increases dramatically when the fractures are inserted into the continuum. It is found that the complexity of the input PLC and the distribution of fractures eventually

	Aspect Ratio	Edge Ratio	Min. angle	Max. angle
Mean	0.9547	0.8253	50.66°	70.97°
Median	0.9731	0.8348	51.44°	69.50°
C _v	0.0573	0.1121	0.1074	0.1101

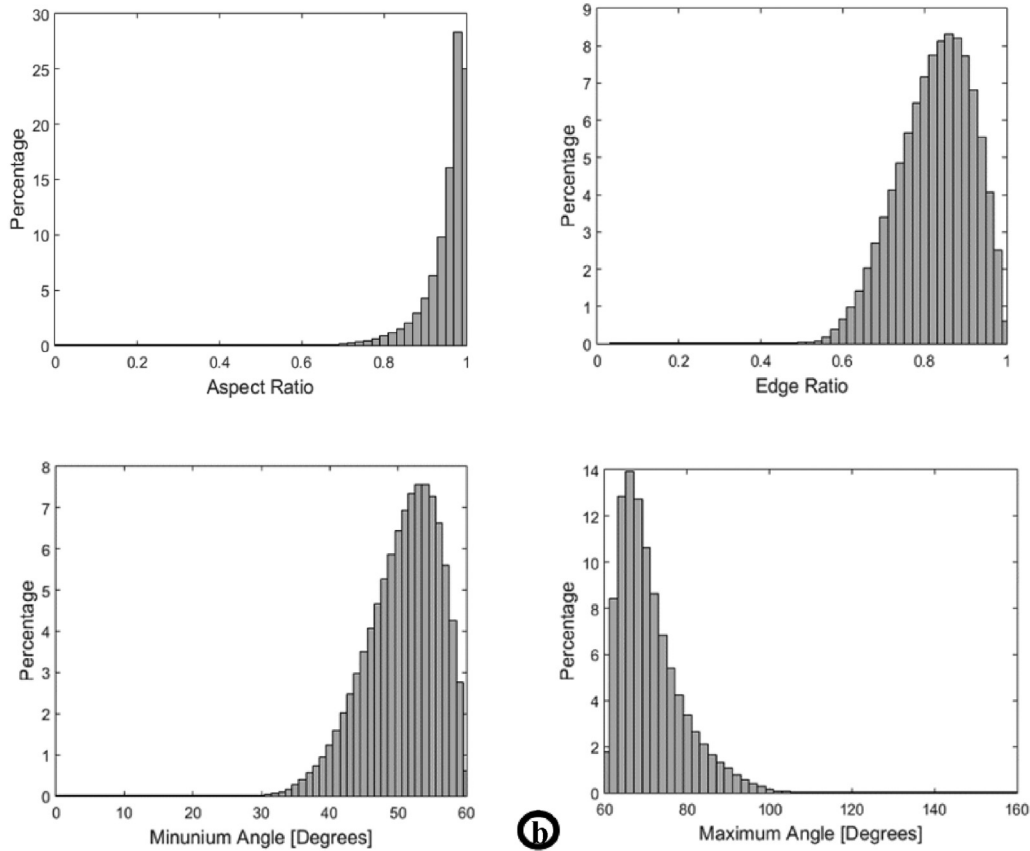


Fig. 8 (continued).

Table 1

Summary of the geometry information, the triangulated fractures, the number of elements and the quality of tetrahedra in the six cases.

	Continuum	Case1	Case2	Case3	Case4	Case5
Fractures	0	50	100	150	200	200
Intersections	0	97	376	869	1405	884
Triangles	0	18385	52459	97652	139167	112055
Points in PLC	8	9382	25306	44824	62138	52885
Tetrahedra	3341	270060	805369	1632608	2700536	3257981
Mean aspect ratio of tetrahedra	0.6267	0.6135	0.6121	0.6095	0.6076	0.5998
Mean maximum dihedral angle [Degrees]	133.1273	133.0498	133.0915	133.2165	133.3079	133.7582
Mean minimum dihedral angle [Degrees]	47.5381	46.6458	46.4678	46.2355	46.1539	45.6704
Largest volume(m ³)	0.99791	0.46721	0.15646	0.16846	0.080103	0.09216
Smallest volume(m ³)	0.022904	1.2182×10^{-10}	1.9863×10^{-12}	2.9753×10^{-14}	1.112×10^{-13}	1.8222×10^{-15}

Table 2

Summary of the CPU times of different meshing stages in the six cases.

CPU time (seconds)						
Meshing Steps	Continuum	Case1 (50)	Case2 (100)	Case3 (150)	Case4 (200)	Case5 (200)
Geometric treatment	0	2.296	8.986	22.297	34.931	34.630
Triangulation	0	17.271	47.825	113.685	136.834	108.476
Building PLC	0	1.127	3.026	5.907	10.462	6.513
Tetgen (tetrahedralization)	0.015	1.544	7.897	27.754	63.628	87.312

affect the time-cost in TetGen.

5. Demonstration

5.1. Complex models

The capacity of the method to discretize highly discontinuous geological media will be presented in this section using some large-scale DFN examples with a large number of fractures. In addition, the model of an underground petroleum storage system is built to present its potential on the simulation of large-scale engineering problems.

5.1.1. Highly discrete fractures models

There are six examples displayed in this section. [Supplementary Table 2](#) summarizes the basic information about these examples. Example 1 has 200 uniformly distributed elliptical fractures. A smaller Fisher constant ($k = 1$) makes them orientate divergently. In example 2, two sets of hexagonal fractures are generated with confined orientation because the larger Fisher constant ($k = 20$). The phenomenon of fracture clustering is given in example 3. By confining the positions of fractures, two clusters of fractures are created. [Fig. 9](#) shows the contoured stereonets and the rose diagrams of the first three examples. The corresponding DFNs and the meshes are displayed in [Fig. 10](#).

The shape of a domain will not always be a cuboid in geological problems. More flexible modeling of the rock mass is necessary in many simulations, such as slope stability analysis and underground seepage simulations. Example 4 is an open-pit slope model with three sets of joints. The orientations and distribution of the three sets of joints are shown in [Fig. 11\(a\)](#). The grey-patched blocks in [Fig. 11\(b\)](#) are used to cut the original block sequentially to form the desired shape of the model. [Fig. 11\(c\)](#) displays the mesh result and the traces of the fractures on the slope surface.

[Fig. 12](#) shows the tetrahedral mesh of the rest two examples. Example 5 is a large-scale model with 2000 hexagonal fractures. These fractures are relatively small and distributed sparsely in the domain. In other words, each fracture has a small probability to intersect with others so that the degree of the geometrical complexity of the fracture network is not high. In contrast, example 6 has relatively complicated internal configuration of fractures. Three sets of large elliptical fractures, a total of 450, are confined to a small space so that each of them is more likely to intersect with others.

5.1.2. Underground petroleum storage system

The petroleum storage example is assumed to consist of two access tunnels, nine main storage caverns, five water curtain tunnels, and many water injection boreholes as shown in [Fig. 13\(a\)](#). The width, height, and length of each storage cavern from north to south are 20 m, 30 m, and 450–600 m, respectively. The access tunnels and water curtain tunnels are 9 m wide and 8 m tall. The water curtain boreholes are orientated horizontally and spaced at 50 m. The 200 in-situ fractures in the domain are generated stochastically based on field survey results and assembled into the geometrical model as shown in [Fig. 13\(b\)](#). The adaptive and conforming mesh is displayed in [Fig. 13\(c\)](#). Moreover, slicing is carried out at the middle position in south-north and east-west orientations. Detailed information about the fractures on each slice is illustrated in [Fig. 13\(d\)](#). These slices can be used to statistically analyze the fractures or simulate the 2-D cases simplified from a 3-D model.

5.2. Numerical simulations

Two application examples of the simulations of water flow are presented for demonstrating the utility of the developed mesh method. The first one is the simulation of water flow in a fractured rock with 1000 random fractures and two tunnels. The matrix is assumed to be

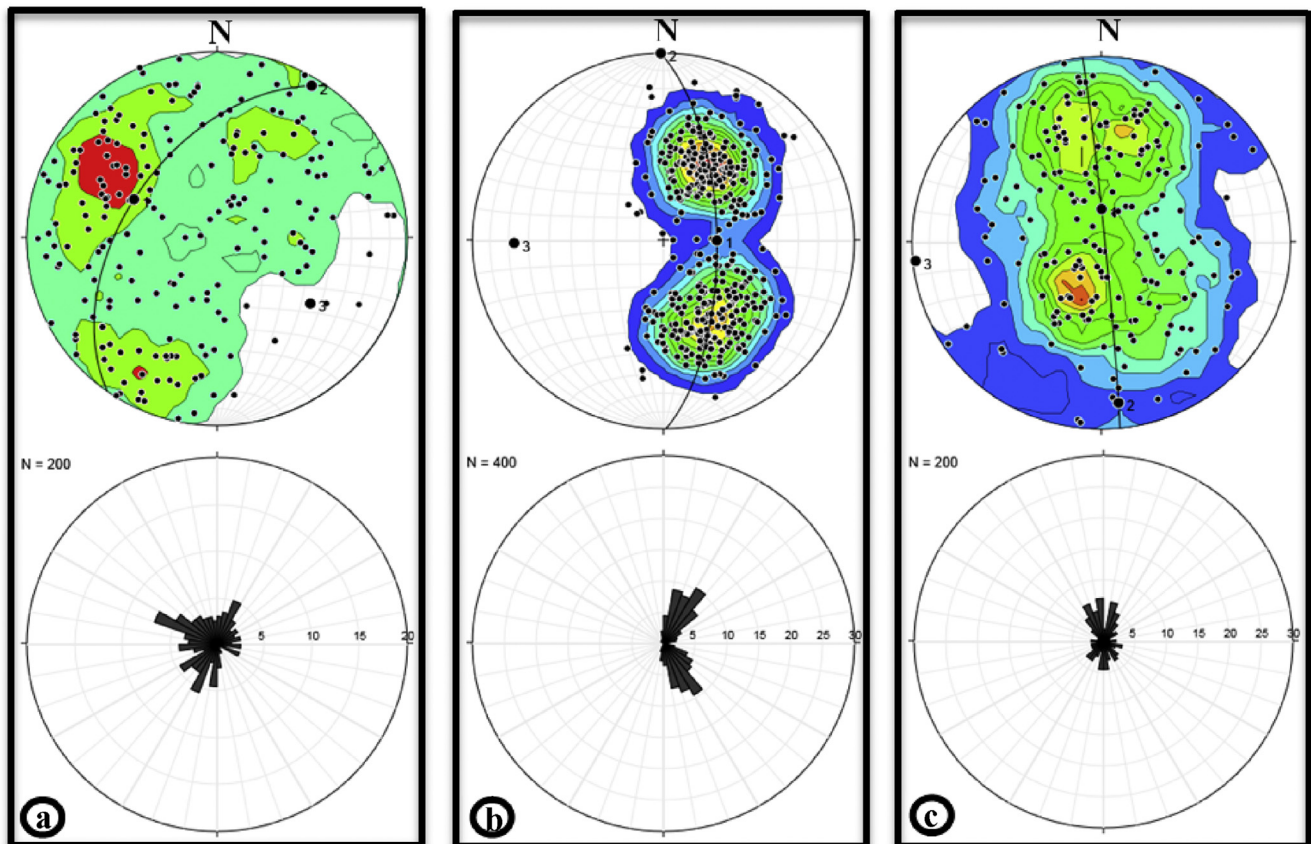


Fig. 9. Contoured stereonets and rose diagrams of fracture poles: (a) Random fractures; (b) Two sets of fractures; (c) Two clusters of fractures.

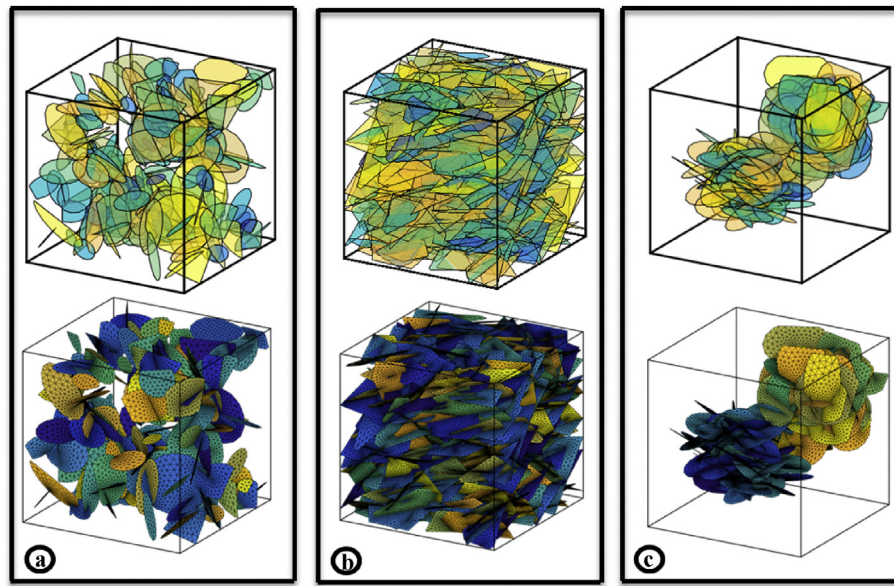


Fig. 10. Illustration and planar mesh of different DFNs in cubes with 50 m sides: (a) Random fractures (200 fractures): 65466 triangles; (b) Two sets of fractures (400 fractures): 510260 triangles; (c) Two clusters of fractures (200 fractures): 280017 triangles.

impervious, and water flows through fractures only. The steady-state free surface and the pore pressure distribution are simulated after two tunnels are excavated. The saturated-unsaturated water flow in the numerical model is governed by the mass conservation law (Richards, 1931)

$$\frac{\partial(\rho S\phi)}{\partial t} - \nabla \cdot (\rho \mathbf{v}) = \rho q \quad (16)$$

where ρ and S are the water density and the water saturation,

respectively; \mathbf{v} is the average flow velocity; ϕ is the medium porosity; and q is the source term. In the second example, example 5 in section 5.1.1 is used to simulate the saturated flow in the highly discretely fractured porous medium. Therefore, the governing equation can be simplified by substituting $S = 1$ into Eq. (16),

$$\frac{\partial(\rho\phi)}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = \rho q \quad (17)$$

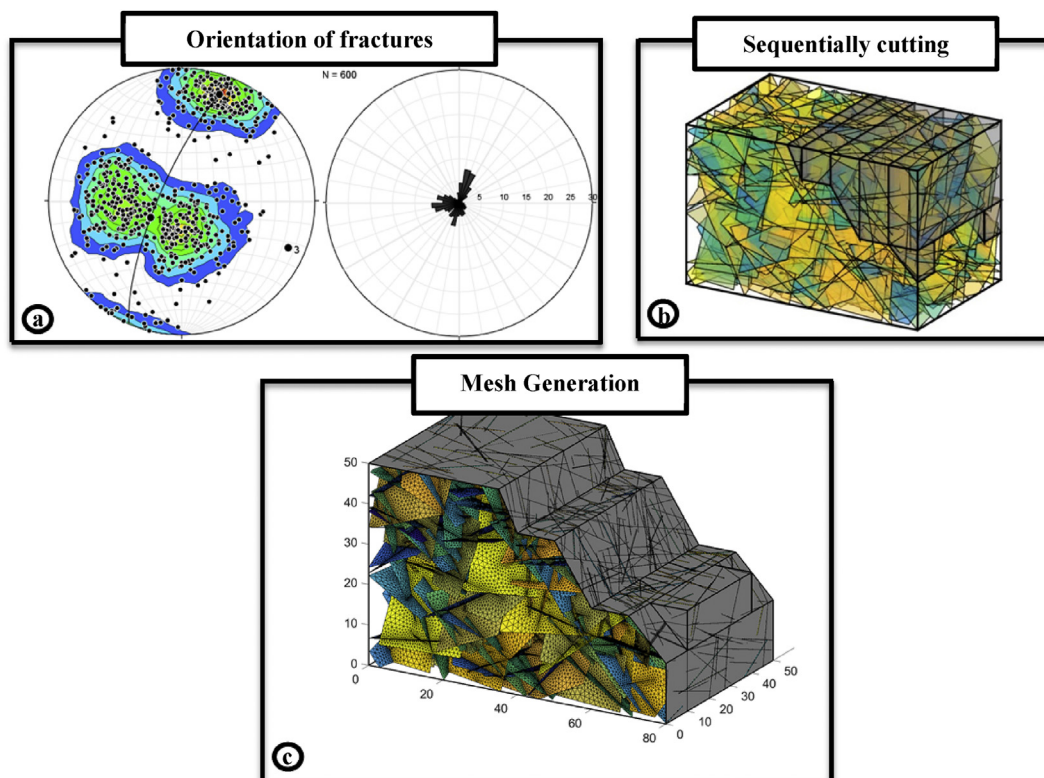


Fig. 11. An open-pit model with three sets of fractures (600 fractures): (a) Contoured stereonet and rose diagrams of the fracture poles; (b) DFN model and cutting block (grey blocks); (c) Planar mesh of the fractures after cutting: 1279498 triangles.

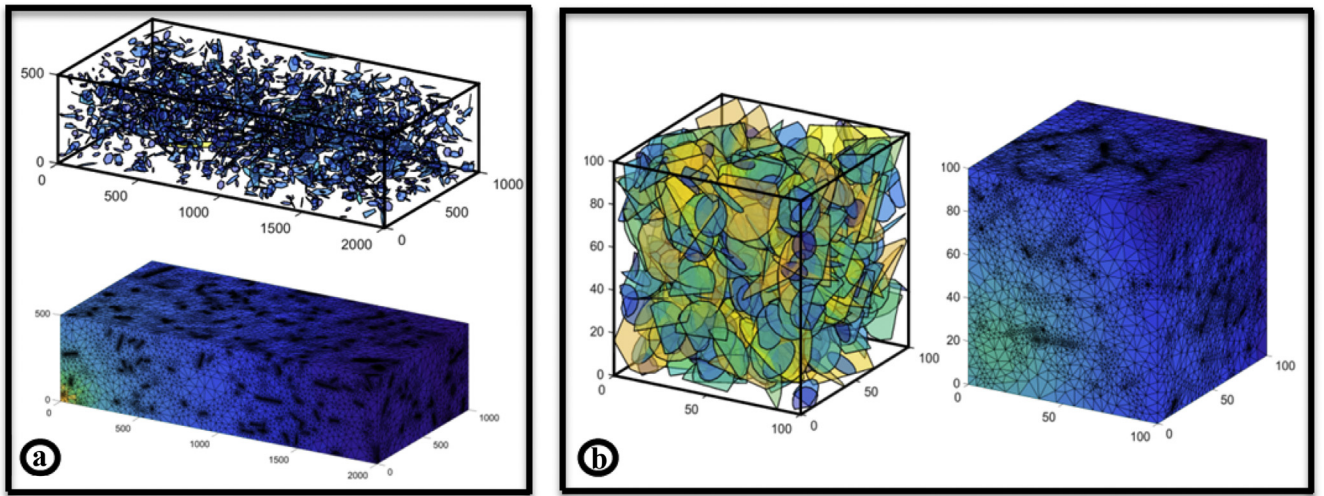


Fig. 12. Tetrahedral mesh of two different DFN models: (a) 2000 sparse fracture and 3535498 tetrahedra; (b) 450 highly intersected fractures and 1178472 tetrahedra.

In both examples, the water is assumed to flow in the pipe elements and obey to Darcy's law (Darcy, 1856). The average flow velocity \mathbf{v} in pipes can be expressed as

$$\mathbf{v} = -\frac{\mathbf{k}}{\mu}(\nabla p - \rho \mathbf{g}) \quad (18)$$

$$\mathbf{k} = \mathbf{k}_s k_r(S) \quad (19)$$

in which, p is the total pressure, μ is the dynamic viscosity of the water, \mathbf{g} is the gravitational acceleration, and \mathbf{k} , \mathbf{k}_s , and $k_r(S)$ are the effective permeability, the intrinsic permeability depending on the pore geometry, and relative permeability depending on the saturation, respectively. For saturated flow, \mathbf{k} equals to \mathbf{k}_s because $k_r(S) = 1$.

5.2.1. Water flow in fractured medium

The effect of the underground excavation in a fractured rock on the

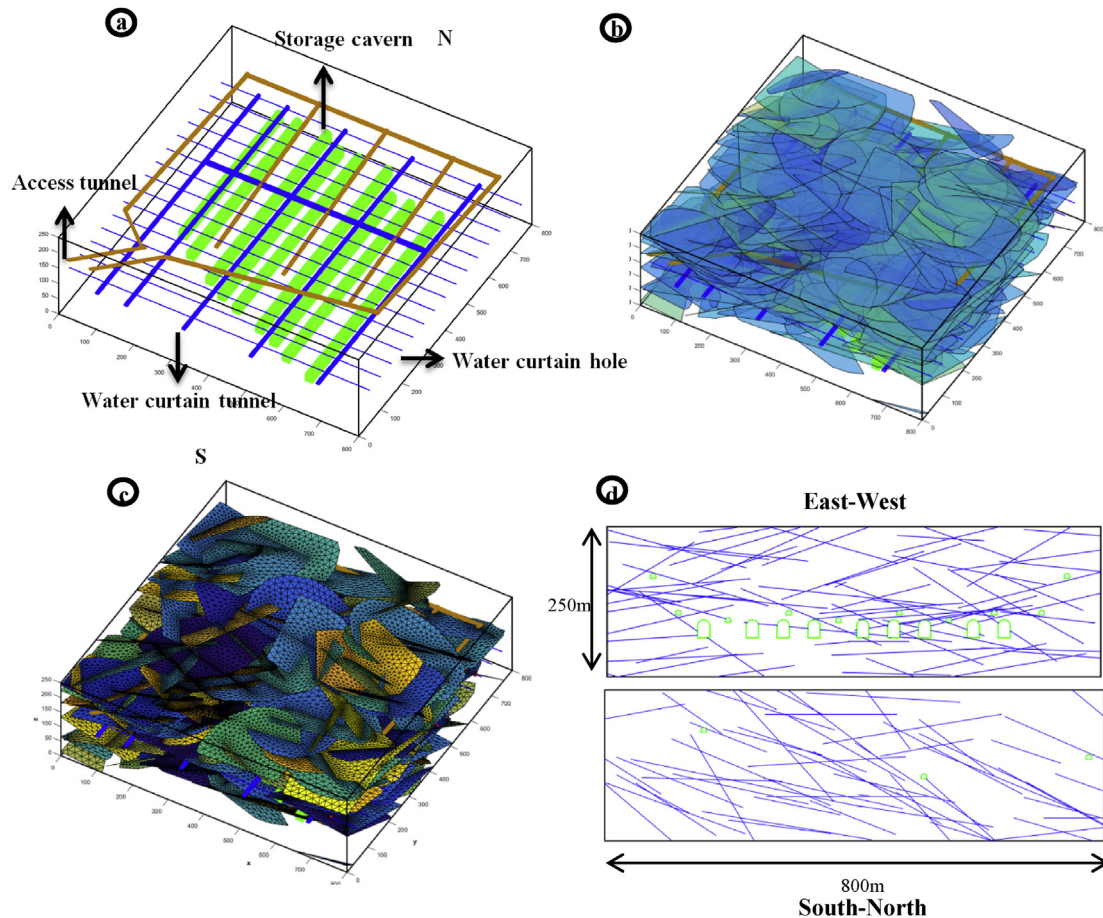


Fig. 13. Modeling a facility of underground petroleum storage: (a) Outline of the facility; (b) Adding 200 stochastic fractures into the model; (c) Planar mesh: 148316 triangles; (d) Slices at the middle position and in East-West and South-North orientations.

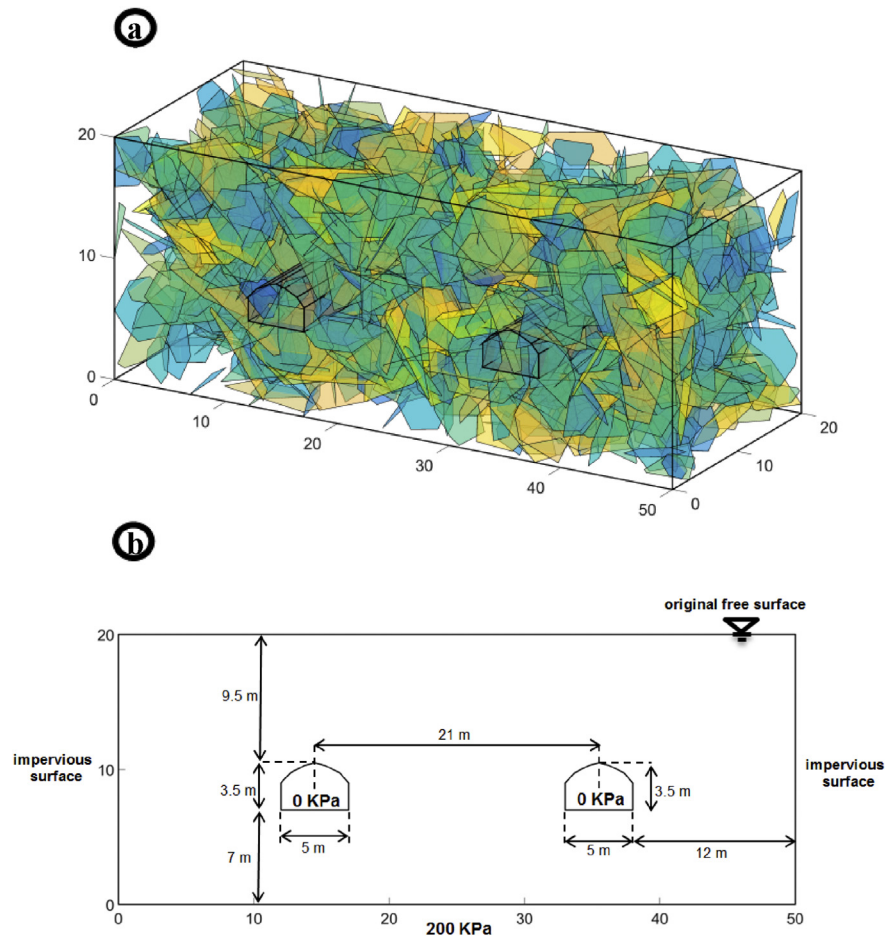


Fig. 14. Geometrical model for excavating two tunnels in a fractured rock: (a) 1000 random fractures and two excavated tunnels; (b) Front view of the model and boundary conditions.

water-free surface is simulated for verifying the triangular mesh. As shown in Fig. 14 (a), the dimension of the model is $50\text{ m} \times 20\text{ m} \times 20\text{ m}$, and 1000 random fractures are generated in the domain. Fig. 14(b) illustrates the detailed layout of the model and the boundary conditions in the simulation. The original free surface is located at the top boundary.

All boundary surfaces are impervious besides the bottom. A hydrostatic pressure of 200 kPa is imposed uniformly on the bottom boundary. Two identical tunnels are expected to be excavated at depth of 9.5 m. Each tunnel is 5 m wide and 3.5 m tall, and they are spaced at 21 m. The pressures on the boundaries of the tunnels are 0 kPa, and the inrush water

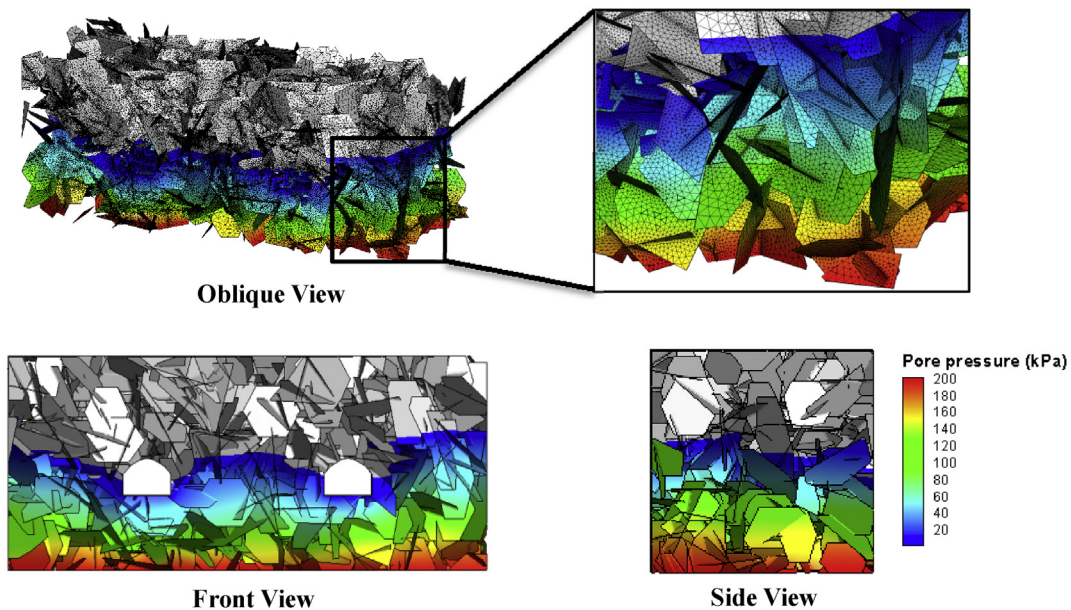


Fig. 15. Numerical result of the pore pressure distribution in the discrete fracture network.

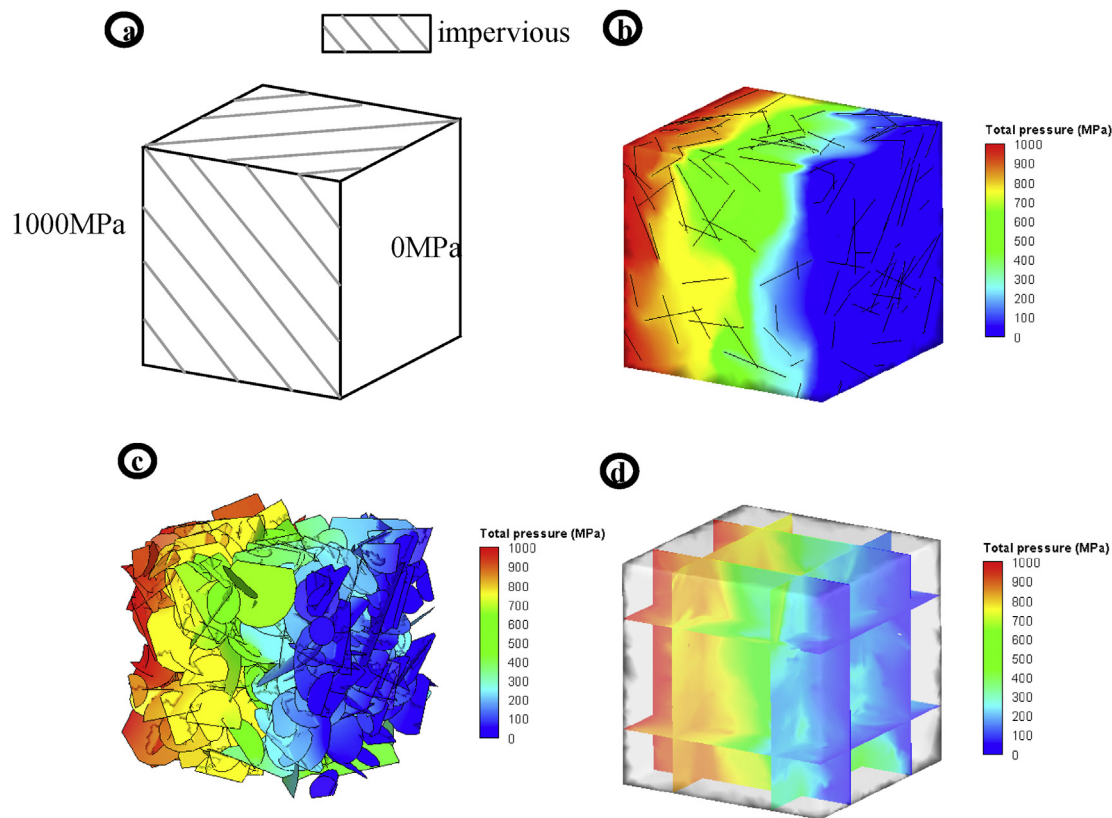


Fig. 16. Numerical simulation of the steady-state water flow in a fractured porous medium: (a) Boundary conditions; (b) The pore pressures along the surfaces of the domain; (c) The pore pressures on the fracture network; (d) The slices of the pressure distribution within the domain.

is assumed to be pumped out immediately. The permeability of fractures is $7.5 \times 10^{-7} \text{ m}^2$. After the two tunnels are excavated, the steady-state pore pressure distribution is simulated, and the results are displayed in Fig. 15. The contour of zero pore pressure is the eventual location of the free surface. It is obvious that the free surface drops dramatically in fractures during the excavation. In addition, the enlarged picture verifies that the conforming meshes allow for the smooth transition of the pore pressure on the fracture intersections.

5.2.2. Water flow in fractured porous medium

The simulation of saturated steady-state water in example 5 in section 5.1.1 is tested for demonstrating the generated tetrahedral meshes. The permeability of fractures is assumed to be much larger than the porous medium. Therefore, the permeability of fractures and porous medium are $7.5 \times 10^{-7} \text{ m}^2$ and $1 \times 10^{-15} \text{ m}^2$, respectively. The boundary conditions of the simulation are illustrated in Fig. 16(a). The high-pressure boundary condition of 1000 MPa is applied to the left side of the domain, and 0 MPa is applied to the right side. The rest of the surfaces in the domain are impervious. As Fig. 16(b) and (c) show, the gradient of pore pressure along the impervious surfaces and the fracture network are not uniform because of randomly distributed fractures. The slices in Fig. 16(d) also provide the sight of the pore pressure distribution within the domain.

6. Conclusion

In this study, a constrained Delaunay discretization method is developed to discretize the highly discontinuous geological media into adaptive triangular elements or tetrahedral elements. The high quality of triangular elements and good quality of tetrahedral elements have been verified in sections 4.1 and 4.2, respectively. The capacity of the proposed method to create complex models with discrete fracture networks and the utility of the generated meshes have also been demonstrated in section 5. However, the proposed method can only control the size and

quality of elements, not the number of triangles and tetrahedra in the final mesh. Moreover, the time-cost of the mesh generation method is also not optimal. In future work, the efficiency of the method will be improved to simulate fracture propagation, multiphase fluid, and mechanical fluid in highly discontinuous geological media.

Acknowledgments

The executable version (DFNsMeshGenerator3D) of the algorithm is available from <<https://github.com/wywy941/DFNsMeshGenerator3D>>.

Appendix A. Supplementary data

Supplementary data related to this article can be found at <http://dx.doi.org/10.1016/j.cageo.2017.07.010>.

References

- Bern, M.W., Plassmann, P.E., 1999. *Mesh Generation*. Pennsylvania State University.
- Berrone, S., Pieraccini, S., Scialò, S., 2014. An optimization approach for large scale simulations of discrete fracture network flows. *J. Comput. Phys.* 256, 838–853. <http://dx.doi.org/10.1016/j.jcp.2013.09.028>.
- Blackner, T.D., Stephenson, M.B., 1991. Paving-A new approach to automated quadrilateral mesh generation. *Int. J. Numer. Methods Eng.* 32, 881–847.
- Blessent, D., Therrien, R., MacQuarrie, K., 2009. Coupling geological and numerical models to simulate groundwater flow and contaminant transport in fractured media. *Comput. Geosci.* 35, 1897–1906. <http://dx.doi.org/10.1016/j.cageo.2008.12.008>.
- Cacace, M., Blocher, G., 2015. MeshIt — a software for three dimensional volumetric meshing of complex faulted reservoirs 5191–5209. <http://dx.doi.org/10.1007/s12665-015-4537-x>.
- Chew, P.L., 1989. Constrained Delaunay triangulations. *Algorithmica* 4, 97–108.
- Darcy, H., 1856. *Les fontaines publiques de la ville de Dijon: exposition et application*. Victor Dalmont.
- De Dreuzy, J.R., Méheust, Y., Pichot, G., 2012. Influence of fracture scale heterogeneity on the flow properties of three-dimensional discrete fracture networks (DFN). *J. Geophys. Res. B Solid Earth* 117, 1–22. <http://dx.doi.org/10.1029/2012JB009461>.
- Edelsbrunner, H., 2001. *Geometry and Topology for Mesh Generation*. Cambridge University Press, Cambridge.

- Erhel, J., De Dreuzy, J.-R., Poiriez, B., 2009. Flow simulation in three-dimensional discrete fracture networks. *SIAM J. Sci. Comput.* 31, 2688–2705. <http://dx.doi.org/10.1137/080729244>.
- Ertlen, H., Üngör, A., 2007. Triangulations with locally optimal Steiner points. In: *Eurographics Symposium on Geometry Processing*, pp. 143–152. <http://dx.doi.org/10.1137/080716748>.
- Field, D.A., 2000. Qualitative measures for initial meshes. *Int. J. Numer. Methods Eng.* 47, 887–906. [http://dx.doi.org/10.1002/\(SICI\)1097-0207\(20000210\)47:4<887::AID-NME804>3.0.CO;2-H](http://dx.doi.org/10.1002/(SICI)1097-0207(20000210)47:4<887::AID-NME804>3.0.CO;2-H).
- Hyman, J.D., Gable, C.W., Painter, S.L., Makedonska, N., 2014. Conforming Delaunay triangulation of stochastically generated three dimensional discrete fracture networks: a feature rejection algorithm for meshing strategy. *SIAM J. Sci. Comput.* 36, 1871–1894.
- Jing, L., 2003. A review of techniques, advances and outstanding issues in numerical modelling for rock mechanics and rock engineering. *Int. J. Rock Mech. Min. Sci.* 40, 283–353. [http://dx.doi.org/10.1016/S1365-1609\(03\)00013-3](http://dx.doi.org/10.1016/S1365-1609(03)00013-3).
- Li, S.C., Xu, Z.H., Ma, G.W., Yang, W.M., 2014. An adaptive mesh refinement method for a medium with discrete fracture network: the enriched Persson's method. *Finite Elem. Anal. Des.* 86, 41–50. <http://dx.doi.org/10.1016/j.finel.2014.03.008>.
- Maryška, J., Severýn, O., Vohralík, M., 2005. Numerical simulation of fracture flow with a mixed-hybrid FEM stochastic discrete fracture network model. *Comput. Geosci.* 8, 217–234. <http://dx.doi.org/10.1007/s10596-005-0152-3>.
- Murphy, M., Gable, C.W., 1998. Strategies for nonobtuse boundary Delaunay triangulations. *IMR* 309–320.
- Murphy, M., Mount, D.M., Gable, C.W., 2001. A point-placement strategy for conforming Delaunay tetrahedralization. *Int. J. Comput. Geom. Appl.* 11, 669–682.
- Mustapha, H., Dimitrakopoulos, R., Graf, T., Firoozabadi, A., 2011. An efficient method for discretizing 3D fractured media for subsurface flow and transport simulations. *Int. J. Numer. Methods Fluids* 67, 651–670. <http://dx.doi.org/10.1002/fld.2383>.
- Mustapha, H., Mustapha, K., 2007. A new approach to simulating flow in discrete fracture networks with an optimized mesh. *SIAM J. Sci. Comput.* 29, 1439–1459.
- Parthasarathy, V.N., Graichen, C.M., Hathaway, A.F., 1994. A comparison of tetrahedron quality measures. *Finite Elem. Anal. Des.* 15, 255–261. [http://dx.doi.org/10.1016/0168-874X\(94\)90033-7](http://dx.doi.org/10.1016/0168-874X(94)90033-7).
- Peraire, J., Vahdati, M., Morgan, K., Zienkiewicz, O.C., 1987. Adaptive remeshing for compressible flow computations. *J. Comput. Phys.* 72, 449–466. [http://dx.doi.org/10.1016/0021-9991\(87\)90093-3](http://dx.doi.org/10.1016/0021-9991(87)90093-3).
- Persson, P.-O., 2005. *Mesh Generation for Implicit Geometries*. Ph.D Thesis. Massachusetts Institute of Technology.
- Persson, P., Strang, G., 2004. A simple mesh generator in MATLAB. *SIAM Rev.* 46, 329–345.
- Pichot, G., Erhel, J., De Dreuzy, J.R., 2010. A mixed hybrid mortar method for solving flow in discrete fracture networks. *Appl. Anal.* 89, 1629–1643.
- Richards, L.A., 1931. Capillary conduction of liquids through porous mediums. *J. Appl. Phys.* 1, 318–333. <http://dx.doi.org/10.1063/1.1745010>.
- Ruppert, J., 1995. A Delaunay refinement algorithm for quality 2-dimensional mesh generation. *J. Algorithms* 18, 548–585. <http://dx.doi.org/10.1006/jagm.1995.1021>.
- Shewchuk, J.R., 2002. Delaunay refinement algorithms for triangular mesh generation. *Comput. Geom.* 22, 21–74. [http://dx.doi.org/10.1016/S0925-7721\(01\)00047-5](http://dx.doi.org/10.1016/S0925-7721(01)00047-5).
- Si, H., 2015. TetGen, a delaunay-based quality tetrahedral mesh generator. *ACM Trans. Math. Softw.* 41, 11.
- Si, H., Shewchuk, J.R., 2014. Incrementally constructing and updating constrained Delaunay tetrahedralizations with finite-precision coordinates, pp. 253–269. <http://dx.doi.org/10.1007/s00366-013-0331-0>.
- Tavakkoli, M., Mohammadzadeh, M., Shaheabadi, A., Khajoei, S., Malakooti, R., Beidokhti, M.S., 2009. Deterministic versus stochastic discrete fracture network (DFN) modeling, application in a heterogeneous naturally fractured reservoir. In: *Kuwait International Petroleum Conference and Exhibition*. Society of Petroleum Engineers, Kuwait City. <http://dx.doi.org/10.2118/127086-MS>.
- Üngör, A., 2009. Off-centers: a new type of Steiner points for computing size-optimal quality-guaranteed Delaunay triangulations. *Comput. Geom. Theory Appl.* 42, 109–118. <http://dx.doi.org/10.1016/j.comgeo.2008.06.002>.
- Verbree, E., 2006. Piecewise linear complex representation through conforming Delaunay tetrahedralization. In: *Geographic information science, fourth international conference. GIScience*, pp. 385–387.
- Vu, M.N., Pouya, A., Seyed, D.M., 2013. Modelling of steady-state fluid flow in 3D fractured isotropic porous media: application to effective permeability calculation. *Int. J. Numer. Anal. Methods Geomech.* 37, 2257–2277. <http://dx.doi.org/10.1002/nag.2134>.
- Xu, C., Dowd, P., 2010. A new computer code for discrete fracture network modelling. *Comput. Geosci.* 36, 292–301. <http://dx.doi.org/10.1016/j.cageo.2009.05.012>.
- Zehner, B., Börner, J.H., Götz, I., Spitzer, K., 2015. Computers & Geosciences Work flows for generating tetrahedral meshes for finite element simulations on complex geological structures. *Comput. Geosci.* 79, 105–117. <http://dx.doi.org/10.1016/j.cageo.2015.02.009>.
- Zhuang, X., Huang, R., Liang, C., Rabczuk, T., 2014. A coupled thermo-hydro-mechanical model of jointed hard rock for compressed air energy storage. *Math. Probl. Eng.* 2014 <http://dx.doi.org/10.1155/2014/179169>.