Engineering Computations

An efficient algorithm for generation of conforming mesh for three-dimensional discrete fracture networks

Soheil Mohajerani, Duruo Huang, Gang Wang, Seyed-Mohammad Esmaeil Jalali, Seyed Rahman Torabi,

Received 13 March 2018
Revised 11 July 2018
15 August 2018
Accepted 3 September 2018

An efficient algorithm for generation of conforming mesh for three-dimensional discrete fracture networks

Soheil Mohajerani
Shahrood University of Technology, Shahrood, Iran

Duruo Huang
Tsinghua University, Beijing, China

Gang Wang
Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong, and

Seyed-Mohammad Esmaeil Jalali and Seyed Rahman Torabi
Shahrood University of Technology, Shahrood, Iran

Abstract

Purpose – This study aims to develop an efficient algorithm for generation of conforming mesh for seepage analysis through 3D discrete fracture networks (DFN).

Design/methodology/approach – The algorithm is developed based on a refined conforming Delaunay triangulation scheme, which is then validated using analytical solutions. The algorithm is well able to meet the challenge of meshing complex geometry of DFNs.

Findings – A series of sensitivity analysis have been performed to evaluate the effect of meshing parameters on steady state solution of Darcy flow using a finite element scheme. The results show that an optimized minimum internal angle of meshing elements should be predetermined to guarantee termination of the algorithm.

Originality/value – The developed algorithm is computationally efficient, fast and is of low cost. Furthermore, it never changes the geometrical structure and connectivity pattern of the DFN.

Keywords FEM, Triangulation, Conforming mesh, Delaunay, DFN, Meshing

Paper type Research paper

1. Introduction

Numerical simulation of fluid flow through fractured rocks plays a vital role in many applications in energy industry, such as hydrocarbon reservoirs, geothermal resources, underground fluid storage, groundwater aquifers, nuclear waste disposal and clearance of contaminated areas in the fractured rocks (Wang *et al.*, 2001; Ren *et al.*, 2017; Xue, 2017). Generally, numerical models of the fluid flow through fractured rocks can be classified into

The author (GW) acknowledges support from Theme-based Research Scheme Grant No. T22-603-15N from the Hong Kong Research Grants Council.

Conflicts of Interest: The authors declare that they have no conflicts of interest.

Engineering Computations
Vol. 35 No. 8, 2018
pp. 2860-2882
© Emerald Publishing Limited
0264-4401
DOI 10.1108/EC-03-2018-0127

three subcategories: the equivalent continuum model, the double-medium model and the discrete fracture network (DFN) (Karimi-Fard and Firoozabadi, 2003; Parashar and Reeves, 2012; Zhang and Yin, 2014; Xie and Wang, 2014; Ye *et al.*, 2016; Mohajerani *et al.*, 2017). The DFN is one of the most widely used methods to simulate fluid flow through fractured rocks. In this method, the effect of discrete discontinuities on the fluid flow is explicitly considered with the assumption of impermeability of rock matrix.

The foundation of DFN method is partition of a *n*-dimensional domain to a *n*-1-dimensional statistically distributed set of fractures. The partition has a significant effect on computational cost of the flow models, particularly for the three-dimensional models. The fracture locations are generated in a desired domain using a statistical process, and the geometrical parameters of the fractures, such as the orientation (dip and dip direction), and length are modeled using the probability density functions (PDF) based on sampling methods (Jin *et al.*, 2003; Gallager, 2012). Geological data directly mapped from wellbores, surface outcrops, trenches using one of the mapping methods (the scanline, mapping window and circular estimator) and geophysical mappings are the most important part of DFN modeling (Wu *et al.*, 2011). The shape of the fractures is a hypothetical parameter and often simulated by circles, ellipses and polygons in the various research works (Dershowitz and Einstein, 1988; Han *et al.*, 2016). Hydrological parameters such as aperture and roughness of the fracture wall surfaces may be estimated either by laboratory tests or by using in situ field tests. These parameters can be assigned to the location of fractures as a constant value or a PDF (Tsang, 1992).

In the literature, various numerical solution schemes have been used to solve the fluid flow problem in the fractured rocks. In this regard, the finite element method (FEM) (Karimi-Fard and Firoozabadi, 2003; Moradi *et al.*, 2017), finite volume method (FVM) (Koudina *et al.*, 1998), boundary element method (BEM) (Olson, 1993), finite difference method (FDM) (Rutqvist *et al.*, 2013), discontinuous deformation analysis (DDA) (Jing *et al.*, 2001) and hybrid methods (Elsworth, 1986) have been given considerable attention. In general, such these methods require a high-quality meshing framework to solve the flow with adequate precision.

Unfortunately, there are some serious challenges to discrete a DFN model into a high-quality mesh. On one hand, the structured meshing is not convenient to represent complex three-dimensional geometry of a fractured medium and on the other hand, a high-quality unstructured meshing must be able to meet particular geometrical requirements (Mustapha *et al.*, 2011). As shown in Figure 1, a network of statistically generated fractures can include

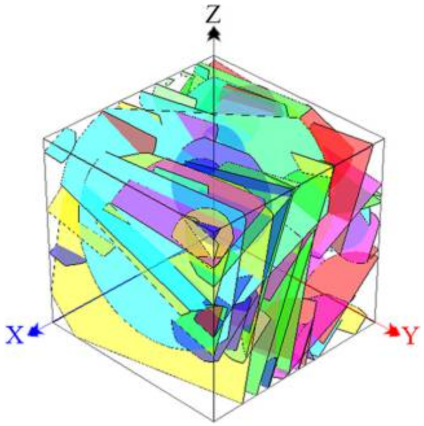


Figure 1. Realization of DFN generated by the developed code in the research

fractures' length spanning over several orders of magnitude. To solve the flow field in small fractures, they must have small enough meshing elements compatible to their length. A good mesh requires a balance between two sides: its elements should not be too small to avoid significant increase in computational costs, yet, they should not be too large to deteriorate the numerical precision. Also, due to complex structure of a three-dimensional DFN with arbitrary shape and spatial position of fractures, challenges are encountered to mesh parallel or crossover fractures. If the distance between parallel intersections, or the angle between crossover intersections, is too small, low-quality meshing may be generated. The meshing elements may lead to an ill-conditioned discretization matrix and cause divergence in numerical solution (Hyman *et al.*, 2014).

To date, conforming and non-conforming meshing methods have been developed (Hautefeuille *et al.*, 2009; Mackerle, 2001). The conforming mesh refers to the case that nodes on the intersection line are unique and common to these two intersecting fractures. On the other hand, the non-conforming mesh discretizes each fracture plane independently. Although, the non-confirming mesh is more flexible, additional system of equations has to be implemented to ensure continuity of the hydraulic head and flow rate on the intersection of fractures, therefore, the solution scheme may be time-consuming (Hyman *et al.*, 2014).

Koudina *et al.* (1998) provided one example of a conforming meshing algorithm for DFN based on the advancing front technique. An alternative algorithm based on the paving method was proposed by Wang *et al.* (2016). Although, these methods are used successfully in simple networks of fractures, they cannot well meet the aforementioned challenges when the number of fractures increases and the geometrical structure of DFN becomes more complex. The challenge was partially addressed by Maryška *et al.* (2005). In their method, however, geometrical structure of DFN changes during meshing. The intersections of fractures are changed with length variation and displacement. It can change the connectivity pattern of fractures; therefore, it seems that the mesh is no longer a faithful representative of geometrical structure of the DFN.

Two similar meshing methods were also suggested by Mustapha and Mustapha (2007) and Erhel *et al.* (2009). Although, these methods can generate a high-quality mesh, they are not able to model intersection of more than two fractures. The method was generalized later to improve meshing quality. Another generalization of these methods was developed by Karimi-fard *et al.* (2003), in which a conforming mesh is generated by altering the geometric structures of the DFM through adding, moving, deleting and merging the vertices of meshing triangles. Method proposed by Hyman *et al.* (2014) uses the feature rejection algorithm (FRAM) to prevent creation of inconvenient fractures during generating DFN. However, the geometry of the network and the connectivity pattern of fractures are altered. Li *et al.* (2014) provided a method to generate a conforming mesh for DFN in which using Persson and Strang meshing generator, the location of vertices of triangles are determined with solving a system of equations of force balance in trusses and results in a high-quality mesh. This method is not optimal due to high computational cost and is not able to meet the meshing challenges completely. Adaptive meshing could be implemented to refine the mesh locally based on solution variables, for example, Hernández *et al.* (1997). Zhang (2015) developed another interesting algorithm to triangulate complex DFNs by two steps: firstly, subdividing 2D domains of the fractures into closed loops surrounded by intersection lines between fractures, then, triangulating these arbitrary loops without adding any node.

Some studies have been focused on developing the non-conforming meshing methods. As mentioned before, these methods require more computational efforts than the conforming methods; therefore, they are not generally appropriate for networks of a huge number of fractures. Benedetto *et al.* (2016) provided a combined conforming and non-conforming method to solve a fluid flow problem in DFN using the virtual element method (VEM). In this method, some additional unique vertices are added on intersections of fractures and each fracture is meshed independently. The application of this method is limited to VEM.

The main purpose of this research is developing a new computationally efficient algorithm to mesh three-dimensional DFN structures. This algorithm maintains the geometrical structure of the network and thereby connectivity pattern of fractures remains unchanged. Triangular elements of the meshing structure are generated based on Delaunay criterion and are refined to increase the quality. Thus, discretization matrices assembled by the numerical schemes are not ill-conditioned and often solution converges using this algorithm. Because of a large number of fractures with a wide range of lengths in DFN, an optimal element size is proposed to reduce computational cost and achieve high numerical precision. Furthermore, this algorithm is able to well cover the critical meshing conditions such as junction of two fracture intersections with a small angle or two parallel fracture intersections with a small distance on third fracture plane.

This paper is organized as following: in Section 2, the algorithm of DFN generation, finding the intersections of the fractures, triangulation and refinement are provided. In Section 3, the details of the validation of the present algorithm with the finite element scheme, solutions of the flow problem in two simple regular geometrical structures and comparison of the results with the analytical results are discussed. Also, a series of sensitivity analyses on the meshing parameters in a complex DFN are conducted to determine the effect of various parameters and demonstrate the performance of the algorithm.

2. The meshing algorithm

In this section, various algorithm and methods used to develop the present meshing algorithm are described.

2.1 Generation of DFN

Depending on geological origin, the rock fractures are grouped into joint sets that has similar geometrical properties (dip and dip direction). In a three-dimensional geometrical model, the joint sets are estimated using the hemispherical projection. The joint sets are simulated independently and the ultimate model is a union of all. Each joint set includes certain geometrical distribution parameters such as, the location, orientation (dip and dip direction) and length of planar fractures.

The location of fractures is the first parameter that must be considered in simulation of the joint-sets. A single-point homogeneous Poisson process is generally used to determine the location of fractures in the domain of model. Given a constant number as fracture intensity (λ) (the number of fracture planes per unit volume of the model), the location is distributed in three-dimensional space. The average of Poisson distribution is calculated from equation (1):

$$\mu = \lambda \cdot V_m, \tag{1}$$

where, V_m is volume of the model. As the center of some of fractures is outside of the model domain, while their length is large enough to enter it and affect the connectivity pattern, the domain of the generation of the fractures is initially considered to be several times larger than model domain. The cube of model is extracted from generating domain after the completion of generating process. To generate the location of fractures, a random variable (η) from Poisson distribution function is generated using equation (2) (Xu and Dowd, 2010):

$$P(\eta = n) = e^{-\mu\frac{\mu^n}{n!}}, \tag{2}$$

A sequence of random numbers (x_i) in the range of [0,1] as far as, $\prod_{i=1}^{k} x_i < e^{-\mu}$, are generated using uniform distribution function. For any $n = k$ events, three independent values are calculated by setting P in the uniform distribution function. These values are considered as coordinates of the location of the fracture (o). Figure 2 demonstrates an image of the generated locations of DFN fractures.

After that, the orientation and length of fractures are generated using PDF and Monte Carlo sampling, these and are assigned to the locations of the fractures. The parameters, dip (α), dip direction (β) and rotation angle (γ) have been schematically shown in Figure 3. The uniform and Fisher PDF are usually used to model the dip and dip directions, respectively. The rotation angle is modeled by uniform PDF as well (Baghbanan and Jing, 2008).

In the literature, the power-law or log-normal is used as PDF of the fracture length (L). The shape of fractures is a hypothetical parameter, which is simulated as circular, elliptical or polygonal (Figure 3). Then, hydraulic parameters of fractures such as aperture and roughness are assigned to the location of fractures as required similar to the generation of geometrical parameters. The PDF of the aperture is usually uniform (Baghbanan and Jing, 2008).

A number of equations have been suggested to determine the relation between the aperture and length of fractures. An example of such equations has been represented in equation (3) (Vermilye and Scholz, 1995):

$$a = \varsigma\sqrt{L}, \tag{3}$$

where a is the aperture in millimeter, L is the fracture length in millimeter and ς is a constant coefficient which is determined depending to the conditions of fractures and in general case is equal to 0.004 (Vermilye and Scholz, 1995). Various PDFs and their parameters required to generate DFN, have been listed in Table I.

As mentioned before, after generating the fractures, the desired representative elementary volume (REV) is extracted from a much bigger originally generated domain.

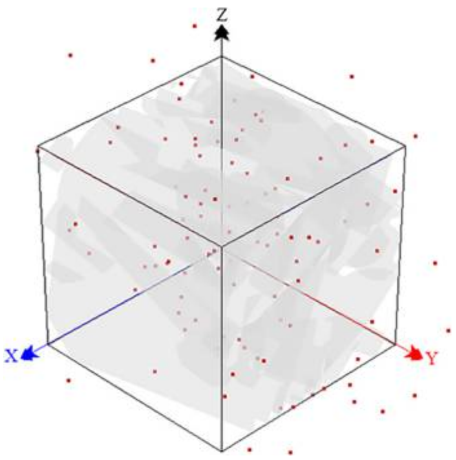


Figure 2.
Locations of DFN fractures generated using developed code in the research

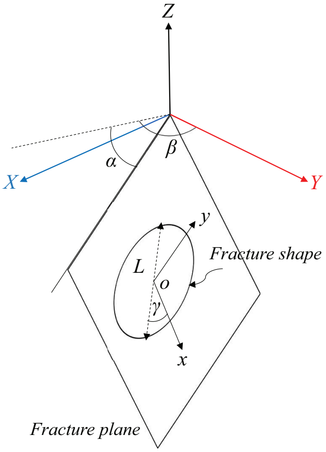


Figure 3
A schematic fracture, its geometrical parameters and the global (X, Y, Z) and local (x, y) systems of coordinates

Each fracture and the set of all the fractures of the model are shown by f_i and $F = \bigcup_{i=1}^{N_f} f_i$, respectively, in which N_f is the number of all fractures. Referring to Figure 3, two systems of coordinates are introduced. The global Cartesian system (X, Y, Z) defines a three-dimensional space and the local coordinate (x, y) defines a two-dimensional plane on each fracture.

2.2 Determination of intersection of fractures

In this research, the fractures are considered planar, and the intersection of two fractures is a linear segment. For the sake of determination of coordinates of two ends of an intersection segment in (X, Y, Z), each fracture is intersected by other fractures and the boundary facets of the model. Figure 4(a) shows how intersections are formed. Each of intersections and the set of all of them are represented by $s_i = f_j \cap f_k$, where $j, k = 1 \ldots N_f$ and $S = \bigcup_{i=1}^{N_s} s_i$, respectively, where, N_s is the number of all intersections of the model. Therefore, the domain is characterized as $\Omega = F \cup S$.

Fractures can have one of three main types of the connectivity with other fractures or boundaries of the model: multiple connectivity (persistent fractures), only one connection (dead-end fractures) and no connection (single fractures). As Figure 4(b) shows, the persistent fractures (blue colored) usually have a larger length and several (at least two) intersections with the other entities. Such fractures can be intersected by the boundaries of the model or be connected to dead-end fractures and be completely located inside the model. However, dead-end (green colored) and single (red colored) fractures can have important effects on ultimate strength and mechanical properties of rock-mass, they do not have significant effect on its hydraulic properties. Because hydraulic analyses are the main aim of generation of three-dimensional DFNs in this research, it is convenient to remove dead-end and single fractures from the model domain. It dramatically increases performance and speed of solution, particularly if the model deals with a large number of fractures. Therefore, the isolated and dead-end fractures are searched within the set F. Accordingly, these fractures and their corresponding intersections s are removed from the sets of F and S, respectively. Figure 4(c) shows an image of the identified intersections in the model domain.

Table I.
PDF and their parameters

PDF	Formula	Parameters
Uniform	$f(x)=\begin{cases}\dfrac{1}{b-a} & a\le x\le b\\ 0 & \text{otherwise}\end{cases}$	a, b
Fisher	$f(\theta)=k\sin\theta\, e^{k\cos\theta}/e^{k}-e^{-k}$	θ, k
Normal	$f(x)=\begin{cases}\dfrac{1}{\sqrt{2\pi}\sigma}e^{-\dfrac{(x-\mu)^2}{2\sigma^2}} & x>a\\ 0 & x\le a\end{cases}$	a, μ, σ
Log-normal	$f(x)=\begin{cases}\dfrac{1}{\sqrt{2\pi}\sigma e^{x}}e^{-\dfrac{(\ln(x-a)-\mu)^2}{2\sigma^2}} & x>a\\ 0 & x\le a\end{cases}$	a, μ, σ
Power-law	$f(x)=ax^{-k}$	a, k
Negative exponential	$f(x)=\begin{cases}\lambda\, e^{-\lambda(x-\mu)} & x>\mu\\ 0 & x\le\mu\end{cases}$	λ, μ

2.3 Mesh generation

The present meshing algorithm generates triangular elements in a two-dimensional local coordinate system (x, y) on the surface of each fracture using Delaunay criterion. The ultimate meshing geometry in the three-dimensional global system of coordinates (X, Y, Z) is a union of all planar triangles which is transposed using a transpose matrix. As will be discussed in the following section, there are vertices of triangles on intersection of fractures. Because these vertices are unique, it results in a conforming mesh. To increase the quality of triangulation, the Ruppert algorithm is used to refine low-quality triangles (Cheng *et al.*, 2012). This algorithm provides an optimized unstructured triangulation to mesh the complex DFN geometry and random shape and position of fractures and their intersections in three-dimensional space, without changing geometrical structure of DFN to avoid change in connectivity pattern of fractures.

The present algorithm includes four main steps:

- *Step I*: Vertices $(v_{s_i}^{j})$ on the intersection lines are formed in S.
- *Step II*: Boundary vertices $(v_{b_i}^{j})$ are formed on the boundaries of the fractures (Γ_{f_i}).
- *Step III*: A Delaunay-based triangulation (T_i) is generated using a set of all vertices of fracture f_i, $V_i^j=\left(\bigcup_{j=1}^{N_{vs_i}}v_{s_i}^{j}\right)\cup\left(\bigcup_{j=1}^{N_{vb_i}}v_{b_i}^{j}\right)$, where N_{vs_i} and N_{vb_i} are the total number of intersection and boundary vertices of f_i, respectively.
- *Step IV*: T_i is refined using Ruppert algorithm.
- Steps I to IV are repeated for $i = 1\ldots.N_f$.

The flowchart of the algorithm is illustrated in Figure 5. In Step I, find the possible crossover where the intersection line s_i from the S is intersected by another intersection s_j.

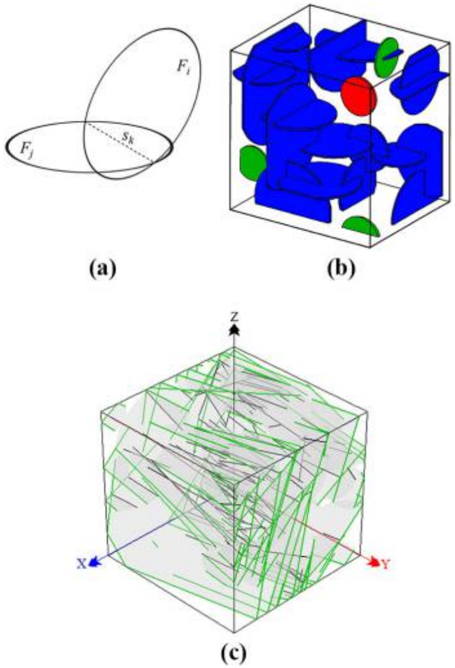


Figure 4. Schematics of (a) formation of intersections, (b) isolated (red), dead-end (green) and persistent fractures and (c) intersections of DFN fractures on boundaries of the model (green) and inside the model (black) identified by the developed code

If such a crossover is found, the crossover $v_{s_i}^k = s_i \cap s_j$, where $i,j = 1,\cdots,N_s$ is registered as a vertex and a coverage radius (h_s) is associated with it. Note that h_s governs the mesh size and is discussed in the next section. Then, the middle point of the intersection segment s_i is identified using coordinates of its two ends. If this point is not inside a sphere centered at $v_{s_i}^k$ with a radius of h_s, a new vertex ($\hat{v}_{s_i}$) is registered with coordinates of the middle point and h_s is associated to it. Attention should be given that in order to maintain connectivity pattern of fractures, at least one of the vertices, $v_{s_i}^k$ or $\hat{v}_{s_i}$, must be saved if both of them are located in the same circle within coverage radius of previous registered vertices. After that, an iteration loop is created and for each iteration i, two vertices with spacing $i \times h_s$ are characterized on both sides of the midpoint $\hat{v}_{s_i}$ on the intersection line s_i.

These vertices are registered with the previous ones that they are not inside a sphere with radius of h_s and center of previous saved vertices. These vertices are named $\check{v}_{s_i}\ k$ and a coverage radius h_s is dedicated to them. The loop is terminated when the distance between two identified vertices is larger than length of segment s_i.

This process goes on to determine all unique intersection vertices in (X, Y, Z) for all s_i, where $i = 1,\ldots, N_s$. The set of these vertices is represented by $V_s = \bigcup_{i=1}^{N_s}\left(\bigcup_k v_{s_i}^k \cup \check{v}_{s_i}\ k\right) \cup \hat{v}_{s_i}$.

In the second step, the algorithm is focused on the fracture plane f_i, where $i = 1,\ldots, N_f$ from the set of F. All intersections (s_j) of S included in f_i, are identified. Then, the vertices of s_j are selected from V_s and placed in V_i^j. The coordinates of these vertices are transposed from global (X, Y, Z) to local (x, y) on the fracture plane using a transpose matrix. The boundary vertices $v_{b_i}^j$ are equally spaced on the perimeter boundary of the fracture (Γ_{f_i}) with the spacing h_s. These vertices are added to V_i^j and a h_s is associated to them if they are

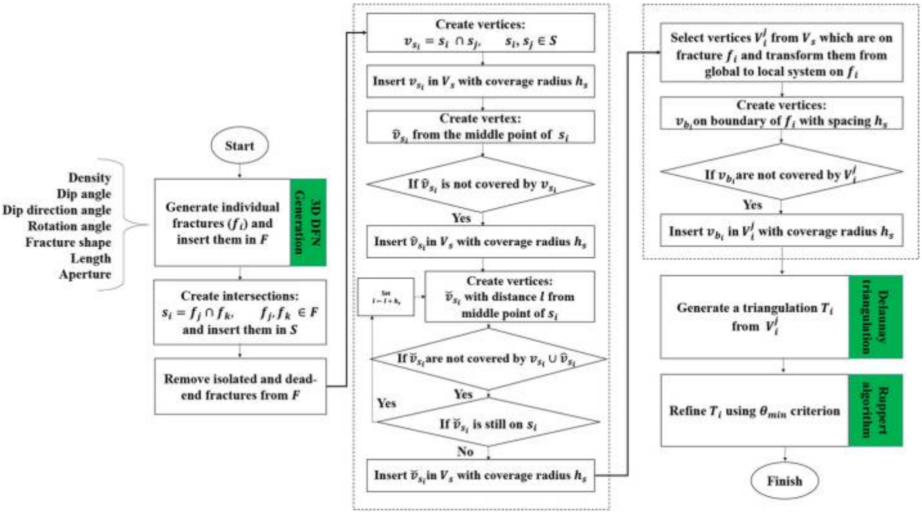

Start
Density
Dip angle
Dip direction angle
Rotation angle
Fracture shape
Length
Aperture
Generate individual fractures (f_i) and insert them in F
3D DFN Generation
Create intersections: $s_i = f_j \cap f_k$, $f_j, f_k \in F$ and insert them in S
Remove isolated and dead-end fractures from F
Create vertices: $v_{s_i} = s_i \cap s_j$, $s_i, s_j \in S$
Insert v_{s_i} in V_s with coverage radius h_s
Create vertex: $\hat{v}_{s_i}$ from the middle point of s_i
If $\hat{v}_{s_i}$ is not covered by v_{s_i}
Yes
Insert $\hat{v}_{s_i}$ in V_s with coverage radius h_s
Set $l = l + h_s$
Create vertices: $\breve{v}_{s_i}$ with distance l from middle point of s_i
If $\breve{v}_{s_i}$ are not covered by $v_{s_i} \cup \hat{v}_{s_i}$
Yes
Yes
If $\breve{v}_{s_i}$ is still on s_i
No
Insert $\breve{v}_{s_i}$ in V_s with coverage radius h_s
Select vertices V_i^j from V_s which are on fracture f_i and transform them from global to local system on f_i
Create vertices: v_{b_i} on boundary of f_i with spacing h_s
If v_{b_i} are not covered by V_i^j
Yes
Insert v_{b_i} in V_i^j with coverage radius h_s
Generate a triangulation T_i from V_i^j
Delaunay triangulation
Refine T_i using θ_{min} criterion
Ruppert algorithm
Finish

Figure 5.
Flowchart of the proposed meshing algorithm

"covered" by previously registered vertices in V_i^j. At the end of this step, V_i^j is obtained with the total number of the vertices (N_{v_i}) on the fracture f_i.

A Delaunay-based triangulation can be generated with any arbitrary set of vertices (Cheng *et al.*, 2012). Therefore, a Delaunay-based triangulation (T_i) is generated with V_i^j in the third step. In a two-dimensional space, T_i is Delaunay-based if and only if empty-circle criterion is satisfied for all elements of T_i. This criterion checks whether the circumcircle of the jth triangle (t_i^j) includes another vertex except t_i^j or not. Figure 6 displays a schematic of the empty-circle criterion and Delaunay-based triangulation. As shown in this figure, empty-circle criterion means that no vertex is inside the circumcircles of triangles. Moreover, three independent vertices of V_i form a triangle must satisfy the visibility requirement. That is, these vertices must be on the same open surface domain of s_i. The visibility criterion results in independent meshing on each side of intersections.

Till this step, the size and quality of the formed triangles may still not be appropriate in the unstructured T_i and may still result in ill-conditioned discretization matrices. In fact, there will be many short edges and small acute angles resulting in bad-shaped triangles if arbitrary DFNs are considered. In this case, using Delaunay-based triangulation is not practical enough. Therefore, in Step IV, Ruppert algorithm (Shewchuk, 2002) is used to refine the original mesh structure T_i. The basis of the refinement algorithm is to preserve triangulation as Delaunay by adding some vertices to reach a high-quality triangulation, which is theoretically validated and practically satisfying for two-dimensional triangulated meshes. This algorithm finds low-quality triangle t_i^j and remove them from T_i during a forward-searching process, and it inserts a vertex in the circumcenter of the removed t_i^j. Then, the searching process continues to remove triangles that lose their Delaunay property due to inserting the new vertex. Finally, a new triangulation ($\acute{T}_i j$) is generated with the newly inserted vertices and vertices whose corresponding triangles have been removed, and $\acute{T}_i j$ is added to T_i (Shewchuk, 2002). One of techniques to determine the quality of triangles is the ratio of the smallest edge to radius of the circumcircle of the triangle (ω). By definition, the minimum internal angle of the triangle can be calculated using equation (4) (Cheng *et al.*, 2012):

$$\theta_{min} = \arcsin\left(\frac{1}{2\omega}\right), \tag{4}$$

To ensure termination of the Ruppert algorithm, the critical value $\theta_{\min} = 20.7°$ has been theoretically calculated. If θ_{min} for each triangle is smaller than 20.7°, the triangle should be refined by Ruppert algorithm (Ruppert, 1995). Figure 7 illustrates the refined meshes for a single fracture with two orthogonal intersections, and for three orthogonally intersected fractures.

Note that h_s is a user-defined key parameter in the present triangulation algorithm. This parameter controls the size of triangles and the precision of the problem. As h_s increases, the intersections whose length is smaller than h_s are practically reduced to a single point, but they will never be removed. Therefore, the connectivity pattern of fractures is maintained. With decreasing h_s, the precision of the solution increases. It is obvious that the computational costs will also increases due to increase in the number of triangles. Thus, determination of the optimized h_s is a critical issue and will be discussed in the sensitivity analysis section.

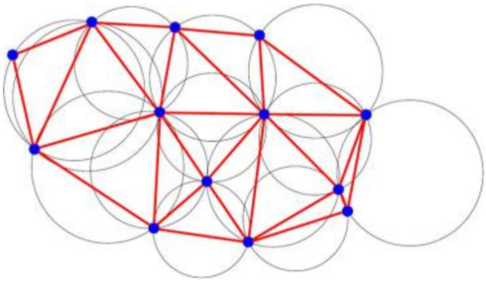


Figure 6.
Delaunay-based triangulation with representation of empty-circle criterion

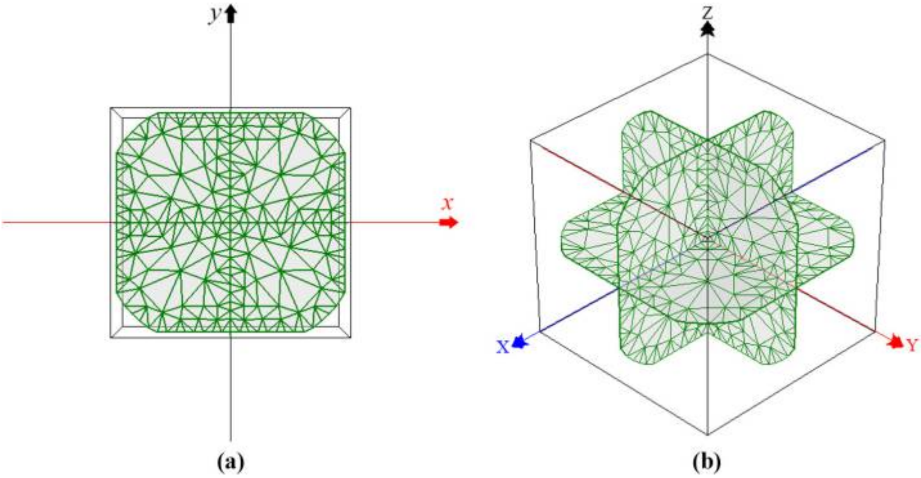


Figure 7.
Refined Delaunay triangulation for (a) a single fracture with two orthogonal intersections and (b) three intersecting fractures, generated by developed code in the present research

The present algorithm is able to generate high quality meshes for tightly spaced parallel plates and for the cases where fracture planes are intersected at small acute angles. To demonstrate the meshing quality, Figure 8 illustrated meshes of two fracture planes intersected with an acute angle of 10°, and they are all perpendicular to a third fracture plane. The intersection nodes (highlighted in red) near the center have been modified to avoid acute angle within the elements to ensure meshing quality.

3. Algorithm validation

In this section, validation of the present meshing algorithm and a series of sensitivity analyses on meshing parameters are discussed using three examples. Regular and simple geometrical structures are investigated in Examples I and II to compare outcomes of numerical flow calculations with analytical results to validate the present algorithm, and in Example III, a DFN is used to conduct the sensitivity analyses.

3.1 Flow numerical solution scheme

Below assumptions are considered in this research:

- Rock matrix is impermeable.
- The flow is in a steady state.
- Two walls of each fracture are planar, smooth and parallel.
- The flow model of fluid is Newtonian.

The planar flow rate is calculated on each f_i in (x, y) with a certain aperture a_{f_i}. It is assumed that $a_{f_i} \ll L_{f_i}$, where L_{f_i} is the length of f_i. In this research, a uniform PDF has been used to determine a_{f_i}. Depending on Poiseuille law, the permeability of a fracture (k_{f_i}) is obtained using equation (5) (Baca *et al.*, 1984):

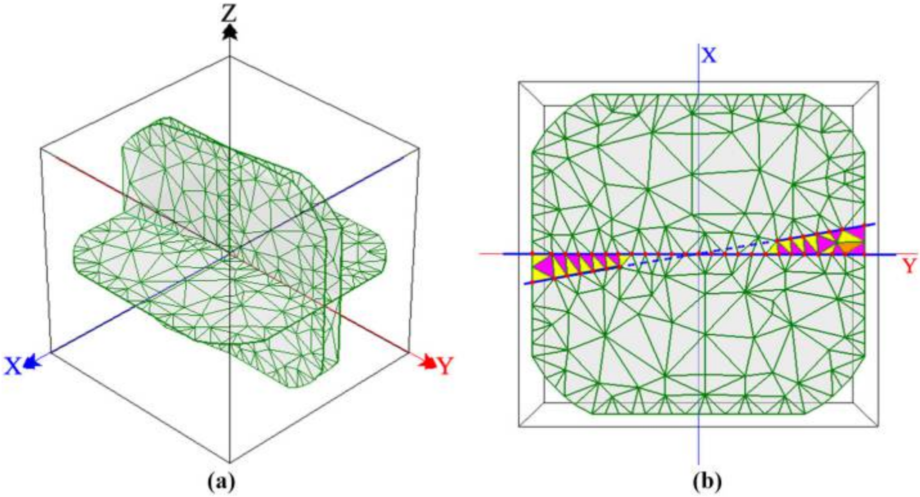


Figure 8.
Meshing structure of two acutely intersected fractures on the third fracture

Notes: (a) 3D view and (b) plan view. The triangles between traces have been differently colored and the intersection nodes have been marked in red

$$k_{f_i} = \frac{{a_{f_i}}^3}{12}. \tag{5}$$

As given in equation (6), the classic equations of Darcy and conservation of mass govern the fluid flow through fractured rock media (Koudina *et al.*, 1998):

$$\begin{cases} q_f = -\dfrac{1}{\mu} K_f \cdot \nabla p \\ \nabla q_f = 0 \end{cases}, \tag{6}$$

$$\nabla p = \rho \cdot g \cdot \nabla h. \tag{7}$$

In these equations, q_f is the average flow rate through fracture [m^2/s], K_f is the permeability matrix of fracture [m^3] assembled from individual fracture, ∇p is pressure gradient [Pa/m], ρ is the fluid density [kg/m^3], g is the gravitational acceleration [m/s^2] and ∇h is hydraulic head gradient.

Any standard boundary condition can be applied to this system of equations. The boundary conditions can be either Dirichlet or Neumann. It is assumed that Γ_D and Γ_N are parts of boundaries of DFN model with Dirichlet and Neumann boundary conditions, respectively; therefore, the boundary conditions are written as equation (8):

$$\begin{cases} h = h_D & on \quad \Gamma_D \\ q = q_N & on \quad \Gamma_N \end{cases}. \tag{8}$$

where, h_D and q_N are boundary conditions of the hydraulic head and flow rate. For each f_i, the permeability matrix $K_{f_i} \in \mathbb{R}^{N_{dof_i} \times N_{v_i}} \times \mathbb{R}^{N_{dof_i} \times N_{v_i}}$ is assembled according to the number of degrees of freedom (N_{dof_i}) and vertices (N_{v_i}) of fracture. Then, the vectors $q_{f_i} \in \mathbb{R}^{N_{dof_i} \times N_{v_i}}$ and $h_{f_i} \in \mathbb{R}^{N_{dof_i} \times N_{v_i}}$ are considered as the vectors of the flow rate and the hydraulic head respectively. As given in equations (9) to (11), the total permeability matrix (K) and total vectors of flow rate (q) and hydraulic head (h) for a DFN model are derived from union of transposed local K_{f_i}, q_{f_i} and h_{f_i}, respectively:

$$K = \begin{bmatrix} K_{11} & K_{12} & \cdots & K_{1N} \\ K_{21} & K_{22} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ K_{N1} & \cdots & \cdots & K_{NN} \end{bmatrix}, \tag{9}$$

$$q = \begin{pmatrix} q_1 \\ \vdots \\ \vdots \\ q_N \end{pmatrix}, \tag{10}$$

$$h = \begin{pmatrix} h_1 \\ \vdots \\ \vdots \\ h_N \end{pmatrix}, \tag{11}$$

where, $\acute{N} = N_{v_t} \times N_{dof_t}$ and N_{v_t} and N_{dof_t} are the total number of total vertices and degrees of freedom of the model, respectively. Amount of hydraulic head must be determined in all the vertices of the model. Therefore, the total number of equations of the system is equal to $\acute{N}$ which is calculated based on the flow equilibrium conditions. For the sake of validation of the present meshing algorithm and performing the sensitivity analyses, a developed code is implemented in c# using FEM scheme with a visual three-dimensional graphical user interface to show the results. The results are calculated using conjugate gradient (CG) method.

3.2 Example I

An example of a simple geometrical structure is provided to investigate the accuracy of the linear flow calculations and validate the meshing algorithm. The geometrical model Ω_1 includes a circular fracture with the center located at the global origin of coordinates and the radius of 5 m, which is enclosed with two parallel planar boundaries by spacing of 2 m. The hydraulic head on the upper boundary $H_1 = 1\ m$ and on the lower boundary is $H_2 = 0$. The permeability of the fracture is $k = 1$ m^2/s. The analytical solution of the total flow rate for a 2- × 10-m rectangular slab is equal to 5 m^3/s (Long *et al.*, 1985). In this numerical model, the summation of the flow rate of vertices located on one of the boundaries of the model is listed in Table II for a range of meshing sizes. The calculations demonstrate that the numerical results converge to 5 m^3/s with a decreasing meshing size, which is in a good agreement with the analytical results. The meshing structure and the diagram of hydraulic head distribution has been shown in Figure 9.

3.3 Example II

Shown in Figure 10(a), the example considers three orthogonal planar fractures embedded in a three-dimensional space. The simulation domain is a cube with side length of 100 m. Center of all three fractures are located at origin of the coordinates. Normal vectors of fractures are in the direction of *X*, *Y* and *Z* axes, respectively. The permeability of fractures is homogeneously equal to $8.172 \times 10^{-5}\ m^2/s$. Boundary conditions of the model have been demonstrated in Figure 10(a). Constant hydraulic heads of $H_1 = 1$ m and $H_2 = 0$ m are applied on the upper and lower boundaries of the model and a constant gradient of hydraulic head is prescribed on the side boundaries. The mesh structure and diagram of hydraulic

Table II.
Meshing size versus the numerical solution of the flow rate for Example I

Meshing size (m)	Flow rate (m^3/s)
0.2	4.909
0.18	4.930
0.16	4.927
0.14	4.942
0.12	4.925
0.1	4.971
0.08	4.954

head distribution have been shown in Figure 10(b) and 10(c). The head gradient along two vertical planes are constant. The total flow rate through the inflow or outflow face of the cube is given in Table III against the meshing size based on numerical solutions. The numerical solutions approach 1.630 × 10^{-5} m^3/s which completely agrees with the analytical solution of Long *et al.* (1985). Also note that the computed hydraulic head is constant (0.5 m) and the flow rate is zero across the horizontal fracture plane, indicating that no flow passes through that plane.

3.4 Example III

In this example, numerical simulation of the fluid flow has been performed for ten realizations of a DFN model ($\Omega_3^i\ ,\ i = 1, \cdots, 10$) with REV of 5 × 5 × 5 m. The numerical results for different realizations are averaged as the representative results. In this example, Ω_3^i is independently generated using geometrical statistical data given in Table IV, based on the technique illustrated in Section 2.1. The number of fractures and intersections of Ω_3^i have been listed in Table V. The density and viscosity of the fluid considered in this example are equal to 1,000 kg/m^3 and 0.001 *Pa·s*, respectively. The meshing parameter (h_s) varies to study its effect on the flow calculation. Altogether, 70 samples will be available to analyze the sensitivity of parameters. Figure 1 shows one of the realizations for which meshing structure and hydraulic head distribution have been illustrated in Figure 11.

For the sake of advancing the sensitivity analyses, it is necessary to determine an optimized h_s. Here, the critical case is evaluation of θ_{min} for different values of h_s. A rudimental survey showed that if θ_{min} is unchanged, and the termination of the meshing algorithm strongly affected as h_s changes. A larger θ_{min} generates a higher-quality triangulation, but choosing a too large θ_{min} for a small h_s can cause instability in the meshing algorithm (lack of termination) due to forming infinite loops to refine it. Therefore, it is important to select an optimized θ_{min} for any h_s to ensure the termination of the algorithm and to maintain the precision of the solution. In fact, θ_{min} is described as the value by which the algorithm can be terminated within a reasonable number of iterations. These results have been displayed in Figure 12, where, as h_s decreases, θ_{min} decreases as

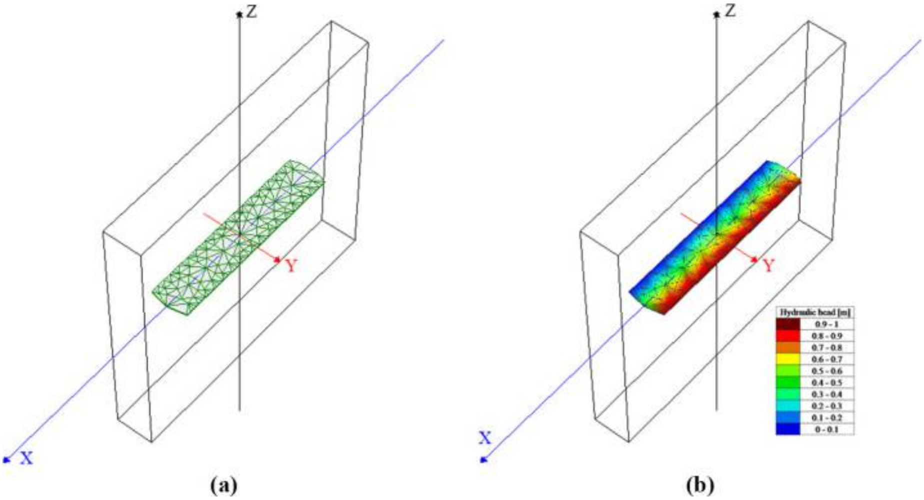
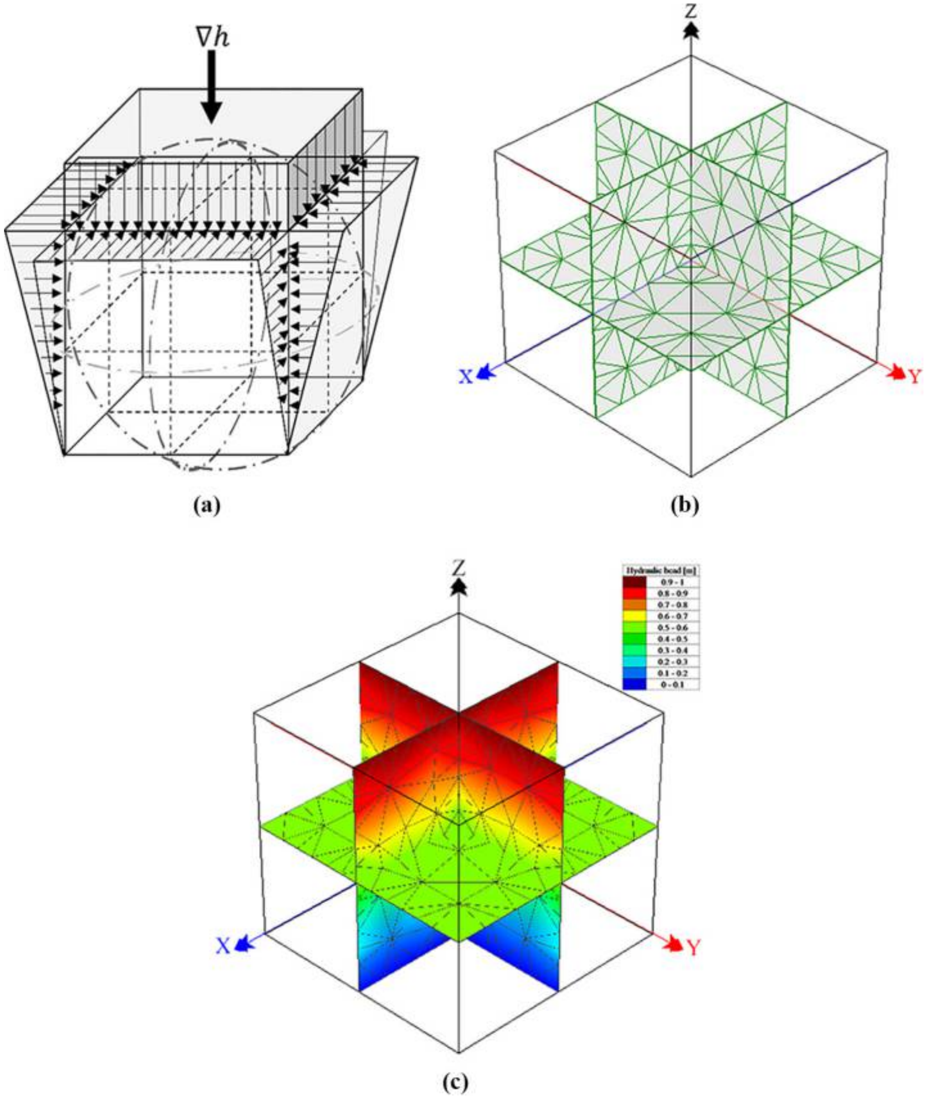


Figure 9.
(a) The meshing structure and (b) the diagram of hydraulic head for Example I



(a)

(b)

(c)

Figure 10.
(a) Boundary conditions of the hydraulic head, (b) the mesh structure and (c) the hydraulic head distribution for Example II

well. So, as an ordinary result, choosing a larger h_s can generate a higher-quality triangulation for $\mathbf{\Omega}_3^i$ in terms of minimum internal angle.

Diagrams of the total number of vertices (N_v) and the total number of triangles (N_t) of the meshing versus h_s have been depicted in Figure 13(a) and 13(b). However, decreasing trend seems obvious in this figure, achieving this power-law trend is indicative of success in triangulation process of $\mathbf{\Omega}_3^i$. Also, with increasing h_s, N_v and N_t are led to constant numbers. Earlier is the number of vertices from union of the intersection and the boundary vertices, and later is the number of Delaunay-based triangles generated by these vertices. Reduction of h_s can significantly increase the number of N_v and N_t and consequently the computational cost.

Figure 14 shows diagram of N_v versus N_t. The relation between these two parameters is approximately according to Equation (12) with a correlation factor of 0.95. Because of discrepancy in the connectivity pattern of fractures in different Ω_3^i, it is possible to generate a various number of triangles with a certain number of vertices:

$$N_v = 0.7552\ N_t + 145.53, \tag{12}$$

In this research, the average flux c is used to determine the precision of the solution [Equation (13)]:

$$c = \frac{\|\mathbf{q}_2\|}{N_{dof_t} \times N_{v_t}}, \tag{13}$$

Table III. Meshing size versus the numerical solution of the flow rate for Example II

Meshing size (m)	Flow rate (m^3/s)
10	1.623×10^{-5}
9	1.630×10^{-5}
8	1.629×10^{-5}
7	1.634×10^{-5}
6	1.628×10^{-5}
5	1.633×10^{-5}
4	1.630×10^{-5}

Table IV. Geometrical statistical data of each joint set

	Dip [Deg]	Dip direction [Deg]		Density [1/m^3]	Length [m]			Aperture [mm]	
	Uniform	Fisher		Poisson	Power-law			Uniform	
Joint set	Average	Average	k	Average	Max	Min	α	Min	Max
1	70	40	45	0.2	1.78	1	10	4	12
2	30	20	135	0.12	1.78	1	10	4	12
3	80	40	135	0.1	1.78	1	10	4	12
4	45	20	315	0.15	1.78	1	10	4	12

Table V. The number of fractures and their intersections for each realization

Realization no.	Joints no.	Intersections no.
1	125	313
2	115	271
3	119	331
4	130	361
5	114	324
6	117	304
7	109	269
8	122	303
9	114	270
10	111	313

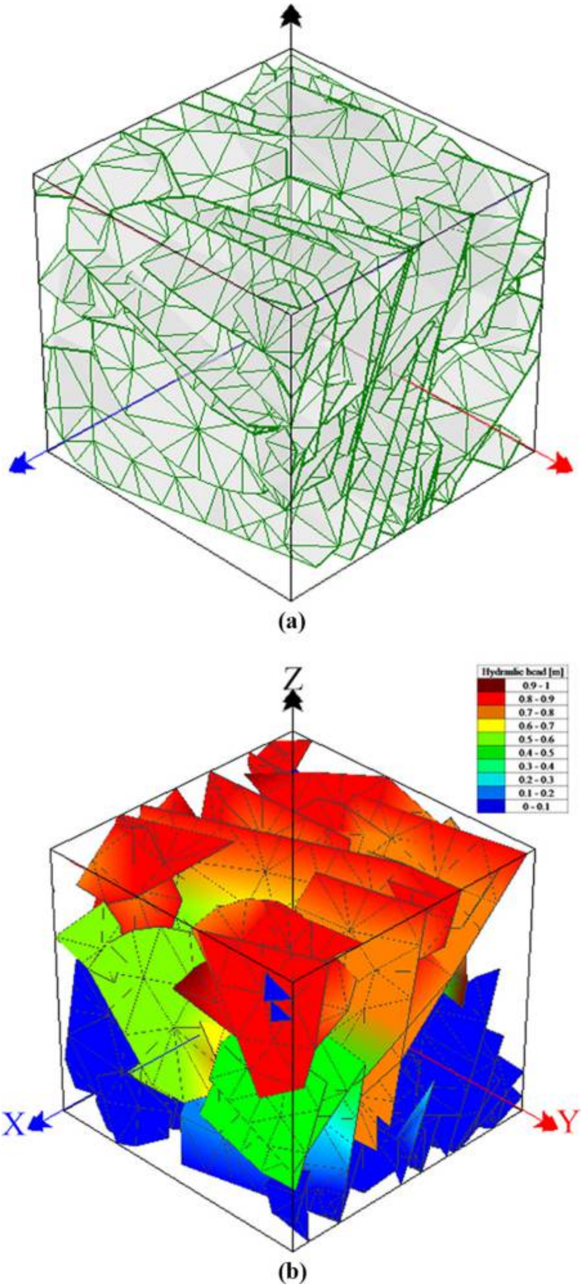


Figure 11.
(a) Mesh structure and (b) the hydraulic head distribution for DFN in Example III

where N_{dof_t} is number of degrees of freedom and N_{v_t} is number of vertices. $\boldsymbol{q}$ is vector of flow rate for all vertices and all degrees of freedom, i.e. $\boldsymbol{q} = \{q_x^1\ ,\ q_y^1,\ q_z^1,\ q_x^2\ldots\}$), and $\|\boldsymbol{q}\|_2$ is the L$_2$-norm of the vector. Equation (13) can be considered as averaged flow rate for whole the REV, and it is used as a benchmark to compare solution of the model for different meshing sizes.

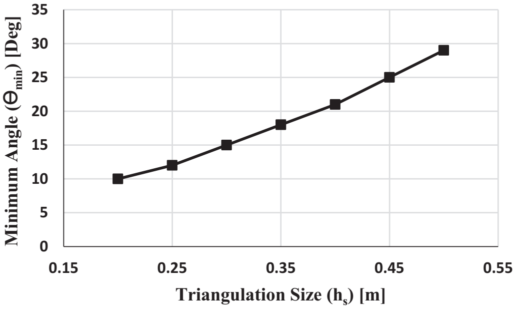
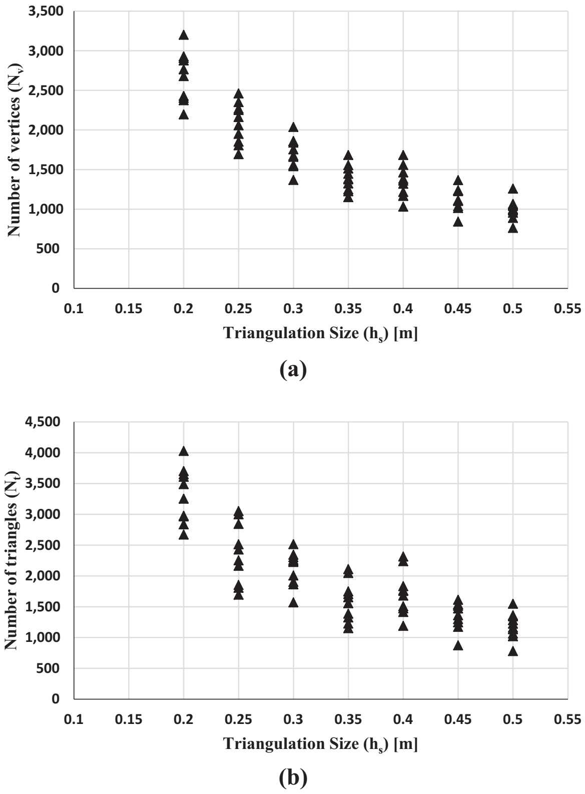


Figure 12. Diagram of the minimum internal angle of triangles (θ min) versus the meshing size (h_s)



(a)

(b)

Figure 13. (a) Total number of vertices (N_v) and (b) the total number of triangles (N_t) versus the meshing size (h_s)

Diagrams of c against h_s for all realizations are shown in Figures 15. The trend demonstrated that the calculated average flow field is rather constant for all mesh sizes. However, we do experience un-convergence if mesh size $h_s > 0.3\,m$, while it seems that the convergence of the numerical solution can be reached within a reasonable number of iterations for $h_s \leq 0.3$ m for all the realizations. Therefore, as a secondary result, choosing a smaller h_s can computationally be more convenient.

In Figure 16(a) and 16(b), diagrams of runtime of the meshing algorithm and the solution scheme against h_s are depicted, respectively. In fact, N_v represents the total number of variables and the consequently the number of equations of the model for each degree of freedom. All the calculations have been done using the same hardware system. Since both variations of the runtime of meshing and the runtime of the solution scheme have power-law trends relative to h_s, use of a small h_s can significantly increase the runtime of calculations. Also, diagram of the solution and triangulation runtimes versus N_v has been shown in Figure 16(c). According to this figure, the trend of both the

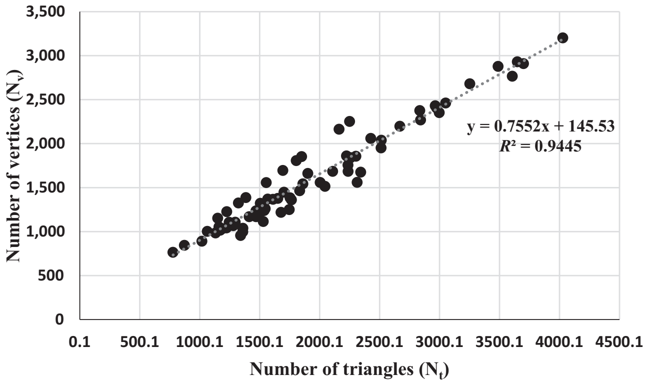
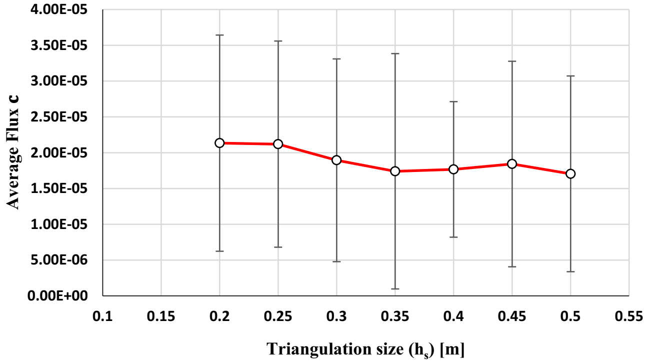


Figure 14.
Diagram of the number of vertices (N_v) versus the number of triangles (N_t) of the meshing



Note: Dots shows the median value, and bars show the standard deviation

Figure 15.
Diagram of average flux c versus the meshing size hs for 10 realizations

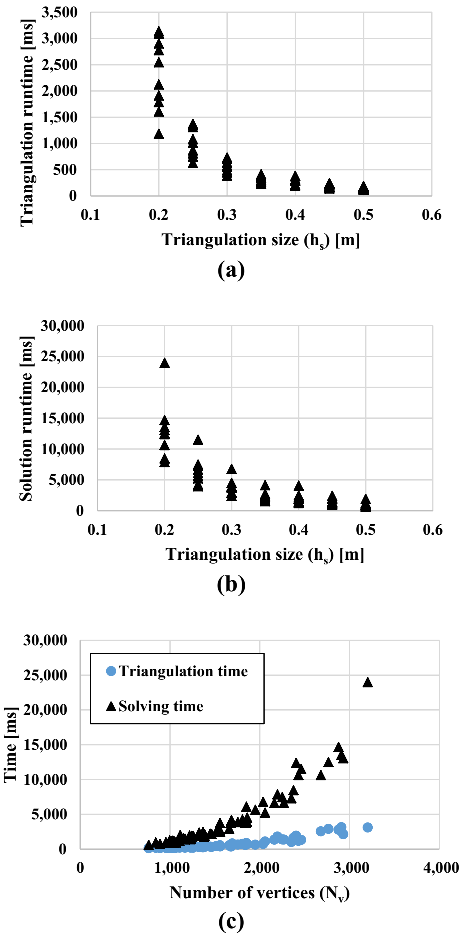
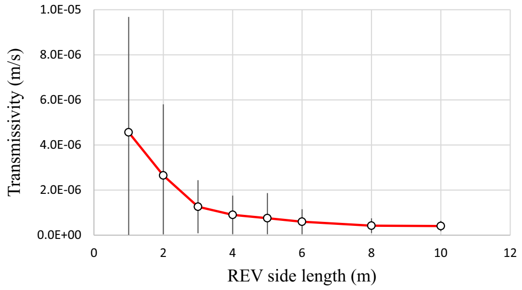


Figure 16.
Diagrams of the meshing size (h_s) versus (a) the meshing runtime, (b) the numerical solution runtime and (c) the diagram of both the runtimes versus the number of the vertices of the model



Note: 20 simulations for each REV size, the bars show plus and minus one standard deviation

Figure 17.
Diagrams of the average flow rate versus REV side length

runtimes follow a power-law and the triangulation runtime is about 25 per cent of solution runtime for all cases.

To understand the effect of REV size on the flow rate solution, sensitivity analysis is performed by increasing the REV side length. Twenty simulations for REV size are conducted. Figure 17 shows the transmissivity solution (i.e. flux per cross section area) converges to a constant value when the side length of REV is greater than 5 m. For this case, the variability of the solution, shown as the bars in Figure 17, also significantly decreases. For this reason, an REV side length of 5 m is used in the sensitivity study of meshing parameters.

Based on results of Figures 12, 13, 15 and 16, it seems that increasing triangular size could increase minimum angle of the mesh, reduce number of triangles, and eventually improve the numerical stability. On the other hand, it may deteriorate the accuracy of numerical solution. In this study, h_s = 0.2-0.25 m is recommended as an optimized meshing size for this particular example. Note that the optimal h_s value may depend on the problem dimension and network connectivity. Sensitivity analyses as demonstrated in this example should be conducted to obtain the optimal h_s for meshing 3D DFN models.

4. Conclusion

In this research, a new Delaunay-based meshing algorithm for triangulation of geometrical structure of three-dimensional discrete fracture network has been developed. This algorithm enjoys benefits of high precision and fast speed, while maintaining connectivity pattern of the fracture network. Also, the process of generating discrete fractures, removing isolated and dead-end fractures, avoiding acute intersection angles have been described. The present algorithm has been validated through comparison with analytical results. A series of sensitivity analyses have been conducted to determine the effect of meshing parameters on the flow and to illustrate performance of the algorithm. It is shown that the meshing size is a key parameter in the present algorithm; therefore, a sensitivity analysis shows how to evaluate an optimized meshing size to ensure accuracy and convergence of the algorithm.

References

Baca, R., Arnett, R. and Langford, D. (1984), "Modelling fluid flow in fractured-porous rock masses by finite-element techniques", *International Journal for Numerical Methods in Fluids*, Vol. 4 No. 4, pp. 337-348.

Baghbanan, A. and Jing, L. (2008), "Stress effects on permeability in a fractured rock mass with correlated fracture length and aperture", *International Journal of Rock Mechanics and Mining Sciences*, Vol. 45 No. 8, pp. 1320-1334.

Benedetto, M.F., Berrone, S. and Scialò, S. (2016), "A globally conforming method for solving flow in discrete fracture networks using the virtual element method", *Finite Elements in Analysis and Design*, Vol. 109, pp. 23-36.

Cheng, S.-W., Dey, T.K. and Shewchuk, J. (2012), *Delaunay Mesh Generation*, CRC Press, Boca Raton, FL.

Dershowitz, W. and Einstein, H. (1988), "Characterizing rock joint geometry with joint system models", *Rock Mechanics and Rock Engineering*, Vol. 21 No. 1, pp. 21-51.

Elsworth, D. (1986), "A hybrid boundary element-finite element analysis procedure for fluid flow simulation in fractured rock masses", *International Journal for Numerical and Analytical Methods in Geomechanics*, Vol. 10 No. 6, pp. 569-584.

Erhel, J., De Dreuzy, J.-R. and Poirriez, B. (2009), "Flow simulation in three-dimensional discrete fracture networks", *SIAM Journal on Scientific Computing*, Vol. 31 No. 4, pp. 2688-2705.

Gallager, R.G. (2012), *Discrete Stochastic Processes*, Springer Science & Business Media, Berlin.

Han, X., Chen, J., Wang, Q., Li, Y., Zhang, W. and Yu, T. (2016), “A 3D fracture network model for the undisturbed rock mass at the Songta dam site based on small samples”, *Rock Mechanics and Rock Engineering*, Vol. 49 No. 2, p. 611.

Hautefeuille, M., Melnyk, S., Colliat, J.B. and Ibrahimbegovich, A. (2009), “Failure model for heterogeneous structures using structured meshes and accounting for probability aspects”, *Engineering Computations*, Vol. 26 Nos 1/2, pp. 166-184.

Hernández, A., Albizuri, J., Ajuria, M.B.G. and Hormaza, M.V. (1997), “An adaptive meshing automatic scheme based on the strain energy density function”, *Engineering Computation*, Vol. 14 No. 6, pp. 604-629.

Hyman, J.D., Gable, C.W., Painter, S.L. and Makedonska, N. (2014), “Conforming delaunay triangulation of stochastically generated three dimensional discrete fracture networks: a feature rejection algorithm for meshing strategy”, *SIAM Journal on Scientific Computing*, Vol. 36 No. 4, pp. A1871-A1894.

Jin, F., Zhang, C.H., Wang, G. and Wang, G.L. (2003), “Creep modeling in excavation analysis of a high rock slope”, *Journal of Geotechnical and Geoenvironmental Engineering*, Vol. 129 No. 9, pp. 849-857.

Jing, L., Ma, Y. and Fang, Z. (2001), “Modeling of fluid flow and solid deformation for fractured rocks with discontinuous deformation analysis (DDA) method”, *International Journal of Rock Mechanics and Mining Sciences*, Vol. 38 No. 3, pp. 343-355.

Karimi-Fard, M. and Firoozabadi, A. (2003), “Numerical simulation of water injection in fractured media using the discrete fractured model and the Galerkin method”, *SPEREE, SPE Reservoir Evaluation and Engineering*, Vol. 6 No. 2, pp. 117-126, SPE-83633-PA, doi: 10.2118/83633-PA.

Karimi-Fard, M., Durlofsky, L.J. and Aziz, K. (2003), “An efficient discrete fracture model applicable for general purpose reservoir simulators”, *SPE Reservoir Simulation Symposium*, Society of Petroleum Engineers, Dallas.

Koudina, N., Garcia, R.G., Thovert, J.-F. and Adler, P. (1998), “Permeability of three-dimensional fracture networks”, *Physical Review E*, Vol. 57 No. 4, p. 4466.

Li, S., Xu, Z., Ma, G. and Yang, W. (2014), “An adaptive mesh refinement method for a medium with discrete fracture network: the enriched persson’s method”, *Finite Elements in Analysis and Design*, Vol. 86, pp. 41-50.

Long, J., Gilmour, P. and Witherspoon, P.A. (1985), “A model for steady fluid flow in random three-dimensional networks of disc-shaped fractures”, *Water Resources Research*, Vol. 21 No. 8, pp. 1105-1115.

Mackerle, J. (2001), “2D and 3D finite element meshing and remeshing: a bibliography (1990-2001)” *Engineering Computations*, Vol. 18 No. 8, pp. 1108-1197.

Maryška, J., Severýn, O. and Vohralík, M. (2005), “Numerical simulation of fracture flow with a mixed-hybrid FEM stochastic discrete fracture network model”, *Computational Geosciences*, Vol. 8 No. 3, pp. 217-234.

Mohajerani, S., Baghbanan, A., Wang, G. and Forouhaandeh, S.F. (2017), “An efficient algorithm for simulating grout propagation in 2D discrete fracture networks”, *International Journal of Rock Mechanics and Mining Sciences*, Vol. 98, pp. 67-77.

Moradi, M., Shamloo, A., Asadbegi, M. and Dezfuli, A.D. (2017), “Three dimensional pressure transient behavior study in stress sensitive reservoirs”, *Journal of Petroleum Science and Engineering*, Vol. 152, pp. 204-211.

Mustapha, H. and Mustapha, K. (2007), “A new approach to simulating flow in discrete fracture networks with an optimized mesh”, *SIAM Journal on Scientific Computing*, Vol. 29 No. 4, pp. 1439-1459.

Mustapha, H., Dimitrakopoulos, R., Graf, T. and Firoozabadi, A. (2011), “An efficient method for discretizing 3D fractured media for subsurface flow and transport simulations”, *International Journal for Numerical Methods in Fluids*, Vol. 67 No. 5, pp. 651-670.

Olson, J.E. (1993), "Joint pattern development: effects of subcritical crack growth and mechanical crack interaction", *Journal of Geophysical Research: Solid Earth*, Vol. 98 No. B7, pp. 2251-2265.

Parashar, R. and Reeves, R.G. (2012), "On iterative techniques for computing flow in large two-dimensional discrete fracture networks", *Journal of Computational and Applied Mathematics*, Vol. 236 No. 18, pp. 4712-4724.

Ren, F., Ma, G., Wang, Y., Li, T. and Zhu, H. (2017), "Unified pipe network method for simulation of water flow in fractured porous rock", *Journal of Hydrology*, Vol. 547, pp. 80-96.

Ruppert, J. (1995), "A delaunay refinement algorithm for quality 2-dimensional mesh generation", *Journal of Algorithms*, Vol. 18 No. 3, pp. 548-585.

Rutqvist, J., Leung, C., Hoch, A., Wang, Y. and Wang, Z. (2013), "Linked multicontinuum and crack tensor approach for modeling of coupled geomechanics fluid flow and transport in fractured rock", *Journal of Rock Mechanics and Geotechnical Engineering*, Vol. 5 No. 1, pp. 18-31.

Shewchuk, J.R. (2002), "Delaunay refinement algorithms for triangular mesh generation", *Computational Geometry*, Vol. 22 Nos 1/3, pp. 21-74.

Tsang, Y. (1992), "Usage of 'equivalent apertures' for rock fractures as derived from hydraulic and tracer tests", *Water Resources Research*, Vol. 28 No. 5, pp. 1451-1455.

Vermilye, J.M. and Scholz, C.H. (1995), "Relation between vein length and aperture", *Journal of Structural Geology*, Vol. 17 No. 3, pp. 423-434.

Wang, M., Kulatilake, P., Panda, B. and Rucker, M. (2001), "Groundwater resources evaluation case study via discrete fracture flow modeling", *Engineering Geology*, Vol. 62 No. 4, pp. 267-291.

Wang, K., Peng, X., Du, Z., Haghighi, M. and Yu, L. (2016), "An improved grid generation approach for discrete fracture network modelling using line fracture concept for two-phase flow simulation", *SPE Asia Pacific Oil & Gas Conference and Exhibition, Society of Petroleum Engineers*, Dallas.

Wu, Q., Kulatilake, P. and Tang, H.M. (2011), "Comparison of rock discontinuity mean trace length and density estimation methods using discontinuity data from an outcrop in Wenchuan area, China", *Computers& Geotechnics*, Vol. 38 No. 2, pp. 258-268.

Xie, Y. and Wang, G. (2014), "A stabilized iterative scheme for coupled hydro-mechanical systems using reproducing kernel particle method", *International Journal for Numerical Methods in Engineering*, Vol. 99 No. 11, pp. 819-843.

Xu, C. and Dowd, P. (2010), "A new computer code for discrete fracture network modelling", *Computers and Geosciences*, Vol. 36 No. 3, pp. 292-301.

Xue, Y. (2017), "Numerical simulation of coal deformation and gas flow properties around borehole", *CMES: Computer Modeling in Engineering and Sciences*, Vol. 113 No. 4, pp. 429-441.

Ye, J.H., Huang, D. and Wang, G. (2016), "Nonlinear simulation of offshore breakwater on sloping liquefied seabed", *Bulletin of Engineering Geology and the Environment*, Vol. 75 No. 3, pp. 1215-1225.

Zhang, Q.H. (2015), "Finite element generation of arbitrary 3-D fracture networks for flow analysis in complicated discrete fracture networks", *Journal of Hydrology*, Vol. 529, pp. 890-908.

Zhang, Q.H. and Yin, J.M. (2014), "Solution of two key issues in arbitrary three-dimensional discrete fracture network flow models", *Journal of Hydrology*, Vol. 514, pp. 281-296.

Corresponding author
Duruo Huang can be contacted at: huangdr@ust.hk