Comput Geosci (2011) 15:385–397
DOI 10.1007/s10596-010-9210-6

ORIGINAL PAPER

G23FM: a tool for meshing complex geological media

Hussein Mustapha

Received: 26 August 2009 / Accepted: 10 September 2010 / Published online: 25 September 2010

Abstract We present G23FM, a mesh generation tool for discretizing two- and three-dimensional complex fractured geological media. G23FM includes different techniques to generate finite element grids that maintain the geometric integrity of input surfaces, and geologic data and produce optimal triangular/tetrahedral grids for flow and transport simulations. G23FM generates grid for two-dimensional cross-sections, represents faults and fractures, for three-dimensional fractured media, and has the capability of including finer grids. Different examples are presented to illustrate some of the main features of G23FM.

Keywords Mesh generation · Fractured media · 2D/3D complex geometry · Mesh and geometry adaptations

1 Introduction

Given the spatial heterogeneity of geological media and interrelation of effective flow properties, numerical modeling is an important contribution to the understanding and forecasting of flow behavior, and management of groundwater. In addition, numerical modeling assists with the integration of multi-scale geological heterogeneity, simulates the hydraulic flow and transport phenomena, and quantifies uncertainty due to the availability of sparse data. Geological fractured media contain highly complex geologic configurations and are multi-scale heterogeneous [1–5, 12, 14–17, 19, 20]. The existence of a large number of fractures with strong variations in geometric and physical proprieties is the origin of the complexity. A significant effect on multiphase flow and solute transport may be caused by fractures in rock formations [6, 11–13]. An efficient and accurate discretization of these fractures within the matrix is required to account for the natural complexity of fractured media and the physics of matrix–fracture interaction.

Figure 1 represents a fracture network generated stochastically using real data. The network is very complex as shown in the configurations of both the network (Fig. 1, left) and the fracture (Fig. 1, right). Meshing these complex structures is a real challenge.

Fracture disctretization or mesh generation refers to generating solid finite element meshes representing the heterogeneous media; those are used to solve various engineering problems based on finite element methods. Moreover, generating a quality mesh is equally important as generating a mesh for multiphase flow problem simulations [16–18].

The motivations to this work come from different points: (1) develop a flexible and easy to use mesh generator fully controlled by the users, (2) handle complex media, and (3) optimizing the mesh by modifying slightly complex and critical configurations.

The last point is realized by introducing different techniques to analyze the major complexities at the fracture as at the network scale. Various methods

The code will be available at: http://cosmo.mcgill.ca/.

H. Mustapha (✉)
Department of Mining and Materials Engineering,
McGill University, Montreal, Canada
e-mail: Hussein.Mustapha@mcgill.ca,
HMustapha.math@gmail.com

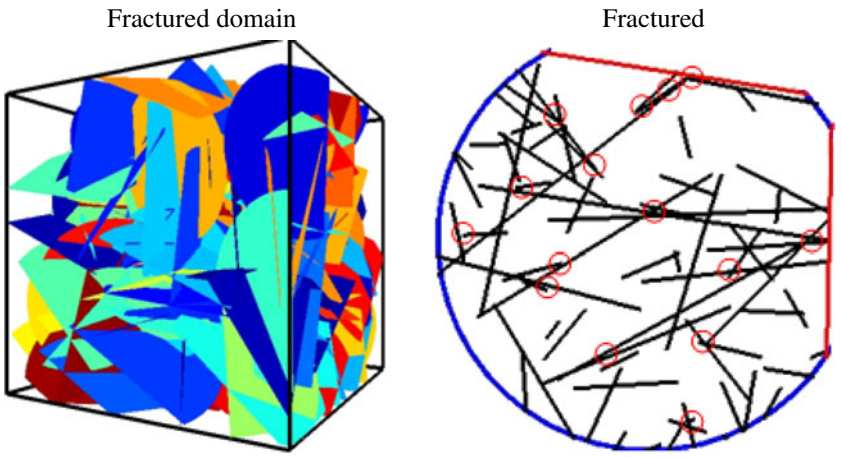

Fig. 1 Example of fracture network (*left*) which contains 500 fractures and a fracture is extracted on the *right*

are developed in the literature to simplify complex configurations created by randomly generated fractures. A simplification procedure using 2D/3D structured grid is used by [14, 15]. A global projection over a 2D/3D unstructured grid is discussed in [9, 16]; recently, a novel method based on a local transformation of the fractures is developed for 2D problems in [16], and extended to 3D problems in [17]. These methods will briefly be presented in this paper; however, for more details and validations, we refer to [16, 17] and the references therein.

This paper presents G23FM or "**G**rid generator for **2**D/**3**D **F**ractured **M**edia" which is a tool for automatically producing unstructured finite element grids tuned to the special needs of geologic and geo-engineering applications. G23FM (1) includes different options based on the methods discussed above and they can be used for flow and transport simulations in porous and fractured media; (2) produces two-dimensional grid for each fracture using triangles, and these two-dimensional grids are connected to have a complete mesh for the 3D fractured network; (3) has the capability to adapt the grid generation for complex geometries; (4) is modular, allowing for flexibility and consistency; (5) can easily incorporate input constraints change into the computational mesh, (6) can also be used to produce coarsened grids for preliminary calculations and refined grids for final, high resolution calculations; and (7) is very flexible, can be used by any numerical algorithm that can utilize unstructured grids and is not specific to any particular computational code.

The paper is organized as follow: Section 2 discusses different procedures for simplifying complex fractured media. G23FM description, input file parameters, and different options of the program are given in Section 3. Section 4 illustrates different examples. Conclusions follow.

2 Numerical model

Stochastically generated fractures [1–4, 14–17] lead to very complex geometrical structures. The complexity mainly arises from the large number of fractures contained in a fractured media, and the corresponding critical configurations generated.

For the reason of simplicity, two fractures are presented in Fig. 2; the angle between the two fractures is 22° and can be close to 0° for other cases. Discretizing Fig. 2 with a reasonable number of triangles may lead to degenerate triangles between fractures. Then, small grids may be included to improve the quality of the triangles; however, this solution is computationally costly. Reducing the complexity of the problem by treating the critical configurations is another solution; this section presents three different methods to simplify slightly the geometry of the fractures instead of generating an adaptive grid. A brief description of each method, for representing the fractures in Fig. 2, is given next.

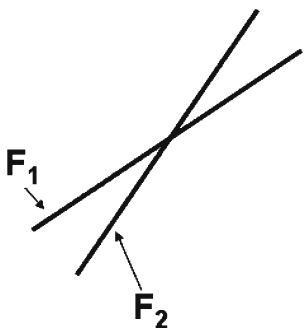

Fig. 2 An example of a critical configuration generated by two fractures

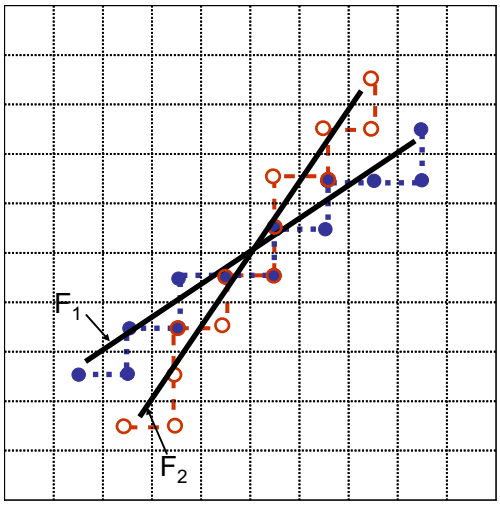

Fig. 3 Representation of the fractures configuration shown in Fig. 2 by the square centers using the simplification procedure SP1

2.1 Simplification procedure 1 using 2D/3D structured grid

Based on a two-dimensional grid, the fractures in Fig. 2 are approximated by some square centers as shown in Fig. 3. In 3D, the fracture boundaries (i.e., the contour and the fracture intersections) are approximated by some cube centers of a regular three-dimensional grid as shown in Fig. 4.

Using this method, (1) most of the degenerate triangles are removed, and (2) the connectivity of the network is respected by the fact that two intersections of fractures are intersected if and only if they have a joint cube. Also, this method ensures a minimal distance and angles between two successive cubes belonging to the discretized object. The method makes the grid optimal in the sense that no mesh refinement procedure is needed to improve the quality because all complex configurations are analyzed. However, the fractures approximations or the flow paths are longer than the original paths. This may influence the results by, for example, underestimating of solute concentrations [9, 16]. Moreover, the mesh cannot be refined at a given location because the transformation made is global and based on only one general structured grid. For more details, we refer to [14–16].

2.2 Simplification procedure 2 using 2D/3D unstructured grid

In references [9] and [16], a discretization method of irregular fractures in 3D for flow and transport simulations is discussed. The technique presented is based on the approximation of the fractures using two-dimensional (for 2D problems) and three-dimensional (for 3D problems) finite element grids that utilize triangles and hexahedral elements, respectively. For a better understanding of this method, the 2D version of the simplification procedure 2 (SP2) is summarized as follow:

1. Discretize a fracture by a series of segments.
2. Generate a 2D finite elements grid using triangular elements.
3. Find intersections of all fracture segments with the triangular element edges.
4. Move the element edge intersections to closet nodes. These nodes become the fracture nodes.
5. Reconstruct the fracture using the corresponding fracture nodes found in 4.

Using the simplification procedure SP2, the fractures in Fig. 2 are transformed as shown in Fig. 5. Both simplification procedures simplification procedure 1 (SP1) and SP2 provide identical results for flow problems in 2D/3D as shown in [9, 16]; however, the simplification procedure SP2 is better than the simplification procedure SP1 for 2D/3D transport simulations.

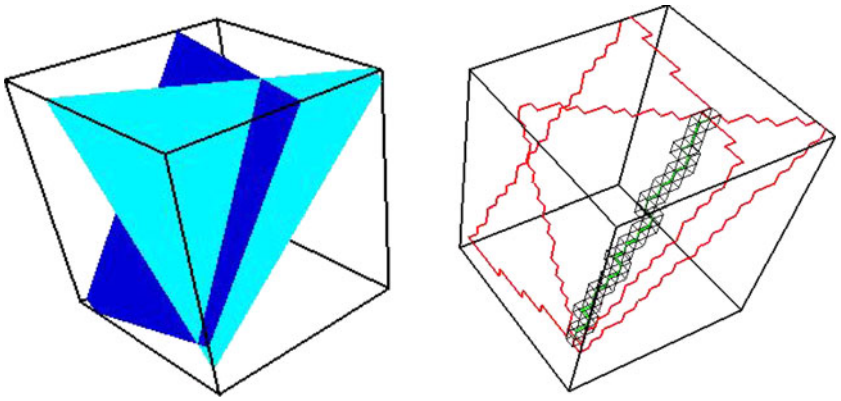

Fig. 4 Two-fracture configuration in 3D (*left*); the projection of the boundaries of the fractures and their intersections on a three-dimensional regular grid (*right*) using simplification procedure SP1

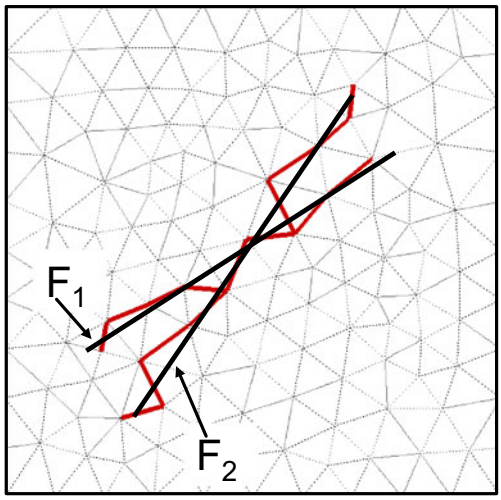

Fig. 5 Representation of the fractures configuration shown in Fig. 2 by the *pecked lines* using the simplification procedure SP2

A major drawback of the simplification procedure SP2 is that it does not allow for the inclusion of a refinement procedure to locally adapt the mesh [16] because, as explained for the simplification procedure SP1, a global transformation of the fractures is used. Computationally and numerically, this may have big effects for large-scale problems and when the fracture sizes vary strongly. In addition, existence of clusters of fractures inside a large domain may only require finer grids in these clusters and coarse grids away. Furthermore and depending on the fractures properties, i.e., size, aperture, or orientation, a fracture may need to have finer grids than others, and in conclusion, the simplification procedure SP2 may not offer these opportunities.

2.3 Simplification procedure 3 using local transformation

A novel simplification procedure, SP3, using a local transformation of the fractures F_1 and F_2 in Fig. 2 is introduced in [16, 17]. The method SP3 can solve the problems arising from the simplification procedures SP1 and SP2 for very complex problems. The concept used by the simplification procedure SP3 is based on reducing locally the critical and complex configurations, as much as possible, by remaining part of the fracture network unchangeable. The method developed is realized in three steps:

1. Step 1: The boundaries of the domain and the fractures are discretized into set of discrete points.
2. Step 2: The discretized points obtained in Step 1 are analyzed by scanning the original image with a square of size h'. This procedure starts with the square centered at the lower left corner of the domain.
   By default $h' = h$ where h is the mesh step. In this case, the minimal distance allowed between two discrete points will be h'.
3. Steps 3: The scanning starts with the square defined in Step 2. The square displays horizontally and vertically until scanning the entire domain. At each time, we search the discrete points, obtained from Step 1, that are within the current square. Different cases are distinguished:
   a. No points are inside the square; then move to another square.
   b. Only one point from Step 1 is inside the square; then, this point is not changed as for example the border's points.
   c. More than one point from Step 1 are inside the elementary square; then, all the points inside the square are moved to the center of the square. Note that the remaining points are not moved.

The simplification described above may change slightly the orientations and lengths of few segments. Consequently, fractures lengths will be changed slightly. This simplification is followed by triangulating the domain using a constrained Delaunay triangulation method [16]. Based on the discrete points resulting from Step 3, a Delaunay algorithm is implemented to generate the internal points as shown in Fig. 6. For more details about triangulations, we refer to [8, 16]. Finally, we note that the algorithm developed is flexible and allows for

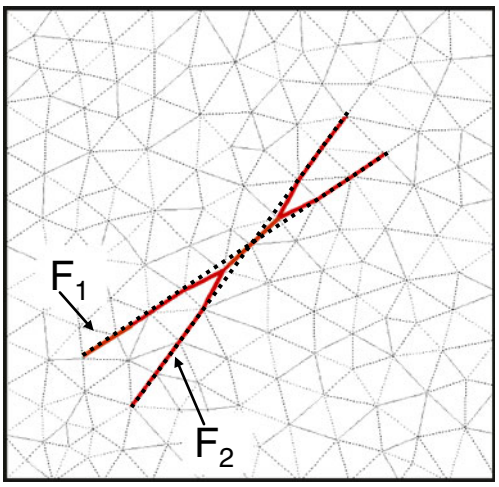

Fig. 6 Representation of the fractures configuration shown in Fig. 2 by the *dotted lines* using the simplification procedure SP3

scanning the image using different geometric shapes, such as triangles or polygons, and not only the squares used here.

The approach developed is extended to 3D in [17]. Complex 3D fracture configurations are simplified to facilitate the generation of a mesh quality without a refinement procedure to improve the mesh quality of complex fractured configurations. In that work, the 3D geometry of the fracture network is adapted instead of the 3D mesh. The simplified procedure (SP3) delivered an optimal solution in terms of solution precision, mesh quality, and computational cost.

```
% Geom: ellipsoid or a rectangle. 1: ellipsoid; 2: rectangle.
% Geom
2
%
% (x0,z0) : coordinates (m) of the left bottom point of the
% rectangular domain (reference point).
% x0,z0
0. 0.
%
% Lx : length(m) of the rectangular domain in the x-direction
% positive to the right.
% Lz : height(m) or width(m) of the rectangular domain in the
% z-direction positive upward or y-direction positive uptward.
% Lx, Lz
8. 8.
%
% Nx : number of subdivisions in the x-direction.
% Nz : number of subdivisions in the z-direction.
% Nx, Nz
8. 8.
%
% nf : number of fractures in the domain.
% nf
2
%
% fi: fracture index (number).
% (x1,y1), (x2,y2) : coordinates (m) of the fracture element ends.
% N_i : number a priori of nodes on the fracture number i.
% fi, x1, y1, x2, y2, N_fi for i = 1, nf
1 1 1 6 6 6
2 4 4.1 7 2 5
%
% scp : scaling parameter (less then 1) for the plotting.
% The unit of the axis that supports the larger of Lx and Lz
% is multiplied by scp. If scp = 1, then no scaling is done.
% scp
1
% mod : option to modify the fractures configurations to avoid very
% small triangular elements.
% They are two parameters used in the modification: ssi and pni.
% If mod = 0, there is no modification and the parameters ssi and pni are not used.
% ssi : maximum length (m) for a fracture to be removed.
% pni : maximum length (m) to be added to a fracture.
% mod, ssi, pni
1 0.2 0.2
%
% nbmax: maximum number of mesh nodes in the domain.
% nbmax
6000
% add: option to generate nodes close to the fractures: 1 = yes , 0 = no.
% When add = 0, no nodes are added and the parameters hap and htf are not used.
% htf: the ratio of the height (ht) of a triangle to the fracture mesh element (Lb).
% hap: radius of the circle with center at the added node.
% add, htf, hap
0 0 0
%<method> <param>
%method: the method used to treat the fractures. Different methods are used:
%method=0: no treatment of the fractures; here param is a mot used.
%method=1: SP1 is used; then param is the size of the regular grid block (Figure 13).
%method=2: SP2 is used; here param is the mesh step i.e. h (Figure 14).
%method=3: SP3 is used; then param is the size of the square in 2D or the cube in 3D used to
%scan the discrete points obtained from Step 1 of the simplification procedure SP3.
0 0
```

Fig. 7 Example of a two-dimensional input data file to generate Fig. 6

3 G23FM general algorithm

G23FM reads only one input file "*name_file.in*" written with a slight difference with respect to the dimension of the problem, the geometry of the domain, and the methods used. G23FM generates one output file that contains the mesh information.

The source code is written in standard Fortran 90 using modules (classes). G23FM can be compiled using *nmake* command in Windows operating machine, and contains three modules: (1) ***mod_initialization***; (2) ***mod_g23dm***, and (3) ***mod_visualisation***. They are described by the following.

3.1 mod_initialization

In this module, we provide the data to define the geometry. The data are stored in input data file "*name_file.in*". This file provides data necessary to create the mesh. The format of this file and relevant illustrations are presented in the following examples and Figs. 7, 8, 9, 10, 11, 12, and 13.

3.1.1 Two-dimensional problem

For two-dimensional problems, the format of the input file and relevant illustrations are presented in the following example and Fig. 7. It contains:

- ***Geom***: the two-dimensional domain is an ellipsoid or a rectangle. 1: ellipsoid; 2: rectangle.

If "rectangle":

- <x_0> <y_0>
  - (x_0,y_0): coordinates (***m***) of the left bottom point of the rectangular domain (reference point).

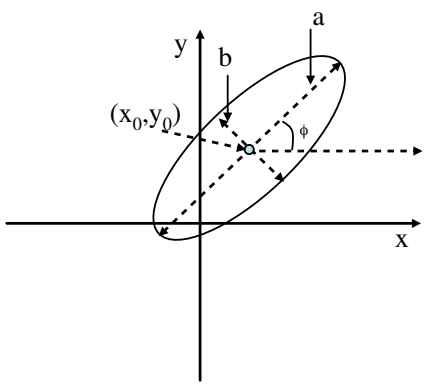

Fig. 8 Ellipsoid parameters illustration

- <***Lx***> <***Ly***>
  - ***Lx***: length (***m***) of the rectangular domain in the x-direction positive to the east.
  - ***Ly***: width (***m***) or height (***m***) of the rectangular domain in the *y*-direction positive to the north or the *z*-direction positive upward.
- <***Nx***> <***Ny***>
  - ***Nx***: number of subdivisions in the ***x***-direction.
  - ***Ny***: number of subdivisions in the ***y***- or ***z***-direction.

If "ellipsoid":

- <x_0> <y_0>
  - (x_0,y_0): coordinates (*m*) of the ellipsoid center (reference point).
- <***a***> <***b***> <ϕ>
  - ***a***, ***b*** and ϕ: ellipsoid parameters as shown in Fig. 8.
- <***Ne***>
  - ***Ne***: number of subdivisions of the ellipsoid's border.

The following parameters are the same for both geometries:

- <***nf***>
  - ***nf***: number of fractures in the domain.
- <***fi***> <x_1> <y_1> <x_2> <y_2> <N_{fi}>
  - ***fi***: fracture index (number).
  - (x_1,y_1), (x_2,y_2): coordinates (***m***) of the fracture element ends.
  - N_{fi}: number of nodes in the fracture number ***fi***.
- <***scp***>
  - ***scp***: scaling parameter (less then 1) for the plotting. The unit of the axis that supports the larger of ***Lx*** and ***Ly*** or ***Lz*** is multiplied by ***scp***. If ***scp*** = 1, then no scaling is performed.
- <***nbmax***>
  - ***nbmax***: maximum number of nodes in the domain.
- <***mod***> <***ssi***> <***pni***>
  - ***mod***: option to modify the fractures configurations to avoid very small triangular elements (Fig. 9). There are two parameters used in the modification: ***ssi*** and ***pni***. If

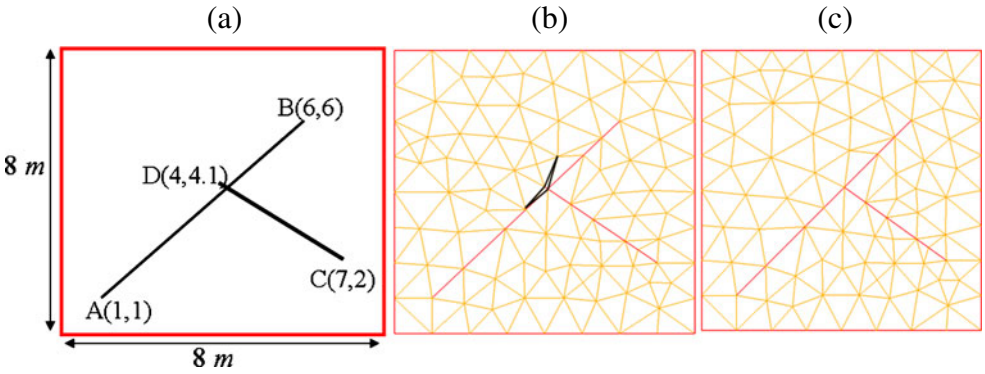

Fig. 9 **a** A domain with two intersecting fractures; **b** ***mod*** $= 0$. Mesh with two flat triangles around D; **c** ***mod*** $= 1$. Remove the flat triangles from the mesh

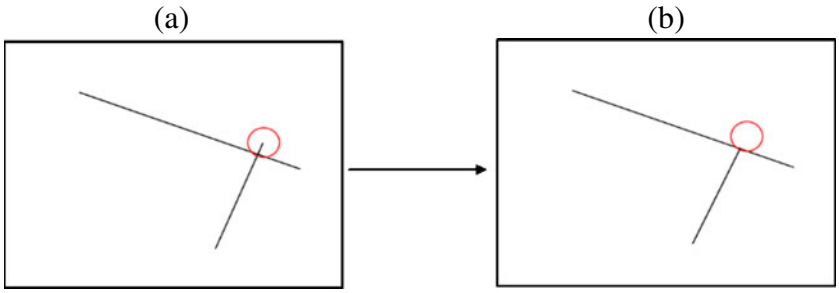

Fig. 10 Two intersecting fractures. The small segment in the circle on the left (**a**) is removed; see (**b**)

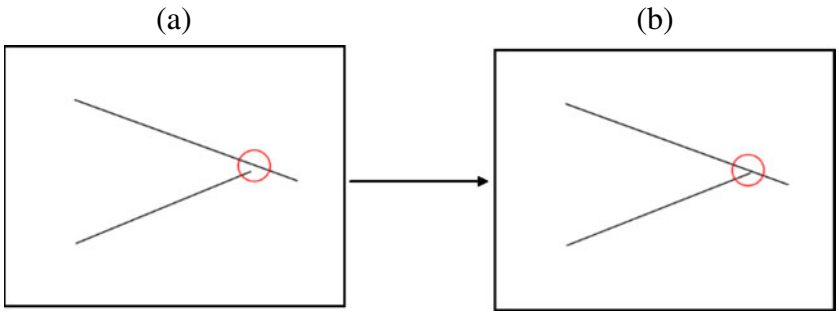

Fig. 11 When the fractures become close (**a**), one is replaced by the configuration shown in (**b**)

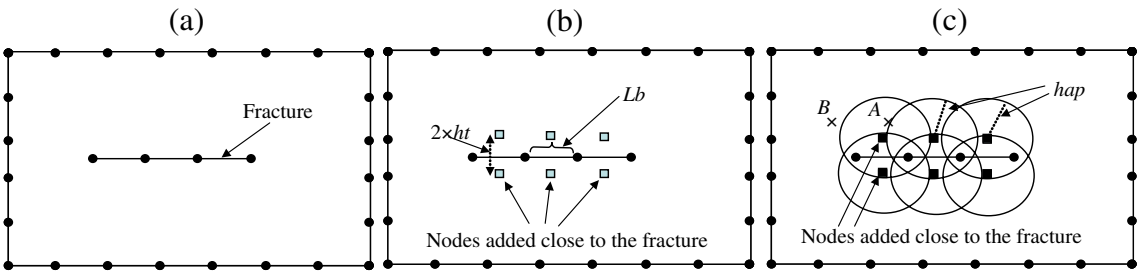

Fig. 12 **a** ***add*** $= 0$; **b** and **c** ***add*** $= 1$. Nodes are added around the fracture

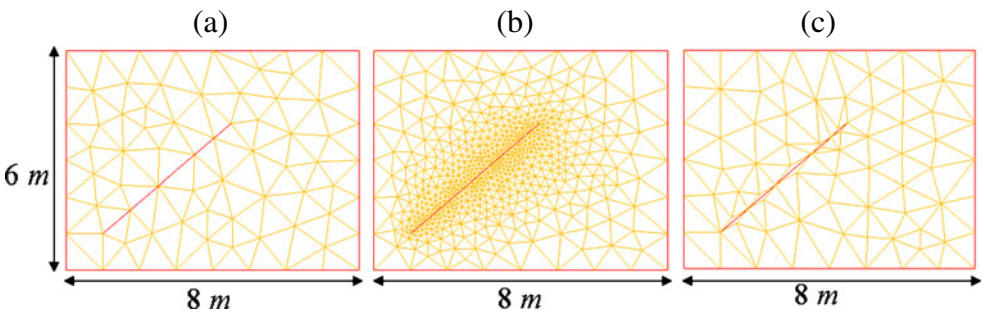

Fig. 13 ***nf*** $= 1$, ***Nx*** $= 8$, ***Nz*** $= 6$, ***mod*** $= 0$. **a** $N_1 = 5$, ***add*** $= 0$, ***nn*** $= 76$; **b** $N_1 = 50$, ***add*** $= 0$, ***nn*** $= 643$; **c** $N_1 = 5$, ***add*** $= 1$, ***nn*** $= 77$

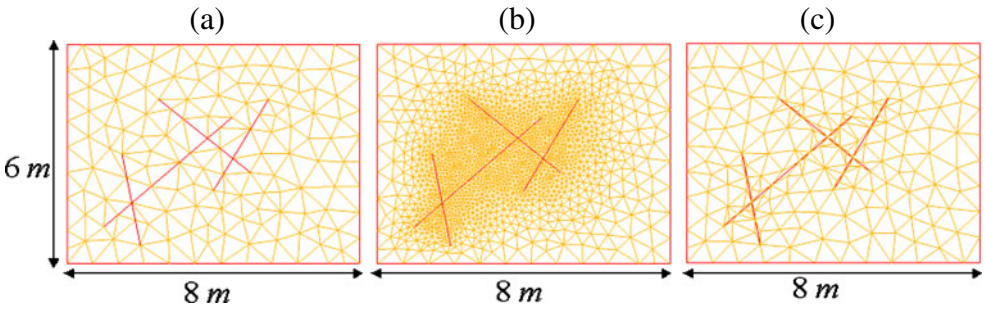

Fig. 14 ***nf*** $= 4$, ***Nx*** $= 12$, ***Ny*** $= 9$, ***mod*** $= 1$. The corresponding data listed in Table 1

mod $= 0$, there is no modification and the parameters ***ssi*** and ***pni*** are not used.

- ***ssi***: maximum length (***m***) for a fracture to be removed (Fig. 10).
- ***pni***: maximum length (***m***) to be added to a fracture (Fig. 11).

- <***add***> <***htf***> <***hap***>
  - ***add***: option to generate nodes close to the fractures (Fig. 12): 1 = yes, 0 = no.
    - If ***add*** $= 0$, no nodes are added and the parameters hap and htf are not used.
    - If ***add*** $= 1$, we add the nodes in two steps. In first step, we add nodes around the fractures (Fig. 12b) and in second step we create the other nodes of the mesh. The nodes added in the second step must be outside the circles with centers the nodes added in first step (Fig. 12c). For example, in Fig. 12c, ***A*** is unacceptable added node and ***B*** is acceptable. The nodes are added in second step using a Delaunay method [8].
  - ***htf***: the ratio of the height (***ht***) of a triangle to the fracture mesh element (***Lb***) (Fig. 12b).
  - ***hap***: radius of the circle with center at the added node in first step of mesh generation nodes (Fig. 12c).

These options are very useful in some applications like, for example, the use of the cross-flow equilibrium concept [10]. This concept combines the fractures cells and two adjacent triangles in a volume control cell. This procedure needs a finer grid close to the fracture. Using these options, the program generates small triangles close to the fractures without need to call for a refinement procedure. Thus, the number of triangles does not increase and approximately the same triangles are deformed to have this characteristic.

Table 1 The corresponding data used for the generation of Fig. 14

	N_1	N_2	N_3	N_4	*add*	*nn*
a	8	4	6	5	0	152
b	56	28	42	35	0	3,208
c	8	4	6	5	1	182

- <***method***> <***param***>
  - ***method***: the method used to treat the fractures. Different methods are used:
    - **method = 0:** no treatment of the fractures; here ***param*** is a not used.
    - **method = 1:** SP1 is used; then ***param*** is the size of the regular grid block (Fig. 13).
    - **method = 2:** SP2 is used; here ***param*** is the mesh step, i.e., h (Fig. 14).
    - **method = 3:** SP3 is used; then ***param*** is the size of the square in 2D or the cube in 3D used to scan the discrete points obtained from Step 1 of the simplification procedure SP3.

Note that if **method** $\neq 0$, then parameters <***mod***>, <***ssi***>, <***pni***>, <***add***>, <***htf***>, and <***hap***> are not used.

3.1.2 Three-dimensional problem

In the case of a three-dimensional problem, the domain is a parallelepiped (for example, the fracture network in Fig. 1) and the input data file is described as follows:

- <x_0> <y_0><z_0>
  - (x_0,y_0,z_0): coordinates (***m***) of the left bottom point of the of the parallelepiped domain (reference point).

- <***Lx***> <***Ly***><***Lz***>
  - ***Lx***: length (***m***) of the rectangular domain in the x-direction positive to the east.
  - ***Ly***: width (***m***) of the parallelepiped domain in y-direction positive to the north.
  - ***Lz***: height (***m***) of the rectangular domain in the z-direction positive upward.

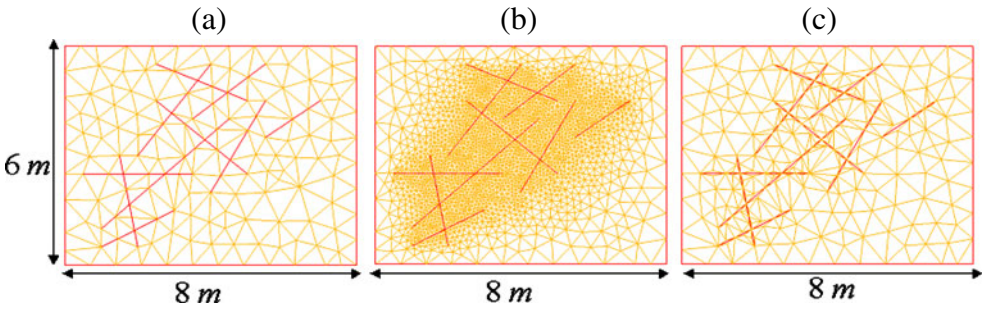

Fig. 15 $nf = 10$, $Nx = 12$, $Ny = 9$, $mod = 1$. The corresponding data listed in Table 2

- <*nf*>
  - ***nf***: number of fractures in the domain.
- <*fi*> <*x_fi*> <*y_fi*> <*z_fi*> <*a_fi*> <*b_fi*> <ϕ*_fi*> <ψ*_fi*> <θ*_fi*><h_{fi}>
  - ***fi***: fracture index (number).
  - (***x_fi***,***y_fi***,***z_fi***): coordinates (***m***) of the fracture's (i.e., the ellipsoid) center.
  - ***a_fi*** and ***b_fi***: large and small axis of the ellipsoid as shown in Fig. 8.
  - ϕ***_fi***, ψ***_fi***, and θ***_fi***: Euler angles (***phi***, ***psi***, and ***theta***) in degree used for the ellipsoid orientation. If these angles are set to zero, then fracture is in x–y plane and oriented as shown in Fig. 8b.
  - h_{fi}: mesh step used to discretize the fractures' border and intersections. Note that the mesh of the network is a conforming mesh and the mesh step is the same to disctretize all the intersections between the fractures of the network.
- <***nbmax***>
  - ***nbmax***: maximum number of nodes in the domain. Note that the number of nodes in the mesh of a given fracture can be estimated by calculating the ratio of the fracture's area to the size of an equilateral triangle with an edge size equal to h_{fi}.
- <***method***> <***param***>
  - Same as for 2D problems with extension of the simplification procedures to 3D.

Note that the options <***mod***>, <***ssi***>, <***pni***>, <***add***>, <***htf***>, and <***hap***> are not used because if an intersection is moved, prolonged, or shortened then, the same operation has to make in the corresponding intersected fractures.

3.2 Mod_g23dm

This module contains three main subroutines:

- ***fractures_intersections***: after reading the fractures characteristics, this subroutine calculates the intersections between the fractures. These intersections are segments for three-dimensional problems (i.e., the black lines in the fracture in Fig. 1 (right)) and are points for two-dimensional problems (i.e., two lines intersect by a point).
- ***network_discretization***: this subroutine discretizes the fractures into points. This global discretization provides a similar discretization of a given intersection in all the fractures containing it. Then, the fractures meshes communicate together through the similar discretized intersections to form a global conforming mesh for the whole domain. Here, the results are a set of nodes, connected by edges.
- ***g23dm***: this subroutine uses the set of points and edges from the previous subroutine to generate the internal nodes and the final mesh. In two-dimension, the Delaunay algorithm is applied to generate one triangulation based on the previous set of points and edges. In three-dimension, this algorithm is applied to each fracture given its discretized points and edges. The internal nodes and edges in the resulting mesh are numbered locally and the intersections nodes and edges have local and global numbers. When the triangulations of all

Table 2 The corresponding data used for the generation of Fig. 15

	N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8	N_9	N_{10}	*add*	*nn*
a	4	6	9	6	5	6	3	8	4	5	0	158
b	28	42	63	42	35	42	21	56	28	35	0	2,628
c	4	6	9	6	5	6	3	8	4	5	1	275

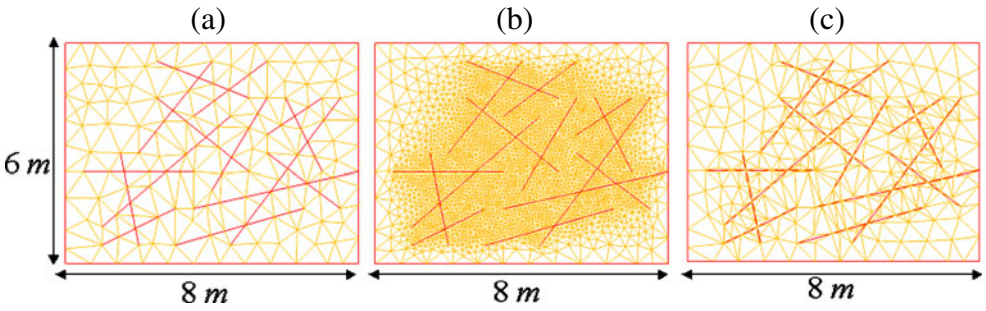

Fig. 16 $nf = 15$, $Nx = 12$, $Ny = 9$, $mod = 1$. The corresponding data listed in Table 3

fractures are obtained, a renumbering procedure is called to provide global numbers to all the internal nodes and edges of the fractures' triangulations.

3.3 Mod_visualization

This module contains one subroutine to write the output data in a file ("*name_file.mesh*"); it can be used with the finite element method, and can be visualized using Medit software [7].

4 Numerical examples

We present several examples with different level of complexity to illustrate the main features of G23FM. From sample configurations of one fracture to more complex configurations with thousands of fractures are meshed. All runs were performed on a 3.2 Ghz Intel(R) Xeon (TM) PC with 2 GB of RAM.

4.1 Example 1: a demonstration of G23FM performances

Denote by ***nn*** the number of nodes in the mesh of the two- and three-dimensional domains. Figures 12 to 15 present the meshing for different number of fractures. For example, in Fig. 12, we have a discrete fractured media with one fracture ($nf = 1$).

The fracture and the domain's boundaries are discretized by the same mesh step, and the mesh around the fractures is not refined. The mesh obtained is of good quality. Also, GFM offers an option to refine the grid close to the fractures as shown, for example, in Fig. 12b. For another application, we may need finer grids near the fractures, and coarser grids far from the fractures. The solution to this is presented in Fig. 12c. Such mesh helps to improve the precision of the solution (using for example the cross-flow-equilibrium concept [10] and to reduce the memory capacity and the CPU time).

In Figs. 14 to 16, we present three examples of fractured media with, respectively, 4, 10, and 15 fractures. For the figures, we chose the following input parameters:

- $Geom = 2$; $(x_0, y_0) = (0,0)$; $Lx = 8\ m$ and $Ly = 6\ m$; $scp = 1$; if $mod \neq 0$, then $ssi = 0.00001\ m$, $pni = 0.00001\ m$; $nbmax = 6{,}000$; if $add \neq 0$, then $htf = 10\ m$ and $hap = 0.5\ m$; $method = 0$.

In Fig. 9, we present a simple example for modification of fractures configuration. The following input has been chosen:

- $Geom = 2$; $(x_0, y_0) = (0,0)$; $Lx = 8\ m$ and $Ly = 8\ m$; $nf = 2$; $Nx = 8$; $Ny = 8$; $N_1 = 7$; $N_2 = 6$; $scp = 1$; if $mod \neq 0$, then $ssi = 0.2\ m$, $pni = 0.2\ m$; $nbmax = 6000$; $add = 0$; $method = 0$.

Figure 9a shows the initial domain. The point D is close to the fracture AB. First, the mesh of the domain with an option $mod = 0$ is generated as shown in Fig. 6b. Two flat triangular elements are generated around D. These triangles are removed on in Fig. 6c when the

Table 3 The corresponding data used for the generation of Fig. 16

	N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8	N_9	N_{10}	N_{11}	N_{12}	N_{13}	N_{14}	N_{15}	*add*	*nn*
a	4	6	9	6	5	6	3	8	4	5	5	10	4	9	6	0	167
b	28	42	63	42	35	42	21	56	28	35	35	70	28	63	42	0	3,611
c	4	6	9	6	5	6	3	8	4	5	5	10	4	9	6	1	355

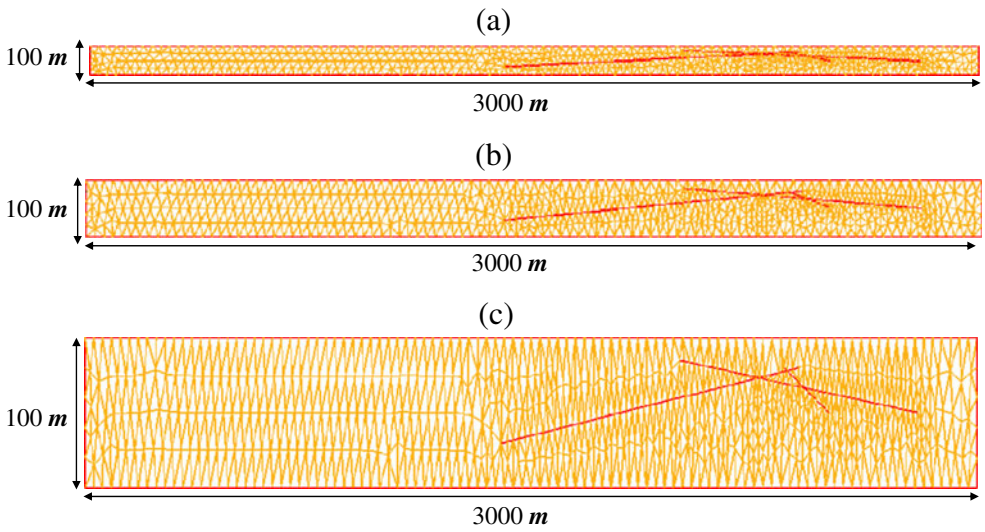

Fig. 17 Scaling for plotting. ***nn*** = 1,510. **a** ***scp*** = 1; **b** ***scp*** = 0.5; **c** ***scp*** = 0.2

modification option is activated, i.e., ***mod*** = 1. Note that, any of the simplification procedures is used in the examples for the reason of simplicity.

Figure 17 shows the usability of the scaling option. Different examples, for various values of ***scp***, are presented. For this example, we have only changed ***scp*** and we selected the following data:

- ***Geom*** = 2; $(\boldsymbol{x_0},\boldsymbol{y_0})$ = (0,0); ***Lx*** = 3000 *m* and ***Ly*** = 100 *m*; ***nf*** = 3; ***Nx*** = 100; ***Ny*** = 5; $\boldsymbol{N}_1$ = 50; $\boldsymbol{N}_2$

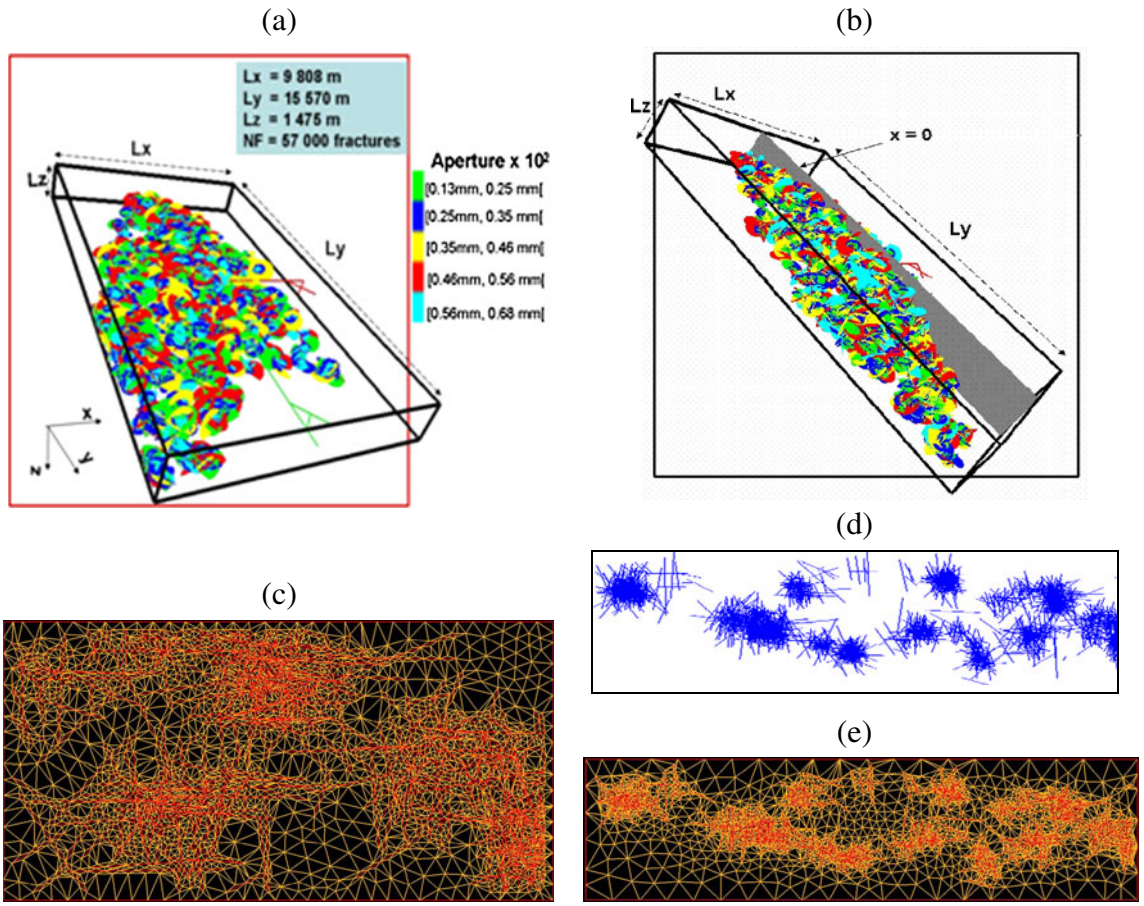

Fig. 18 **a** A fracture network contains 57,000 fractures generated in a parallelepiped of (9,808 *m*) × (15,570 *m*) × (1,475 *m*); **b** a cross-section ($x = 0$) of the network; **c** the mesh of the zone ($z = 0$ and $y > 0$) from (**a**); **d** and **e** are cross-section at $x = 0$ and its mesh, respectively

$= 50$; $\boldsymbol{N}_3 = 10$; ***mod*** $= 1$; ***ssi*** $= 0.00001\ m$; ***pni*** $= 0.00001\ m$; ***nbmax*** $= 6{,}000$; ***add*** $= 0$; ***method*** $= 0$.

4.2 Example 2: meshing a real complex fractured media

An example of meshing real complex geometry is shown in Fig. 18. Figure 18a consists of a complex fractured media that contains 57,000 fractures, stochastically generated. A cross-section at $x = 0$ is presented in Fig. 18b and d. The two-dimensional sections of the network show some clusters of fractures. G23FM adapts the mesh in a consistent and economic way as shown in Fig. 18c and e. These figures show the ability of controlling the grids size to generate different gridding resolutions depending of the position of each cluster. Note that the simplification procedure SP3 is used in this example.

An elementary criterion has been adopted and consists of the evaluation of the element quality with respect to the equilateral simplex (as the best possible element). For a triangular element, the quality is expressed as [8, 16, 17]

$$q = \alpha \frac{A}{a^2 + b^2 + c^2}, \tag{1}$$

where A is the area of the triangle, a, b, and c are the lengths of its sides and is a normalizing coefficient which justifies the quality of an equilateral triangle to 1. Figure 19 shows the mesh quality plots for both the original domain mesh and the mesh after using the simplification procedure SP3. This figure shows clearly how the degenerate triangles in the original domain gridding are removed from the gridding after using SP3. Most of the triangles obtained using the simplification procedure SP3 are of high quality, in contrary to the original mesh as shown by the important number of degenerate triangles in the range [0,0.6] of Fig. 19.

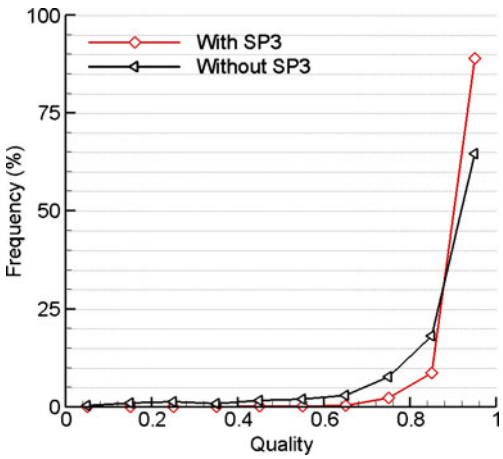

Fig. 19 Comparison between the mesh quality of the fractured domain before and after using the simplification procedure SP3

4.3 Conclusions

A mesh generation tool, G23FM, is presented in this paper. Different simplification procedures (i.e., SP1, SP2, and SP3) of complex fractured media are discussed. The simplification procedure SP1 is a simple concept based on an approximation of the fractures using a regular grid; this concept is mostly suitable for flow problems. For transport problems, the simplification procedure SP2 performs better than the simplification procedure SP1, and is based on a global transformation of the fractures using an unstructured grid. Note that for orthogonal fractures, SP1 is better than SP2 because the fractures coincide with the regular grid used. For complex fractured media, an efficient simplification procedure, SP3, is presented; SP3 uses a local transformation to make the minimal needed modifications. This transformation is decided by scanning the original domain and defining the critical configurations based on given criteria. The simplification procedure SP3

- removes the degenerate triangles and provides a good mesh quality
- maintains approximately the geometric integrity of the input surfaces and geological data; consequently, very close configurations are obtained after removing the critical complexities
- allows the inclusion of a refinement procedure, for example, to locally adapt the mesh of complex configurations of fractured clusters
- is typical in the sense that the solution obtained is exactly the "*perfect solution*", when no critical configurations are found

G23FM is a flexible tool and contains various options like adding points, refinement, controlling triangles' sizes, scaling, and other useful options. G23FM is fully controlled by the users as shown in different examples. Complex three-dimensional networks have been meshed and the mesh obtained after using the simplification procedure SP3 is shown to be much better the mesh before using simplification procedure SP3. Complex real fracture networks are mesh in this paper. Finally, the grids produced by G23FM are widely applicable and can be used by any numerical algorithm that can utilize unstructured grids.

Acknowledgements The support of University of Rennes, IRISA/INRIA, and McGill University is gratefully acknowledged. The authors also thank the members of RERI Institute, Palo Alto, California, Prof. Roussos Dimitrakopoulos and Prof. Thomas Graf for their support.

References

1. Berkowitz, B.: Characterizing flow and transport in fractured geological media: a review. Adv. Water Resour. **25**, 861–884 (2002)
2. Berkowitz, B., Bour, O., Davy, P., Odling, N.: Scaling of fracture connectivity in geological formations. Geophys. Res. Lett. **27**, 2061–2064 (2000)
3. Bonnet, E., Bour, O., Odling, N.E., Davy, P., Main, I., Cowie, P., et al.: Scaling of fracture systems in geological media. Rev. Geophys. **39**, 347–383 (2001)
4. Bour, O., Davy, P., Darcel, C., Odling, N.: A statistical scaling model for fracture network geometry, with validation on a multiscale mapping of a joint network (Hornelen Basin, Norway). J. Geophys. Res. **107**(B6), 2113 (2002)
5. Desbarats, A.J., Dimitrakopoulos, R.: Geostatistical modeling of transmissibility for two-dimensional reservoir studies. SPE Form. Eval. **5**, 437–443 (1990)
6. Dowd, P.A., Xu, C., Mardia, K., Fowell, R.J.: A comparison of methods for the stochastic simulation of rock fractures. Math. Geol. **39**, 697–714 (2007)
7. Frey, P.: MEDIT: An Interactive Mesh Visualization Software, p. 44. User's manual (2004)
8. Frey, P.J., George, P.L.: Mesh Generation: Application to Finite Elements, p. 816. Hermes Science, Oxford (2000)
9. Graf, T., Therrien, R.: A method to discretize non-planar fractures for 3D subsurface flow and transport simulations. Int. J. Numer. Methods Fluids **56**, 2069–2090 (2008)
10. Hoteit, H., Firoozabadi, A.: Numerical modeling of diffusion in fractured media for gas injection and recycling schemes. SPE J. **14**(2), 323–337 (2009)
11. Karimi-Fard, M., Firoozabadi, A.: Numerical simulation of water injection in 2D fractured media using discrete fracture model. SPE Reserv. Evalu. Eng. **4**, 117–126 (2003)
12. Matsuura, T., Tezuka, K., Tamagawa, T.: Naturally fractured reservoir modeling using static and dynamic data. Sekiyu Gijutsu Kyokaishi Journal **68**(6), 479–488 (2003)
13. Michael, S., Riley, M.: An algorithm for generating rock fracture patterns: mathematical analysis. Math. Geol. **36**, 683–702 (2004)
14. Mustapha, H.: Simulation numérique de l'écoulement dans des milieux fracturés tridimensionnels. Thèse de Doctorat de l'Universite de Rennes, France (2005)
15. Mustapha, H., Mustapha, K.: A new approach to simulating flow in discrete fracture networks with an optimized mesh. SIAM J. Sci. Comput. **29**(4), 1439–1459 (2007)
16. Mustapha, H., Dimitrakopoulos, R.: Discretizing two-dimensional complex fractured fields for incompressible two-phase flow. Int. J. Numer. Methods Fluids (2009). doi:10.1002/fld.2197
17. Mustapha, H., Dimitrakopoulos, R., Graf, T., Firoozabadi, A.: An efficient method for discretizing 3D fractured media for subsurface flow and transport simulations. Int. J. Numer. Methods Fluids (2010). doi:10.1002/fld.2383
18. Reichenberger, V., Jakobs, H., Bastian, P., Helmig, R.: A mixed-dimensional finite volume method for two-phase flow in fractured porous media. Adv. Water Resour. **29**, 1020–1036 (2006)
19. Silliman, S.E., Berkowitz, B.: The impact of biased sampling on the estimation of the semivariogram within fractured media containing multiple fracture sets 1. Math. Geol. **32**, 543–560 (2000)
20. Maryška, J., Severýn, O., Vohralík, M.: Mixed-hybrid finite elements and streamline computation for the potential flow problem. Comput. Geosci. **18**(8/3), 217–234 (2005)