

MODFRAC, logiciel de modélisation et de maillage de réseaux de fractures

Description technique

G. Pichot¹ H. Borouchaki² P. Laug³

1. Inria Paris & Université Paris-Est, CERMICS (ENPC), geraldine.pichot@inria.fr
2. Université de Technologie de Troyes, houman.borouchaki@utt.fr
3. Inria Saclay (retraité)

1 Introduction

Il est bien connu que les fractures jouent un rôle majeur dans de nombreuses applications de la géologie (ressource en eau, stockage en profondeur, géothermie, entre autres) et ne peuvent pas être négligées dans les modélisations du sous-sol terrestre car elles constituent les principaux trajets de flux. Les fractures sont omniprésentes et ont une grande variété de tailles (de quelques centimètres à plusieurs kilomètres) et d'ouvertures (qui déterminent leur transmissivité).

Une stratégie possible pour modéliser ces réseaux de fractures est l'approche DFN (*Discrete Fracture Network*). Les fractures sont alors modélisées par des ellipses ou des polygones dont les propriétés (orientation, taille, position, transmissivité) sont gouvernées par des lois statistiques dérivées d'observations sur le terrain. Dans les simulations numériques, le domaine tridimensionnel Ω est typiquement un parallélépipède (appelé ici *cube* par souci de brièveté) contenant plusieurs milliers, voire **plus d'un million de fractures**.

Le logiciel MODFRAC, à partir d'un tel modèle DFN, calcule les intersections des fractures entre elles et avec les faces du *cube*, complète ce modèle par des informations géométriques et topologiques, et génère un maillage de ce modèle.

L'exécution se déroule en plusieurs étapes :

1. Lecture des paramètres de pilotage du logiciel.
2. Lecture du réseau de fractures.
3. Optionnellement, lecture de la description des puits et d'autres contraintes géométriques.
4. Conversion éventuelle des ellipses en polygones (sur demande de l'utilisateur).
5. Sélection de toutes les fractures se trouvant à l'intérieur du *cube*.
6. Calcul des intersections des fractures entre elles et avec les faces du *cube*. Ces intersections sont des segments droits sur les fractures et les faces du *cube*, et produisent éventuellement des arcs d'ellipses à la frontière des fractures.
7. Dans le cas d'un maillage réalisé sur l'amas connecté du DFN uniquement (le milieu poreux environnant n'est pas maillé), sélection des fractures connectées à des faces données du *cube*, à l'aide d'une structure de graphe. Seules ces fractures seront utiles aux simulations numériques ultérieures.
8. Calcul des intersections des segments droits précédents entre eux. Ces intersections sont des points à l'intersection de 3 faces qui sont soit des fractures, soit des faces du *cube*. Ces points d'intersection subdivisent les segments en sous-segments. Il est fondamental de gérer ces données de manière rigoureusement cohérente.

9. Transmission des informations précédentes à un mailleur surfacique « indirect », où le maillage tridimensionnel résulte de la construction de maillages plans des domaines de paramètres. La méthodologie de maillage repose sur une approche combinée frontale-Delaunay dans un contexte riemannien.
10. Amélioration de la qualité minimale du maillage.
11. Écriture du maillage surfacique, qui pourra éventuellement être utilisé par un mailleur volumique indépendant.
12. Écriture de fichiers qui pourront être utilisés pour des simulations numériques.

Le logiciel exploite au maximum la puissance de calcul disponible et la plupart des étapes sont parallélisées en utilisant un nombre arbitraire de *threads*, ce qui augmente fortement la vitesse de traitement sur les machines courantes comprenant un ou plusieurs processeurs multi-cœurs.

2 Fonctionnalités

Le logiciel MODFRAC est piloté par le fichier texte *modfrac.env*. Dans ce fichier, l'utilisateur peut modifier chaque paramètre (ou variable d'environnement) au moyen d'un mot-clé et d'une valeur (ou de plusieurs valeurs). Il convient ici de distinguer deux grandes classes de paramètres. Les premiers, appelés génériques, sont utilisés dans une grande variété d'applications de maillage. Nous donnons ci-dessous les paramètres génériques utilisés dans le contexte qui nous concerne, à savoir le maillage de réseaux de fractures. Les seconds paramètres, appelés spécifiques, ont été ajoutés pour contrôler certains aspects particuliers du logiciel MODFRAC.

Paramètres génériques

La liste suivante donne, pour chaque paramètre générique, son mot-clé, son type, sa description et sa valeur par défaut.

- ***angle_mesh*** : réel (0 à 90) indiquant la tolérance angulaire pour la discrétisation géométrique des ellipses délimitant les fractures (si *hgeo_flag* est positionné). Il s'agit de l'angle (en degrés) entre une arête de la discrétisation et la tangente à l'ellipse en chaque extrémité. La valeur par défaut est de 8°.
- ***export_amadeus*** : entier (0 à 4) indiquant le contenu du fichier de maillage *amadeus.mesh* qui sera généré : 0 pas de fichier, 4 fichier lisible par le mailleur volumique GHS3D, sinon fichier de visualisation (avec les champs Edges et Normals). Si *FRACporous* = 1 et *export_amadeus* ≠ 4, deux fichiers sont générés : *amadeus.mesh* pour le mailleur volumique et *amadeus-visu.mesh* pour la visualisation (il contient en plus les informations sur les arêtes). La valeur par défaut est 0.
- ***gradation*** : réel dans l'intervalle [1.1, 1.e+30] indiquant le rapport maximal de longueurs entre deux arêtes adjacentes quelconques (1.e+30 signifie que l'on ne veut pas de gradation). La valeur par défaut est 1.e+30.
- ***hgeo_flag*** : booléen (0, 1) indiquant que l'on souhaite un maillage géométrique (on peut positionner simultanément *hgeo_flag* et *hphy_flag*). Dans notre contexte, le maillage est contrôlé par la courbure des ellipses délimitant les fractures. La valeur par défaut est 0.
- ***hgeomin*** : réel indiquant la taille minimale des arêtes pour un maillage géométrique (*hgeo_flag* ≠ 0). La valeur par défaut est *diag* * 0.001, où *diag* est la longueur de la diagonale du *cube*.
- ***hgeomax*** : réel indiquant la taille maximale des arêtes pour un maillage géométrique (*hgeo_flag* ≠ 0). La valeur par défaut est *diag* * 0.1, où *diag* est la longueur de la diagonale du *cube*.
- ***hphy_flag*** : booléen (0, 1) indiquant que l'on souhaite un maillage physique, c'est-à-dire respectant des tailles spécifiées par l'utilisateur (on peut positionner simultanément *hgeo_flag* et *hphy_flag*). La valeur par défaut est 1.

- ***hphydef*** : réel indiquant la taille des arêtes pour un maillage physique (*hphy_flag* ≠ 0). La valeur par défaut est $diag * 0.01$, où *diag* est la longueur de la diagonale du *cube*.
- ***hphymmin*** : réel indiquant la taille minimale des arêtes pour un maillage physique (*hphy_flag* ≠ 0). La valeur par défaut est celle de *hgeommin*.
- ***hphymax*** : réel indiquant la taille maximale des arêtes pour un maillage physique (*hphy_flag* ≠ 0). La valeur par défaut est celle de *hgeommax*.
- ***intermedfile*** : booléen (0, 1) activant la génération de fichiers intermédiaires afin de vérifier le déroulement correct du logiciel. Si *intermedfile* = 1 et *refs* = 0, seuls des fichiers généraux sont créés. Si *refs* ≠ 0, *intermedfile* est forcé à 1 et des fichiers concernant plus particulièrement la fracture *refs* sont aussi créés. La valeur par défaut est 0.
- ***memory*** : entier indiquant le nombre maximal de sommets générés sur chaque fracture. La valeur par défaut est 100 000.
- ***number_of_threads*** : entier *n* indiquant le nombre souhaité de *threads* sur lesquels s'exécutent les parties parallèles du logiciel (s'il vaut 0 alors le processus reste séquentiel). Par exemple, sur un ordinateur de bureau équipé d'un seul processeur quadricœur, on obtiendra de bonnes performances avec *n* = 4. Ces performances sont parfois légèrement améliorées avec *n* > 4, par un phénomène d'*hyper-threading*. La valeur par défaut est 0.
- ***refs*** : entier donnant un numéro de fracture (cf. paramètre *intermedfile*). La valeur par défaut est 0. La correspondance entre le numéro *refs* et le numéro d'origine de la fracture se trouve dans le fichier *modfrac_frac_meshed.vector* (voir ci-dessous).
- ***size_computation*** : booléen (0, 1) forçant une gradation de taille autour des petites arêtes. La valeur par défaut est 0.
- ***verb*** : entier (0 à 100) indiquant le pourcentage de verbosité, ou niveau d'impression des messages du logiciel. La valeur par défaut est 0.

Paramètres spécifiques

Les paramètres ci-dessous ont été ajoutés spécialement pour le logiciel MODFRAC.

- ***FRACconforming*** : booléen (0, 1) indiquant que les discrétisations des sous-segments générés à l'étape 8 doivent être conformes. Dans le cas contraire, si un sous-segment est à l'intersection de deux fractures f_1 et f_2 , le nombre d'arêtes de la discrétisation de ce sous-segment sur la fracture f_1 peut être différent du nombre d'arêtes sur la fracture f_2 . La valeur par défaut est 1.
- ***FRACcube_faces*** : entier *n* indiquant les faces du *cube* auxquelles doivent être connectées les fractures (étape 7). Cet entier contient plusieurs chiffres, chaque chiffre représentant une face du *cube*. Par convention, les faces sont numérotées 1 ($x = xmin$), 2 ($x = xmax$), 3 ($y = ymin$), 4 ($y = ymax$), 5 ($z = zmin$) et 6 ($z = zmax$). Par exemple, si *n* = 12, toutes les fractures connectées (directement ou indirectement) aux faces 1 et 2 seront retenues. La valeur par défaut est 0 (toutes les fractures, connectées ou non, sont retenues).
- ***FRACcubemin*** : 3 réels (*xmin*, *ymin*, *zmin*) représentant les coordonnées du coin inférieur du *cube*. Ce paramètre est obligatoire.
- ***FRACcubemax*** : 3 réels (*xmax*, *ymax*, *zmax*) représentant les coordonnées du coin supérieur du *cube*. Ce paramètre est obligatoire.
- ***FRACcell_to_pol*** : entier *n* indiquant que chaque ellipse doit être remplacée par un polygone à *n* côtés (étape 4). La valeur par défaut est 0 (dans ce cas, les ellipses ne sont pas converties en polygones avant maillage).
- ***FRACeps_abs*** : réel indiquant un seuil absolu de distance ϵ_a en dessous duquel deux points sont considérés comme confondus. La valeur par défaut est $\epsilon_a = \epsilon_r \cdot diag$, où ϵ_r est donné par le paramètre *FRACeps_rel* et *diag* est la longueur de la diagonale du *cube*.
- ***FRACeps_rel*** : réel indiquant un seuil relatif de distance ϵ_r qui permet de calculer le seuil absolu ϵ_a défini par le paramètre *FRACeps_abs*. La valeur par défaut est 10^{-6} .
- ***FRACfilename*** : chaîne de caractères indiquant le nom du fichier du réseau de fractures. La valeur par défaut est la chaîne "frac-no.vector".

- **FRACfixed_constraints** : booléen (0, 1) indiquant de fixer certains sommets lors de l'optimisation finale du maillage (étape 10), afin de comparer certains tests numériques de manière très précise. Si ce booléen est vrai, l'optimisation ne déplace pas les points situés sur les contraintes géométriques droites (commençant par un 1 dans le fichier *constraints.dat*). L'optimisation ne déplace pas non plus les sommets situés sur les 4 côtés des faces du *cube* qui contiennent ces contraintes, ce qui permet de préserver entièrement les zones contraintes. La valeur par défaut est 0 (sommets libres).
- **FRACL** : réel d définissant un véritable cube (et non un parallélépipède quelconque) centré à l'origine et de côté d . **FRACL** d est équivalent à **FRACcubemin** $-d/2 -d/2 -d/2$ et **FRACcubemax** $+d/2 +d/2 +d/2$.
- **FRACpad_zeros** : booléen (0, 1) indiquant que les fichiers *inter-edges.dat* et *inter-cube-edges.dat* doivent être complétés par des 0 de manière à rendre constant le nombre d'éléments par ligne. La valeur par défaut est 0 (fichiers de tailles plus petites).
- **FRACporous** : booléen (0, 1) indiquant que les simulations numériques portent sur un milieu poreux. Dans ce cas, toutes les fractures sont maillées (**FRACcube_faces** est forcé à 0), les faces du *cube* sont maillées aussi et le maillage de sortie de MODFRAC peut être mis en entrée d'un mailleur volumique. La valeur par défaut est 0.
- **FRACqmin** : réel $qmin$ indiquant la qualité minimale requise pour le maillage (étape 10). Cette valeur, généralement petite (de l'ordre de 10^{-4}), est souvent indispensable pour les simulations numériques ultérieures. Pour tout triangle de qualité $q < qmin$, le logiciel MODFRAC améliore sa qualité jusqu'à une valeur $q' \geq qmin$ en déplaçant ses sommets si cela est possible. La valeur par défaut est 0.0 (pas d'amélioration demandée).
- **FRACsimulation** : booléen (0, 1). S'il vaut 0, seuls le fichier de maillage *amadeus.mesh* est généré. S'il vaut 1, d'autres fichiers utiles pour des simulations numériques sont être générés. La valeur par défaut est 1.
- **FRACwrite_frac** : booléen (0, 1) activant la conversion d'un réseau de fractures au format *.disk* en un réseau au format *.vector* (voir plus bas la description de ce format). Le nom du fichier de sortie est *frac-no.vector*. La valeur par défaut est 0 (pas de conversion).
- **FRACwrite_inter** : entier (0, 1, 2) indiquant l'option de génération des fichiers inter-* (voir plus bas). La valeur par défaut est 2.
- **FRACwrite_subdomain** : entier (0 à 7) indiquant l'option de génération des fichiers de sous-domaines (voir plus bas). La valeur par défaut est 0.

Après avoir lu les paramètres spécifiés par l'utilisateur (étape 1), le logiciel MODFRAC lit un fichier définissant un réseau de fractures (étape 2). Le nom de ce fichier est donné par le paramètre **FRACfilename**. Le suffixe de ce nom (*.vector*, *.disk* ou *.geo*) indique le format du fichier.

Fichier de fractures de suffixe *.vector*, en particulier *frac-no.vector* (entrée)

Dans ce format (cf. Table 9 de la référence [1]), chaque fracture est délimitée par une ellipse définie par son centre, sa normale, son axe majeur et son axe mineur. La première ligne du fichier précise la liste des variables utilisées :

no Laxis_maj Laxis_min c_x c_y c_z n_x n_y n_z no_fracture u_x u_y u_z v_x v_y v_z

Fichier de fractures de suffixe *.disk* (entrée)

Dans cet autre format [2], chaque fracture est délimitée par un cercle. La première ligne du fichier donne à nouveau les variables utilisées : *label id xc yc zc dip dipdir half_length aperture ...*

- *label* et *id* sont des variables qui peuvent être ignorées
- *xc*, *yc* et *zc* sont les coordonnées du centre du cercle
- *dip* est le pendage et *dipdir* l'azimut (données géologiques donnant la normale à la fracture)
- *half_length* est le rayon du cercle
- *aperture* est l'ouverture, qui détermine la transmissivité de la fracture

- Les variables supplémentaires, à la suite de la variable *aperture*, dépendent du contexte d'utilisation.

La deuxième ligne contient 2 entiers : le nombre total de fractures et le nombre de variables supplémentaires à la suite de *aperture*.

Les lignes suivantes contiennent, pour chaque fracture, les valeurs des variables qui la décrivent.

Fichier de fractures de suffixe *.geo* (entrée)

Dans ce format du logiciel GMSH [3], chaque fracture est délimitée par un polygone plan convexe. La syntaxe est rappelée ci-dessous de manière simplifiée :

```
Point(p) = {x, y, z, h}
Line(l) = {p, q}
Line Loop(l1) = {l1, l2, l3, ...}
Plane Surface(s) = {l1}
```

De cette manière, on définit successivement des points, des segments droits joignant deux points, des boucles de segments droits et enfin des surfaces planes délimitées par des boucles. Le logiciel MODFRAC en déduit l'origine et la normale de chaque polygone, puis les intersections des polygones entre eux (même si elles sont déjà données dans le fichier *.geo*) afin de garantir la cohérence topologique de l'ensemble.

Fichier de description de puits *wells.geo* (entrée)

L'exploitation des ressources souterraines nécessite souvent le forage d'un ou de plusieurs puits. Le fichier optionnel *wells.geo* (de format *.geo* comme précédemment) donne la description de tels puits (étape 3) sous forme de faces polygonales planes. Un exemple est donné en Section 3.

Notons que le fichier *wells.geo* permet de décrire non seulement des puits, mais encore toute surface composée d'un ou de plusieurs polygones plans.

Fichier de contraintes géométriques *constraints.dat* (entrée)

Comme indiqué dans l'introduction, l'étape optionnelle 3 permet de spécifier d'autres contraintes géométriques que les précédentes. Chaque contrainte est définie par une ligne de texte ayant l'une des formes suivantes :

- 1 Px Py Pz Qx Qy Qz
- 2 Ox Oy Oz Nx Ny Nz
- 2 Ox Oy Oz Px Py Pz Qx Qy Qz

Si la ligne commence par un 1, il s'agit d'une contrainte linéaire (1D), c'est-à-dire un segment droit joignant les points P et Q situés sur une même face du *cube*.

Si elle commence par un 2, il s'agit d'une contrainte surfacique (2D), c'est-à-dire un plan coupant le *cube*. Ce plan peut être défini par son origine O et sa normale N (6 réels), ou par son origine O et deux points P et Q lui appartenant (9 réels).

Fichiers de sélection *frac_meshed.vector* et *frac_unmeshed.vector* (entrée)

Le fichier *frac_meshed.vector* (resp. *frac_unmeshed.vector*), s'il existe, donne la liste des numéros des fractures à mailler (resp. à ne pas mailler).

Exécution

Après avoir lu les données précédentes, le logiciel MODFRAC s'exécute comme indiqué précédemment en calculant les intersections des fractures entre elles et avec les faces du *cube*, en sélectionnant les seules fractures utiles, en complétant ce modèle par des informations géométriques et topologiques, et en générant un maillage. Diverses informations sont affichées (en fonction du paramètre *verb*), en particulier les temps de calcul dans les parties séquentielles et parallèles. Les fichiers générés par le logiciel sont décrits ci-dessous.

Fichier de maillage *amadeus.mesh* (sortie)

Le fichier de maillage écrit par MODFRAC est au format *.mesh* (cf. Section 2.1 du rapport [4]). Il s'agit d'un fichier texte pouvant contenir plusieurs champs, notamment :

- Vertices : coordonnées des sommets.
- Edges : arêtes de la discrétisation des frontières des fractures et de leurs intersections.
- Triangles : triangles du maillage.

Pour rappel, le paramètre *export_amadeus* permet d'ajouter d'autres champs ou de générer un fichier supplémentaire *amadeus-visu.mesh*. Le format *.mesh* est compatible avec de nombreux logiciels de visualisation 3D et de maillage volumique, en particulier GHS3D [5].

Fichiers générés pour des simulations numériques (sortie)

Dans le cas général (valeurs par défaut des paramètres *FRACsimulation* et *FRACwrite_**), les fichiers ci-dessous sont générés en vue de simulations numériques ultérieures, typiquement dans un environnement MATLAB [6].

- *info.dat* : ce fichier contient diverses informations, à savoir *maxvgo* = plus grand numéro de sommet global des extrémités de sous-segments, *nclassic* = nombre de fractures « classiques » (sélectionnées dans le fichier d'entrée, et non créées artificiellement par des contraintes géométriques), *nconstraints* = nombre de contraintes géométriques, *nsegff*, *nsegfc*, *nsegcc* = nombre de segments intersections fracture/fracture, fracture/cube et cube/cube (ces dernières étant au nombre de 12), *nsubsegff*, *nsubsegfc*, *nsubsegcc* = nombre de sous-segments intersections fracture/fracture, fracture/cube et cube/cube, *ndom* = nombre de domaines, *nsubdom* = nombre de sous-domaines.

Numérotation des faces : par convention, les fractures « classiques » sont numérotées de 1 à *nclassic*, les fractures créées par les contraintes géométriques de *nclassic+1* à $N = nclassic+nconstraints$, et les faces du *cube* de $N+1$ à $N+6$.

- *edges.vector* : arêtes du maillage avec leurs caractéristiques [1, Table 11].
- *inter-edges.dat* : pour chaque intersection discrétisée, son numéro, numéro d'une fracture la contenant (pour les cas non conformes), numéros globaux de ses arêtes de la discrétisation.
- *inter-no.vector* : pour chaque intersection discrétisée, numéro de ligne, nombre d'arêtes, numéros des deux fractures créant l'intersection, numéro d'une fracture (pour les cas non conformes), numéro de l'intersection [1, Table 16].
- *inter-vertices.dat* : pour chaque intersection discrétisée, son numéro, numéro d'une fracture contenant cette intersection (pour les cas non conformes), numéros globaux des sommets de la discrétisation.
- *modfrac_frac_meshed.vector* : pour chaque fracture sélectionnée et maillée par le logiciel MODFRAC, son numéro d'origine dans le fichier *frac-no.vector*.
- *null-patches.vector* : numéros des fractures non maillées, s'il en existe.
- *numbers.vector* : pour chaque fracture, nombre de sommets, nombre d'arêtes et nombre de triangles.
- *quality-data.dat*, *quality-hist.dat*: informations sur la qualité du maillage, sous forme libre ou sous forme adaptée à un tableur de type Excel.
- *triangles.vector* : pour chaque triangle, numéros locaux de ses sommets et de ses arêtes [1, Table 10].

- *vertices-glo.vector* : coordonnées 3D des sommets du maillage.
- *vertices.vector* : coordonnées 2D des sommets du maillage [1, Table 12].

Fichiers intermédiaires générés sous condition (sortie)

Si *intermedfile* $\neq 0$, des fichiers intermédiaires sont générés afin de vérifier le déroulement correct du logiciel. Le fichier *intermed-tln3d.mesh* permet de visualiser tous les sous-segments intersections en 3D. Le fichier *intermed.geo* donne le modèle géométrique au format *.geo* du logiciel GMSH [3].

Si de plus *refs* $\neq 0$, une dizaine de fichiers concernant la face *refs* et de noms *intermed-refs-** sont générés, notamment *intermed-refs-dc-cont-2d.mesh* qui contient la discrétisation utilisée en entrée du mailleur plan et *intermed-refs-frac_meshed.vector* qui contient les numéros des fractures connectées à la fracture *refs*.

Fichier *frac-no.vector* généré sous condition (sortie)

Si *FRACwrite_frac* $\neq 0$, un réseau de fractures au format *.disk* (cercles) est converti au format *.vector* (ellipses). Ces deux formats sont décrits plus haut.

Fichiers *inter-** générés sous condition (sortie)

Si *FRACwrite_inter* $\neq 0$, le logiciel décrit les intersections de fractures dans les fichiers *inter-no.vector*, *inter-vertices.dat* et *inter-edges.dat* (voir plus haut). Si *FRACwrite_inter* = 1, il s'agit des intersections concernant toutes les faces (fractures et faces du *cube*). Si *FRACwrite_inter* = 2, intersections ne concernant aucune face du *cube*. Enfin, si *FRACwrite_inter* = 3, intersections concernant au moins une face du *cube*.

Fichiers de sous-domaines générés sous condition (sortie)

Si *FRACwrite_subdomain* $\neq 0$, les numéros locaux des sous-domaines sont donnés, avec plusieurs options : dans un fichier séparé *subdomain-*.vector* ou non, dans le fichier *amadeus.mesh* ou non, pour toutes les fractures et les 6 faces du *cube*, ou pour les 6 faces du *cube* seulement. Ces options sont déterminées par la valeur du paramètre :

- 0 ou 1 : pas de numéros donnés
- 2 : fichier *subdomain-cube*, sans fractures
- 3 : fichier *subdomain-frac-cube*, avec fractures
- 4 : fichier *amadeus*, sans fractures
- 5 : fichier *amadeus*, avec fractures
- 6 : fichier *subdomain-cube* et *amadeus*, sans fractures
- 7 : fichier *subdomain-frac-cube* et *amadeus*, avec fractures

3 Exemples

Dans cette section, sont présentés plusieurs exemples de maillages générés avec MODFRAC.

La géométrie des réseaux de fractures présentés dans cette section a été fournie par le logiciel DFN.lab développé par Fractory, laboratoire commun entre ITASCA Consultants SAS, le CNRS et l'Université de Rennes (<https://fractorylab.org>).

Les simulations ont été lancées sur un ordinateur portable Intel(R) Core(TM) i7-7820HQ CPU @ 2.90GHz avec 4 cœurs (8 processeurs logiques) et 32 Go de RAM.

❖ **Réseaux de fractures particuliers (de type boîte à sucre et Warren&Root)**

Les premiers exemples portent sur des réseaux particuliers de type boîte à sucre et Warren&Root. Le *cube* est coupé complètement par un réseau organisé de fractures dont la longueur et la largeur sont nettement supérieures à la taille du *cube*.

L'exemple suivant contient 15 fractures s'organisant comme un réseau de type boîte à sucre. Le fichier d'entrée *frac-no.vector* est donné par la Table 1.

No	Laxis_maj	Laxis_min	c_x	c_y	c_z	n_x	n_y	n_z	no_fracture	u_x	u_y	u_z	v_x	v_y	v_z
1	10	10	0	0	0	0	1	0	0	0	0	1	1	0	0
2	10	10	0	1	0	0	1	0	1	0	0	1	1	0	0
3	10	10	0	2	0	0	1	0	2	0	0	1	1	0	0
4	10	10	0	-1	0	0	1	0	3	0	0	1	1	0	0
5	10	10	0	-2	0	0	1	0	4	0	0	1	1	0	0
6	10	10	0	0	0	1	0	0	5	0	0	1	0	1	0
7	10	10	1	0	0	1	0	0	6	0	0	1	0	1	0
8	10	10	2	0	0	1	0	0	7	0	0	1	0	1	0
9	10	10	-1	0	0	1	0	0	8	0	0	1	0	1	0
10	10	10	-2	0	0	1	0	0	9	0	0	1	0	1	0
11	10	10	0	0	0	0	0	1	10	1	0	0	0	1	0
12	10	10	0	0	1	0	0	1	11	1	0	0	0	1	0
13	10	10	0	0	2	0	0	1	12	1	0	0	0	1	0
14	10	10	0	0	-1	0	0	1	13	1	0	0	0	1	0
15	10	10	0	0	-2	0	0	1	14	1	0	0	0	1	0

Table 1. Boîte à sucre : exemple de fichier *frac-no.vector*.

La géométrie correspondant à ce fichier est donnée par la Figure 1.

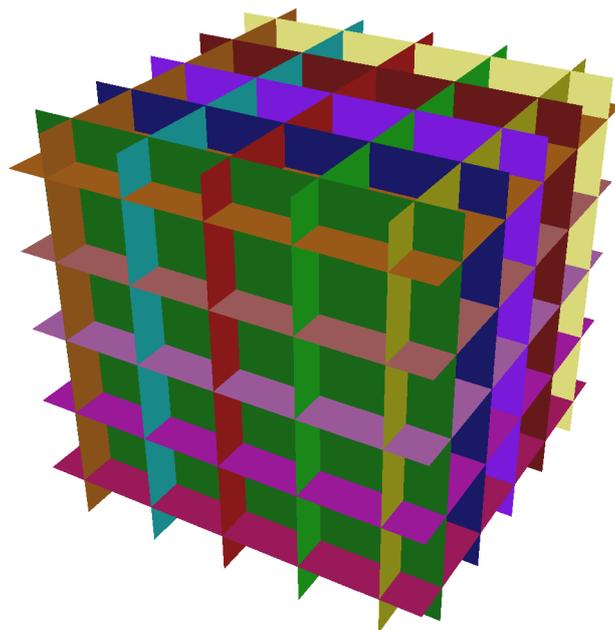


Figure 1. Géométrie de type boîte à sucre – 15 fractures.

Le fichier *modfrac.env* suivant (Table 2) permet de générer le maillage présenté sur la Figure 2. Par défaut le maillage est conforme (*FRACconforming* à 1). Ce maillage a été réalisé par MODFRAC en 1s.

```

verb 10

FRACL 5
FRACcube_faces 12
FRACporous 0
FRACqmin 1.e-6
FRACwrite_frac 0

hgeo_flag 1
hgeomax 1000.
angle_mesh 30
    
```

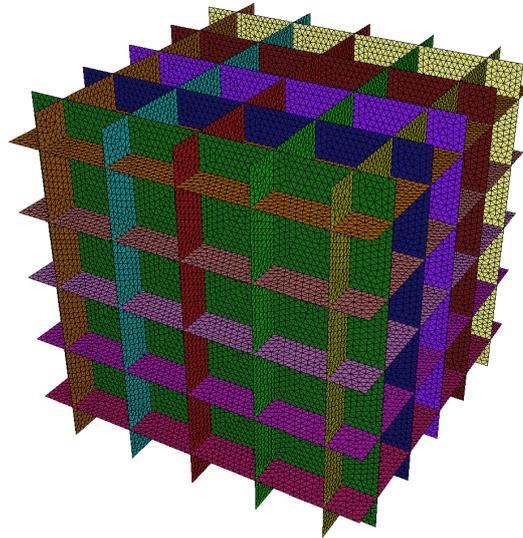


Table 2. Fichier *modfrac.env*.

Figure 2. Maillage de la boîte à sucre – 15 fractures.

La qualité du maillage est donnée par le fichier *quality-data.dat* (Table 3) et par l'histogramme de la Figure 3.

```

111240 (#triangles)
0.789048 (qmin)
42055 (qmin:triangle)
25408 (qmin:v1)
25369 (qmin:v2)
25758 (qmin:v3)
6 (qmin:fracture_ref)
0.980217 (qmoy)
0.999999 (qmax)
80054 (qmax:triangle)
40272 (qmax:v1)
40324 (qmax:v2)
39768 (qmax:v3)
11 (qmax:fracture_ref)
    
```

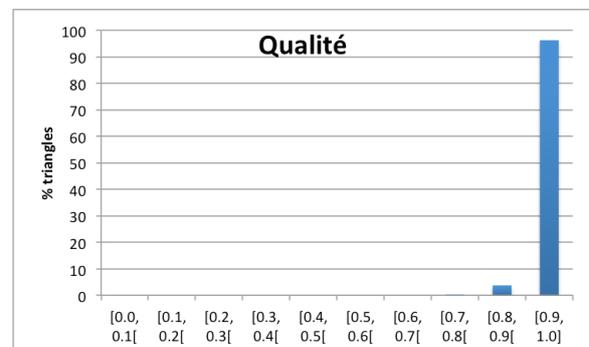


Table 3. Fichier *quality-data.dat*.

Figure 3. Histogramme de qualité du maillage.

Du fichier *quality-data.dat* (Table 3) sont déduites des propriétés du maillage généré :

- Il contient 111.240 triangles (#triangles),
- Il a une qualité minimum 0.789048 (qmin) au triangle 42055 (qmin:triangle) qui a pour sommets 25408 (qmin:v1), 25369 (qmin:v2) et 25758 (qmin:v3) appartenant à la fracture 6 (qmin:fracture_ref). Dans cet exemple, toutes les fractures ont été sélectionnées et maillées. Dans le cas contraire, la correspondance entre le numéro *fracture_ref* et le numéro d'origine de la fracture se trouve dans le fichier *modfrac_frac_meshed.vector*.
- Il a une qualité moyenne de 0.980217 (qmoy),

- Il a une qualité maximum de 0.999999 (qmax) au triangle 80054 (qmax:triangle) qui a pour sommets 40272 (qmax:v1), 40324 (qmax:v2) et 39768 (qmax:v3) appartenant à la fracture 11 (qmax :fracture_ref),
- Le maillage est de bonne qualité et ne contient aucun triangle de qualité inférieure à 0.7.

D'autres géométries particulières peuvent être maillées, comme les réseaux de type Warren&Root pour lesquels les fractures ne sont pas nécessairement régulièrement espacées, comme présenté sur la Figure 4.

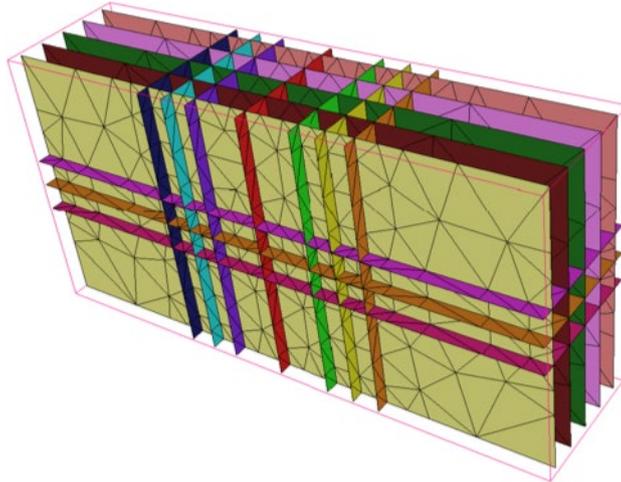


Figure 4. Maillage d'un réseau de type Warren&Root – 15 fractures.

❖ Réseaux de fractures représentées par des ellipses

A présent, nous considérons des réseaux de fractures de type réseaux de fractures discrets (Discrete Fracture Network – DFN) pour lesquels les fractures sont des disques ou des ellipses qui peuvent couper le *cube* et/ou se retrouver complètement immergées dans le *cube*.

Commençons par des réseaux DFN de petite taille pour illustrer les différentes options disponibles en entrée dans le fichier *modfrac.env*.

Le réseau de fractures présenté sur la Figure 5 est contenu dans un cube de côté L20. Il possède 1397 fractures décrites dans un fichier *frac-no.vector*.

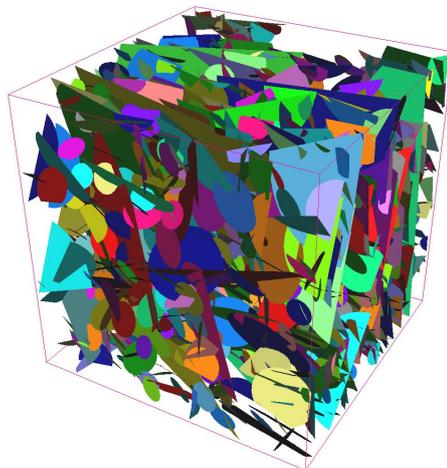


Figure 5. DFN L20 – 1397 fractures.

```

verb 10
FRACL 20
FRACcube_faces 12
FRACporous 0
FRACqmin 1.e-4
FRACwrite_frac 0
hgeo_flag 1
hgeomax 1000.
angle_mesh 30
number_of_threads 4
export_amadeus 1

```

Table 4. Fichier *modfrac.env* .

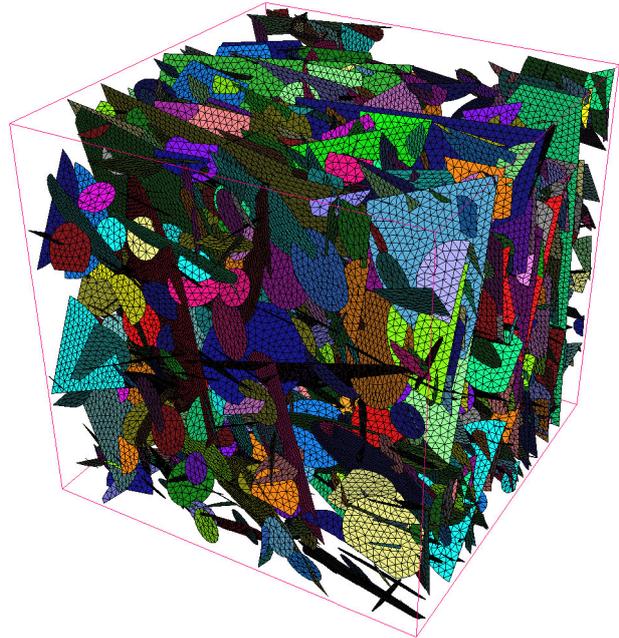


Figure 6. DFN L20 – Maillage – 1397 fractures.

Le maillage généré à partir du fichier *modfrac.env* donné Table 4 est illustré Figure 6. Le maillage contient 178407 triangles et a été réalisé en 17s.

La qualité du maillage est donnée par le fichier *quality-data.dat* (Table 5) et l’histogramme de la Figure 7. La qualité est légèrement inférieure à la précédente, du fait des nombreuses contraintes géométriques formant des angles proches de 0° ou 180°. Notons que le critère de qualité minimale FRACqmin imposé dans le fichier *modfrac.env* est satisfait : 0.000565164 (qmin) > 0.0001 (FRACqmin).

```

178407 (#triangles)
0.000565164 (qmin)
67234 (qmin:triangle)
15628 (qmin:v1)
3337 (qmin:v2)
620 (qmin:v3)
362 (qmin:fracture_ref)
0.945222 (qmoy)
1 (qmax)
74423 (qmax:triangle)
15251 (qmax:v1)
15252 (qmax:v2)
64257 (qmax:v3)
421 (qmax:fracture_ref)

```

Table 5. Fichier *quality-data.dat*.

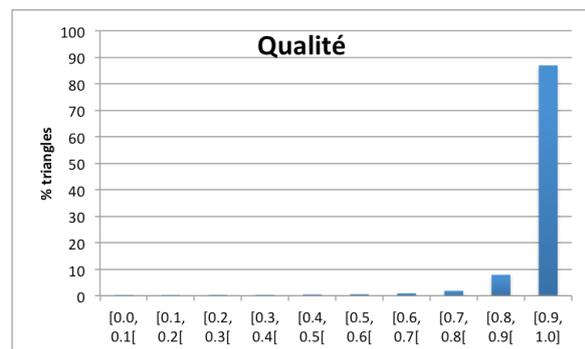


Figure 7. Histogramme de qualité du maillage.

Grâce aux paramètres *gradation* et *angle_mesh*, MODFRAC permet de définir différents niveaux de raffinement, comme indiqué sur les Figures 8 et 9.



Figure 8. Angle 30° and gradation 1.4.



Figure 9. Angle 10° and gradation 1.2.

MODFRAC est capable de mailler des réseaux de fractures beaucoup plus gros. Par exemple le réseau présenté Figure 10 contient 1.176.566 fractures. Le maillage réalisé par MODFRAC et généré à partir du fichier *modfrac.env* donné Table 6 contient 20.723.302 triangles et a été généré en 14 minutes avec 4 threads.

```

verb 10
FRACL 200
FRACcube_faces 12
FRACporous 0
FRACqmin 1.e-4
FRACwrite_frac 0
hgeo_flag 1
hgeomax 1000.
angle_mesh 30
number_of_threads 4
export_amadeus 1
memory 800000

```

Table 6. Fichier *modfrac.env*.

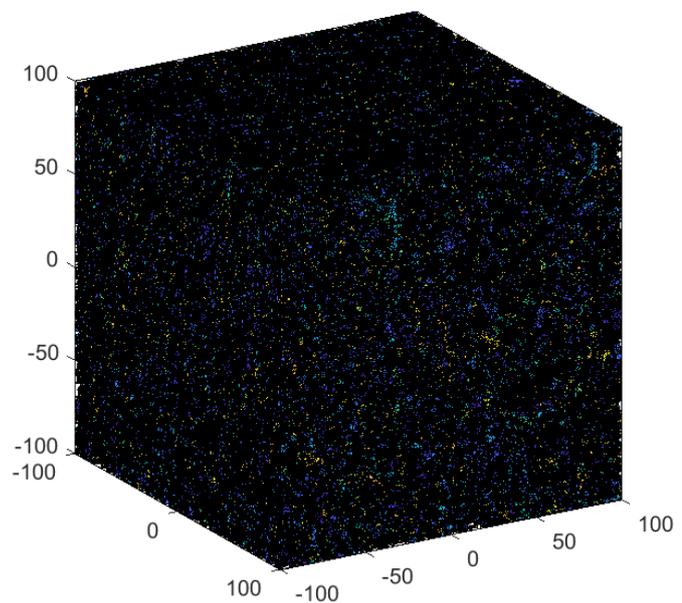


Figure 10. DFN L200 – Maillage – 1.176.566 fractures.

La qualité du maillage est donnée par les fichiers *quality-data.dat* de la Table 7 et par l’histogramme de la Figure 11.

20723302 (#triangles)
4.81633e-05 (qmin)
8229123 (qmin:triangle)
3143093 (qmin:v1)
3143092 (qmin:v2)
10401401 (qmin:v3)
327098 (qmin:fracture_ref)
0.616738 (qmoy)
1 (qmax)
19269394 (qmax:triangle)
9065201 (qmax:v1)
10872669 (qmax:v2)
9065200 (qmax:v3)
1067726 (qmax:fracture_ref)

Table 7. Fichier quality-data.dat.

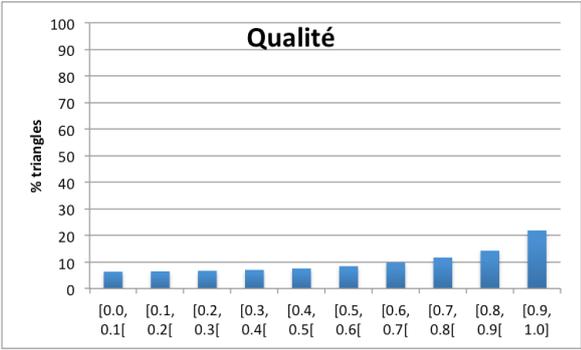


Figure 11. Histogramme de qualité du maillage.

Pour un réseau aussi complexe, le mailleur est capable d'améliorer la qualité minimale des triangles, même si elle reste un peu au-dessous de la qualité demandée ($4.8e-05 < \mathbf{FRACqmin}$). La sortie écran indique que cela arrive pour 3 triangles.

❖ **Réseaux de fractures représentées par des polygones**

En utilisant le paramètre *FRACell_to_pol*, il est possible de transformer les ellipses en polygones avant de les mailler. Attention ! Cela peut changer le réseau de fractures considéré en fonction du nombre *FRACell_to_pol* de côtés choisis. Par exemple, sur le cas test L20 Figure 5, en ajoutant le paramètre *FRACell_to_pol* à 10 au fichier *modfrac.env* de la Table 4, on obtient le maillage Figure 12 qui contient 1311 polygones (alors que le réseau initial contenait 1397 fractures). En effet, une ellipse peut couper le *cube* alors que le polygone qui la discrétise est entièrement extérieur au *cube* et deux ellipses peuvent se couper entre elles alors que les polygones correspondants ne se coupent pas.

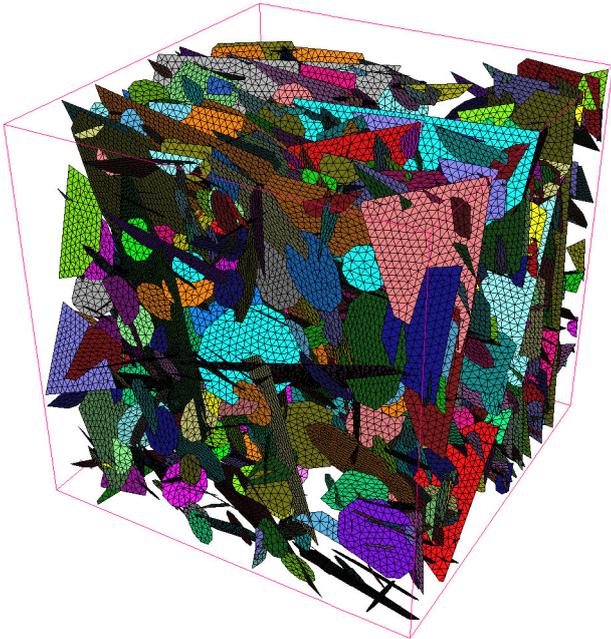


Figure 12. DFN L20 – les ellipses sont remplacées par des polygones à 10 côtés – 1311 polygones.

La qualité est donnée par la Table 8 et par l'histogramme de la Figure 13.

132824 (#triangles)
0.000498204 (qmin)
4090 (qmin:triangle)
427 (qmin:v1)
17855 (qmin:v2)
430 (qmin:v3)
29 (qmin:fracture_ref)
0.941964 (qmoy)
1 (qmax)
68418 (qmax:triangle)
18139 (qmax:v1)
18140 (qmax:v2)
56964 (qmax:v3)
534 (qmax:fracture_ref)

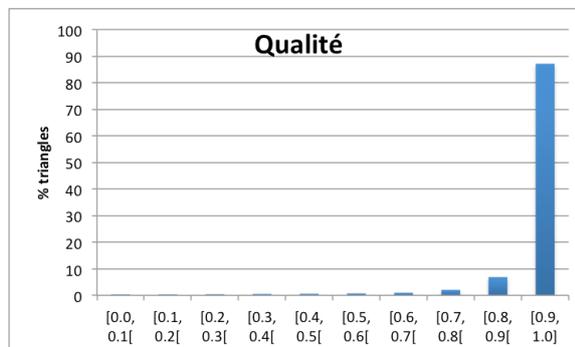


Table 8. Fichier quality-data.dat.

Figure 13. Histogramme de qualité du maillage.

MODRAC est également capable de prendre en entrée un fichier de type **.geo**. Dans ce format du logiciel GMSH [3], chaque fracture est délimitée par un polygone plan convexe.

❖ Ajout de puits

Un exemple simple de fichier **wells.geo** est donné ci-dessous (Table 9).

Point(1) = {0.5, 0, -3, h};	Line(1) = {1, 2};	Line Loop(1) = {-8, -7, -6, -5, -4, -3, -2, -1};
Point(2) = {0.353553, 0.353553, -3, h};	Line(2) = {2, 3};	Line Loop(2) = {9, 10, 11, 12, 13, 14, 15, 16};
Point(3) = {0, 0.5, -3, h};	Line(3) = {3, 4};	Line Loop(3) = {1, 18, -9, -17};
Point(4) = {-0.353553, 0.353553, -3, h};	Line(4) = {4, 5};	Line Loop(4) = {2, 19, -10, -18};
Point(5) = {-0.5, 0, -3, h};	Line(5) = {5, 6};	Line Loop(5) = {3, 20, -11, -19};
Point(6) = {-0.353553, -0.353553, -3, h};	Line(6) = {6, 7};	Line Loop(6) = {4, 21, -12, -20};
Point(7) = {0, -0.5, -3, h};	Line(7) = {7, 8};	Line Loop(7) = {5, 22, -13, -21};
Point(8) = {0.353553, -0.353553, -3, h};	Line(8) = {8, 1};	Line Loop(8) = {6, 23, -14, -22};
Point(9) = {0.5, 0, 5, h};	Line(9) = {9, 10};	Line Loop(9) = {7, 24, -15, -23};
Point(10) = {0.353553, 0.353553, 5, h};	Line(10) = {10, 11};	Line Loop(10) = {8, 17, -16, -24};
Point(11) = {0, 0.5, 5, h};	Line(11) = {11, 12};	Plane Surface(1) = {1};
Point(12) = {-0.353553, 0.353553, 5, h};	Line(12) = {12, 13};	Plane Surface(2) = {2};
Point(13) = {-0.5, 0, 5, h};	Line(13) = {13, 14};	Plane Surface(3) = {3};
Point(14) = {-0.353553, -0.353553, 5, h};	Line(14) = {14, 15};	Plane Surface(4) = {4};
Point(15) = {0, -0.5, 5, h};	Line(15) = {15, 16};	Plane Surface(5) = {5};
Point(16) = {0.353553, -0.353553, 5, h};	Line(16) = {16, 9};	Plane Surface(6) = {6};
	Line(17) = {1, 9};	Plane Surface(7) = {7};
	Line(18) = {2, 10};	Plane Surface(8) = {8};
	Line(19) = {3, 11};	Plane Surface(9) = {9};
	Line(20) = {4, 12};	Plane Surface(10) = {10};
	Line(21) = {5, 13};	
	Line(22) = {6, 14};	
	Line(23) = {7, 15};	
	Line(24) = {8, 16};	

Table 9. Fichier wells.geo.

Le fichier précédent décrit un puits vertical allant de la surface $z = 5$ à une profondeur $z = -3$, le domaine global étant défini dans le fichier d'environnement *modfrac.env* comme un cube centré à l'origine et de côté 10. La section de ce puits est approchée par un octogone, également centré à

l'origine. Le maillage surfacique créé par MODFRAC, ainsi que le maillage volumique provenant de GH3D, sont illustrés par la Figure 14.

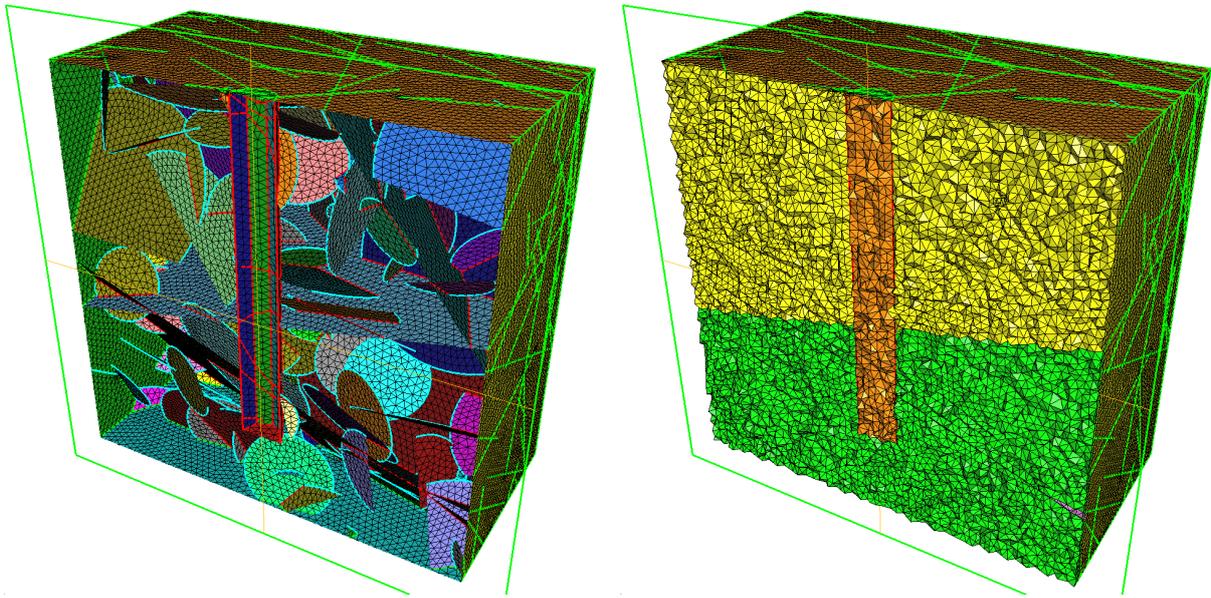


Figure 14. À gauche, coupe d'un maillage surfacique contenant un puits. À droite, coupe du maillage volumique correspondant.

❖ Maillage poreux fracturés

L'option *FRACporeux* à 1 permet de mailler également les faces du *cube*. La sortie de MOFRAC peut alors être proposée en entrée d'un mailleur volumique. La figure 10 donne un exemple de maillage volumique, réalisé avec le logiciel GHS3D (Inria), à partir du maillage surfacique généré avec MODFRAC.

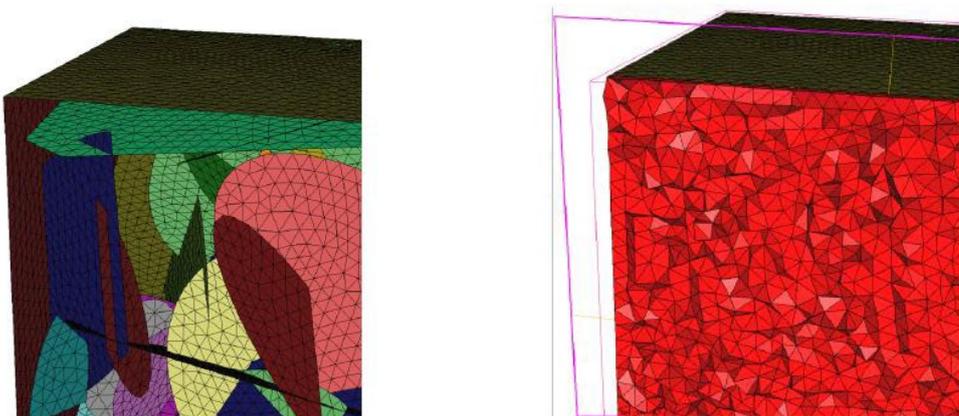


Figure 15. Exemple de maillage volumique (image de droite) réalisé avec GHS3D (Inria) à partir du maillage surfacique généré avec MODFRAC (image de gauche).

A titre d'exemple, dans l'article [7], un réseau de 52 polygones est proposé. La figure 16 illustre le cas test.

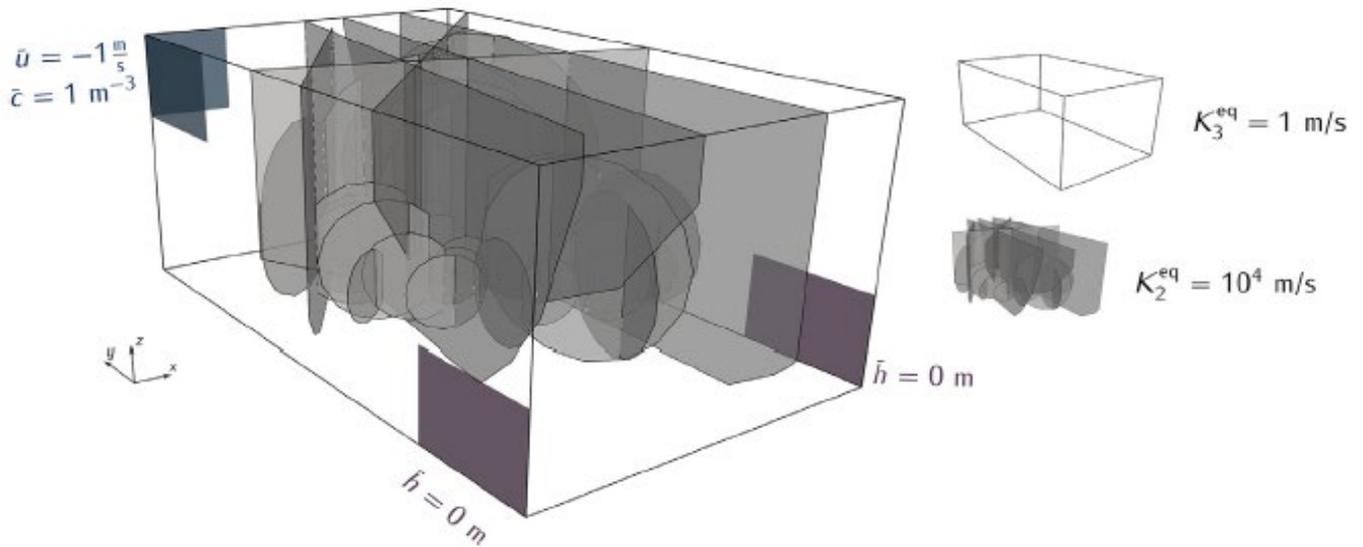


Figure 16. Un exemple de cas test poreux fracturés [7] – 52 fractures représentées par des polygones.

Les données sont disponibles au lien suivant : <https://git.iws.uni-stuttgart.de/benchmarks/fracture-flow-3d.git>.

En utilisant le fichier *modfrac.env* avec **FRACporous** à 1 et **FRACfixed_constraints** à 1 comme proposé Table 11 et le fichier *constraints.dat* proposés Table 12 pour définir les zones grisées sur la Figure 16, MODFRAC permet de générer le maillage illustré Figure 17.

```

verb 10

FRACcubemin -500 100 -100
FRACcubemax 350 1500 500
FRACcube_faces 12
FRACporous 1
FRACqmin 1.e-4
FRACwrite_frac 0
FRACfixed_constraints 1
FRACwrite_subdomain 2

hgeo_flag 1
hgeomax 1000.
angle_mesh 30

number_of_threads 4
export_amadeus 1
size_computation 1
memory 800000

```

Table 11. Fichier modfrac.env.

# d Omega in 0			
1	-200.0	1500.0	500.0
1	-200.0	1500.0	300.0
# d Omega in 1			
1	-500.0	1500.0	300.0
1	-500.0	1200.0	300.0
# d Omega out 0			
1	-500.0	100.0	100.0
1	-500.0	400.0	100.0
# d Omega out 1			
1	350.0	100.0	100.0
1	350.0	400.0	100.0

Table 12. Fichier constraints.dat.

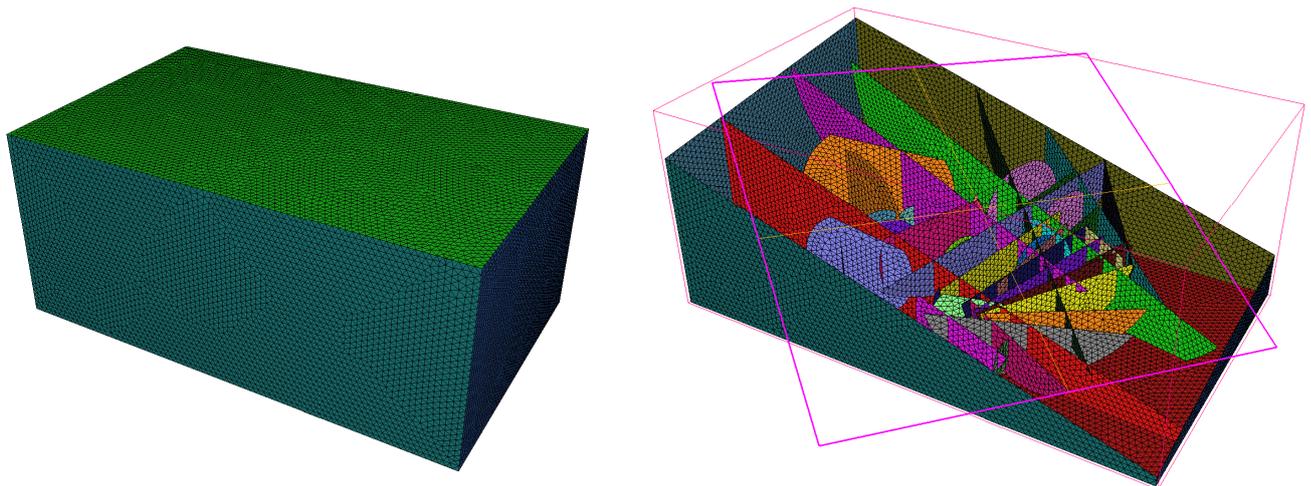


Figure 17. Cas test poreux fracturés [7] – 52 fractures. Maillage surfacique : vue d’ensemble à gauche et coupe à droite.

Ce maillage peut ensuite être proposé en entrée d'un mailleur volumique, tel que GHS3D (Inria) (cf. Figure 18).

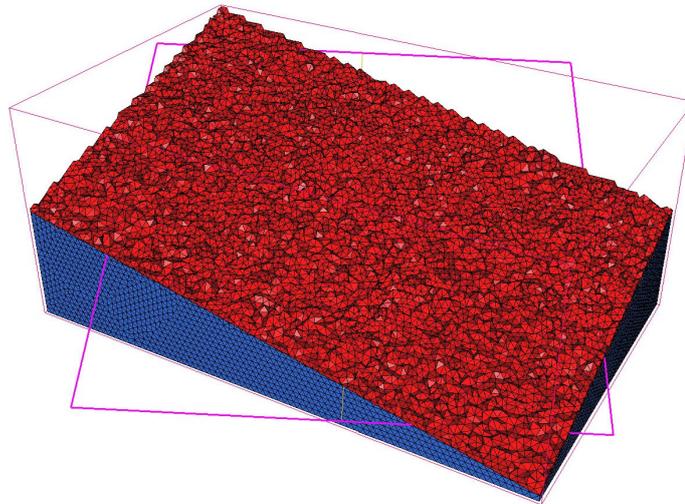


Figure 18. Cas test poreux fracturés [7] – 52 fractures. Maillage volumique généré par le logiciel GHS3D (Inria).

Ces maillages peuvent être utilisés pour faire des simulations numériques. À titre d'exemple, la Figure 19 montre la solution en charge hydraulique obtenue grâce à une résolution d'un écoulement stationnaire monophasique au sein de milieux poreux fracturés [7] par une méthode éléments finis mixte hybride.

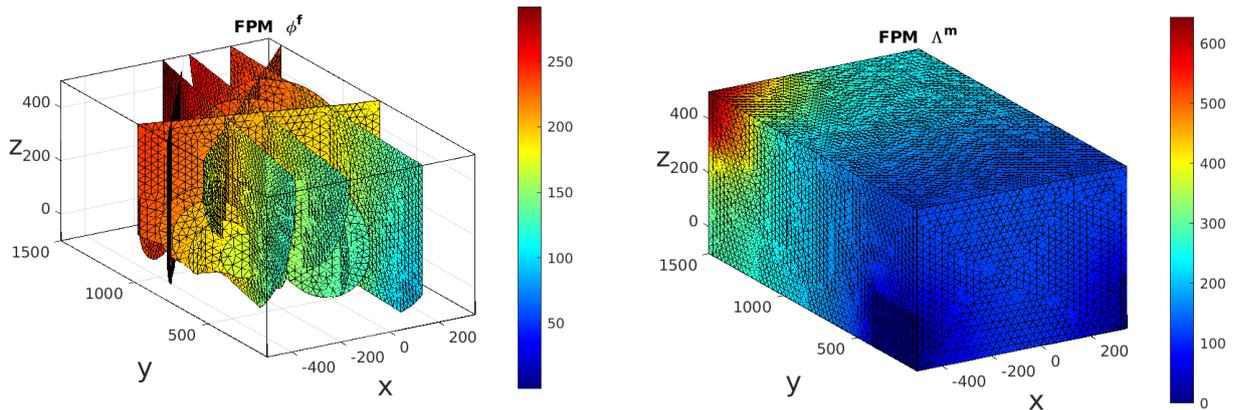


Figure 19. Cas test poreux fracturés [7] – 52 fractures. Charge hydraulique obtenue en résolvant le problème d'écoulement en milieux poreux fracturés [7] par une méthode éléments finis mixte hybride (RT0) avec le logiciel nef-flow-fpm (Inria).

L'exemple suivant montre un réseau contenant 87.335 fractures. La Figure 20 montre le maillage généré avec MODFRAC. Il contient 3.929.522 triangles et a été réalisé en 1 minute 37 secondes avec 4 threads.

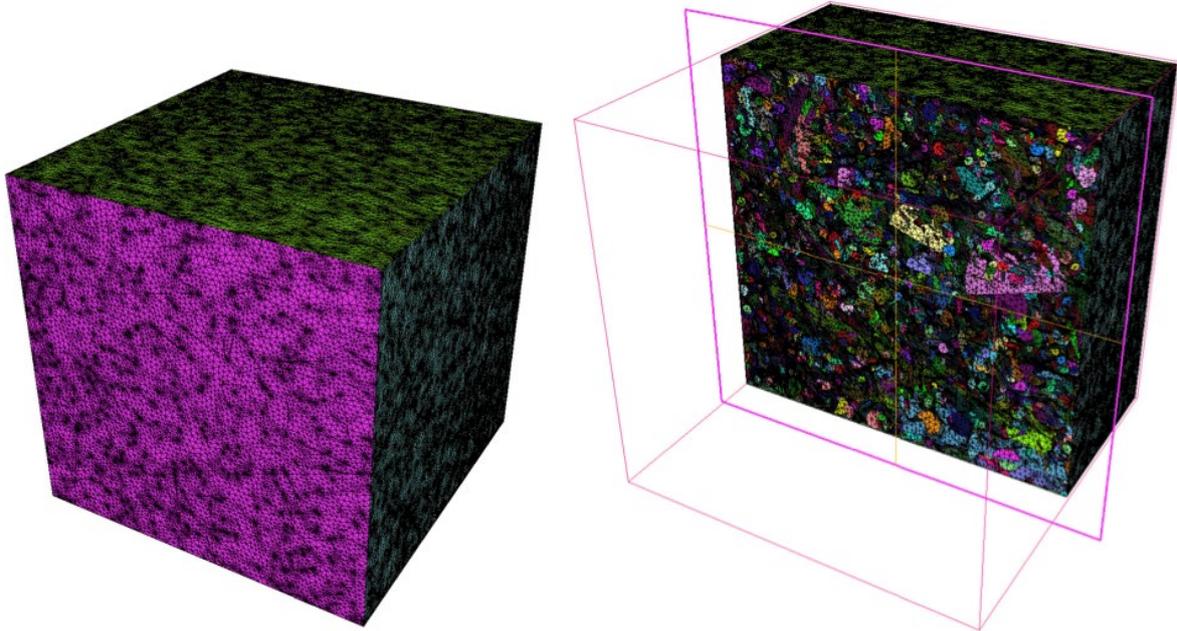


Figure 20. Cas test poreux fracturés – 87.335 fractures. Maillage surfacique : vue d'ensemble à gauche et coupe à droite.

Ce maillage peut ensuite être proposé en entrée d'un mailleur volumique, tel que GHS3D (Inria) (cf. Figure 21).

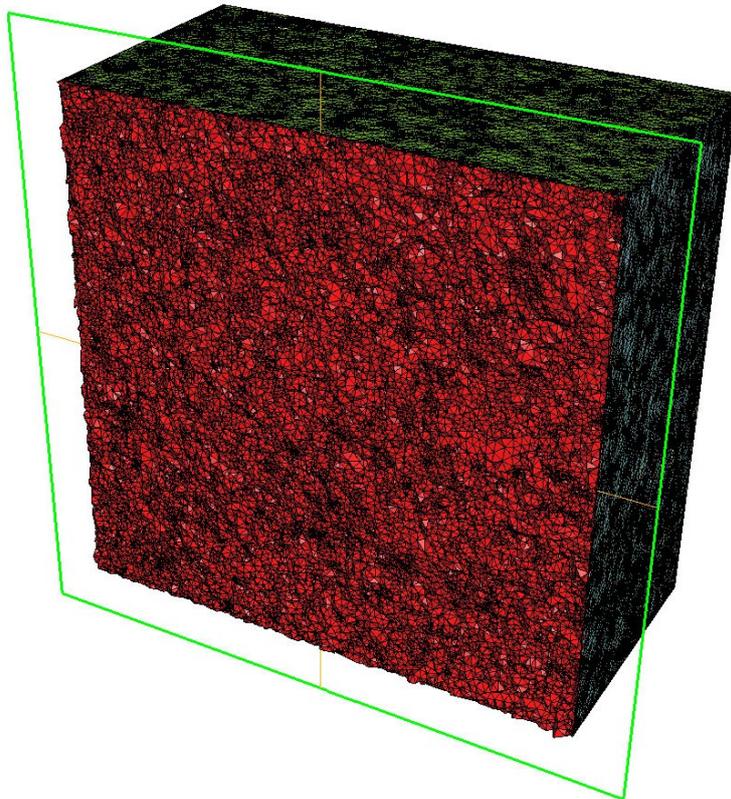


Figure 21. Cas test poreux fracturés – 87.335 fractures. Maillage volumique généré par le logiciel GHS3D (Inria).

3 Travaux futurs

Le logiciel MODFRAC répond dès maintenant aux demandes essentielles concernant la modélisation et le maillage des fractures. Cependant, ces demandes ne cessent d'évoluer, ce qui implique de nouveaux travaux à effectuer. Cette section décrit les principales études en cours.

Surfaces triangulées. Actuellement, les objets traités par MODFRAC (fractures, puits, *cube*) sont définis par des ellipses ou des polygones plans. Il est prévu d'y ajouter des faces triangulées afin de représenter simplement toute surface gauche. Un premier avantage concerne le *cube*, qui pourrait ainsi être généralisé à un domaine Ω de forme quelconque. De plus, il serait possible de représenter avec une meilleure précision des puits de section cylindrique et suivant par ailleurs des tracés quelconques, comme le permettent les techniques de forage directionnel. Enfin, les fractures elles-mêmes pourraient être de formes arbitraires. Pour intégrer cette nouvelle représentation des surfaces, il est nécessaire de calculer les lignes d'intersection tracées sur chaque surface (intersection triangle/triangle en 3D), puis les reporter sur un dépliage isométrique de chaque surface (qui sera un domaine paramétrique associé). Ensuite, ces lignes doivent être rendues conformes dans chaque domaine de paramètres (intersection arête/arête en 2D). Une fois cette représentation établie, on disposera d'un domaine paramétrique valide et d'une paramétrisation discrète de chaque surface, et ainsi générer tout maillage souhaité.

Décomposition de domaine. Une autre étude concerne la décomposition d'un domaine parallélépipédique Ω en plusieurs blocs. Cela permettra une parallélisation supplémentaire des calculs, d'où de nouveaux gains de temps et surtout le traitement d'un nombre encore plus important de fractures, de l'ordre de plusieurs dizaines de millions.

Références

- [1] J.R. de Dreuzy, Géraldine Pichot, B. Poirriez d, J. Erhel. Synthetic benchmark for modeling flow in 3D fractured media. *Computers & Geosciences*. 50 (2013) 59-71.
- [2] R. Le Goc, File format *.disk*, rapport interne Fractory, <https://fractorylab.org>.
- [3] Ch. Geuzaine, J.F. Remacle, Gmsh Reference Manual, <https://gmsh.info/doc/texinfo/gmsh.pdf>.
- [4] P. Frey, Medit : outil interactif de visualisation de maillages, RT-0247, Inria, 2001.
- [5] Inria-Simulog, GHS3D, tetrahedral mesh generator, rapport interne, 2005.
- [6] MATLAB, analyse de données, développement d'algorithmes et création de modèles mathématiques, <https://fr.mathworks.com>.
- [7] I. Berre, W. M. Boon, B. Flemisch, A. Fumagalli, D. Gläser, E. Keilegavlen, A. Scotti, I. Stefansson, A. Tatomir, K. Brenner, S. Burbullah, P. Devloo, O. Duran, M. Favino, J. Hennicker, I. Lee, K. Lipnikov, R. Masson, K. Mosthaf, M. G. C. Nestola, C. Ni, K. Nikitin, P. Schälde, D. Svyatskiy, R. Yanbarisov, P. Zulian. Verification benchmarks for single-phase flow in three-dimensional fractured porous media. *Advances in Water Resources*, Volume 147. 2021.