

# Rapport de stage

Elies Harington, Monique Teillaud, Vincent Despre, Équipe GAMBLE

18 août 2019

## Le contexte général

Les triangulations de Delaunay sont étudiées depuis bien longtemps (la notion ayant été inventée en 1924) de par leur applications dans des domaines divers (génération de maillages, interpolation, intermédiaire dans d'autres algorithmes, etc.). Dans les années 2000, le projet CGAL (bibliothèque de géométrie algorithmique sur laquelle travaille entre autres l'équipe GAMBLE dans laquelle j'ai fait mon stage) a été contacté plusieurs fois par des chercheurs qui demandaient l'implémentation d'algorithmes permettant de calculer des triangulations de Delaunay dans des espaces périodiques (pour des applications notamment en astronomie et en cristallographie). Actuellement, l'algorithme implémenté dans CGAL est une adaptation périodique de l'algorithme de Bowyer-Watson, mais nécessitant certaines hypothèses contraignantes (notamment pour des ensembles de peu de points) pour renvoyer une triangulation de Delaunay.

## Le problème étudié

L'objet de mon stage était donc de voir dans quel mesure un autre algorithme, basé sur des transformations locales, pourrait s'adapter aux tores en 2 et 3 dimensions et s'affranchir des contraintes imposées par l'algorithme actuel.

## Les arguments en faveur de sa validité

Pour le cas 2D, mes tuteurs ont déjà quasiment prouvé un résultat de correction encore plus fort, qui concerne un cadre plus général de surfaces 2D, notamment sur des surfaces hyperboliques.

Concernant le cas 3D, comme on va le voir l'algorithme n'est pas correct (du moins sans modifications supplémentaires non étudiées dans le stage par manque de temps) de par des blocages inhérents à la structure périodique de l'espace.

## Le bilan et les perspectives

Les prochaines questions éventuelles pour tenter de régler les problèmes en trois dimensions soulevés lors de mon stage sont celles détaillées en conclusion du rapport : comment trouver une stratégie de résolution dans l'algorithme permettant d'éviter les blocages après l'insertion d'un point ? Ou comment contourner les blocages en autorisant des transformations combinatoires et plus géométriques ?

# Triangulations de Delaunay sur le tore plat en dimensions 2 et 3 Rapport de Stage

Elies Harington

Sous la supervision de Monique Teillaud et Vincent Despre

4 septembre 2019

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Définitions</b>	<b>2</b>
<b>3</b>	<b>Structures de données</b>	<b>4</b>
<b>4</b>	<b>Triangulation de Delaunay sur le tore plat 2D</b>	<b>4</b>
4.1	Algorithme dans le plan euclidien . . . . .	4
4.2	Adaptation au tore 2-dimensionnel . . . . .	8
<b>5</b>	<b>L'algorithme en 3 dimensions</b>	<b>13</b>
5.1	Initialisation . . . . .	13
5.2	Les transformations locales . . . . .	13
5.3	Résultats empiriques . . . . .	16
<b>6</b>	<b>Conclusion</b>	<b>16</b>
<b>A</b>	<b>Code utilisé</b>	<b>16</b>
<b>B</b>	<b>Contre-exemple pour l'algorithme dans <math>\mathbb{T}^3</math></b>	<b>17</b>
	<b>Références</b>	<b>19</b>

## 1 Introduction

J'ai fait mon stage de L3 les mois de Juin et Juillet 2019 au laboratoire LORIA à Nancy sous la direction de Monique Teillaud et Vincent Despre. Il portait sur l'adaptation d'un algorithme de calcul de triangulations de Delaunay à des tores plats en dimensions 2 et 3. De tels algorithmes ont déjà été étudiés dans [CT16] en adaptant un algorithme d'Adrian Bowyer [Bow81], cependant, la correction de cet algorithme dans un espace périodique demande des contraintes spécifiques qui doivent être maintenues le long du calcul (la triangulation doit rester un complexe simplicial, voir plus loin). Le but était donc de savoir si un algorithme basé sur des transformations locales, qui permettrait a priori de s'affranchir de ces contraintes, serait correct dans le cas périodique.

## 2 Définitions

Pour commencer, on définit des notions de géométrie dans l'espace euclidien  $\mathbb{R}^n$  et dans le tore plat  $\mathbb{T}^n$  (défini plus loin dans cette section) nécessaires pour définir et analyser nos algorithmes.

**Définition 2.1** ([Lee00, Ch.5] Simplexe de  $\mathbb{R}^n$ ). Dans  $\mathbb{R}^n$ , un  **$k$ -simplexe** ( $0 \leq k \leq n$ ) est l'enveloppe convexe de  $k+1$  points affinement indépendants. Ces points sont les **sommets** du simplexe. Les **faces** du simplexe sont les enveloppes convexes de sous-parties de ses sommets (ce sont donc aussi des simplexes). Un 0-simplexe est un point, un 1-simplexe un segment (une arête), un 2-simplexe un triangle, un 3-simplexe un tétraèdre.

**Définition 2.2** ([Lee00, Ch.5] Complexe simplicial de  $\mathbb{R}^n$ ). Un **complexe simplicial** de  $\mathbb{R}^n$  est un ensemble de simplexes  $E$  tel que :

- Si  $\sigma \in E$ , toutes les faces de  $\sigma$  appartiennent à  $E$ .
- Si  $\sigma_1, \sigma_2 \in E$ , alors  $\sigma = \sigma_1 \cap \sigma_2$  est une face commune à  $\sigma_1$  et  $\sigma_2$ , et  $\sigma \in E$ .

**Définition 2.3** ([BY98, Ch.17] Triangulation d'un ensemble de points). Une **triangulation** d'un ensemble  $\mathcal{P}$  de  $k$  points de  $\mathbb{R}^n$  est un complexe simplicial d'union l'enveloppe convexe de  $\mathcal{P}$  et dont l'ensemble des sommets (les 0-simplexes) est  $\mathcal{P}$ .

**Définition 2.4** ([BY98, Ch.17] Triangulation de Delaunay). Une triangulation  $\mathcal{T}$  d'un ensemble  $\mathcal{P}$  de  $k$  points de  $\mathbb{R}^n$  est dite **de Delaunay** si elle vérifie :

Pour tout  $n$ -simplexe de  $\mathcal{T}$ , l'intérieur de la boule circonscrite à ses sommets ne contient aucun point de  $\mathcal{P}$ .

Il faut maintenant définir des notions analogues dans le cas périodique. On se place en dimension  $n$  dans le cadre du tore  $\mathbb{T}^n \stackrel{\text{def}}{=} \mathbb{R}^n / \mathbb{Z}^n$  où  $\mathbb{Z}^n$  agit sur  $\mathbb{R}^n$  par translations : si  $t \in \mathbb{Z}^n, x \in \mathbb{R}^n, t \bullet x \stackrel{\text{def}}{=} t + x$ . On parlera du **domaine fondamental** comme étant  $[0; 1]^n$ . On note  $\pi$  la surjection canonique de  $\mathbb{R}^n$  sur  $\mathbb{T}^n$ .

**Définition 2.5** ([CT16] Simplexe de  $\mathbb{T}^n$ ). Soit  $\sigma$  un  $k$ -simplexe de  $\mathbb{R}^n$ . En notant  $\mathcal{P}_\sigma$  l'ensemble des sommets de  $\sigma$ , si  $\pi|_{\sigma \setminus \mathcal{P}_\sigma}$  est injective, on dit que l'image de  $\sigma$  par  $\pi$  est un  $k$ -simplexe de  $\mathbb{T}^n$ .

On dit de plus que  $\sigma$  est un **représentant** de  $\pi(\sigma)$ .

Les points, simplexes et triangulations de  $\mathbb{T}^n$  seront accompagnés d'une tilde pour les différencier de ceux de  $\mathbb{R}^n$  (e.g.  $\tilde{p}, \tilde{\sigma}, \tilde{\mathcal{T}}$ ).

Pour définir une triangulation du tore, les auteurs de [CT16] gardent la même définition que dans  $\mathbb{R}^n$ , i.e. celle de complexe simplicial, dans le but d'adapter un algorithme dont la correction repose fortement sur cette notion [Bow81]. Cependant, cette notion empêche certaines configurations que l'on aimerait se permettre sur le tore, comme les boucles et les arêtes multiples. On donne alors une définition plus large, basée sur le plongement d'une triangulation périodique de  $\mathbb{R}^n$  :

**Définition 2.6** (Triangulation périodique de  $\mathbb{R}^n$ ). On dit qu'une triangulation  $\mathcal{T}$  de  $\mathbb{R}^n$  est périodique si pour tout  $\sigma \in \mathcal{T}$ , pour toute translation  $t \in \mathbb{Z}^n, t \bullet \sigma \in \mathcal{T}$ .

**Définition 2.7** (Triangulation d'un ensemble de points de  $\mathbb{T}^n$ ). On appelle triangulation d'un ensemble de points  $\mathcal{P}$  de  $\mathbb{T}^n$  un ensemble de simplexes  $\mathcal{T}$  de  $\mathbb{T}^n$  tel que  $\pi^{-1}(\mathcal{T})$  soit une triangulation périodique de  $\pi^{-1}(\mathcal{P})$ . Elle est dite de Delaunay si  $\pi^{-1}(\mathcal{T})$  est de Delaunay.

Les algorithmes étudiés se basent sur des transformations locales d'une triangulation qui visent à éliminer des faces qui ne peuvent pas exister dans la triangulation de Delaunay. On définit alors une notion de légalité locale des faces, qui décidera de si oui ou non on doit les éliminer :

**Définition 2.8** (Légalité locale).

- Dans  $\mathbb{R}^n$ , on dit qu'un  $(n - 1)$ -simplexe  $\sigma$  d'une triangulation  $\mathcal{T}$  est **localement légal** lorsque : en notant  $\sigma_1$  et  $\sigma_2$  les deux  $n$ -simplexes incidents à  $\sigma$ , la boule ouverte circonscrite à  $\sigma_1$  ne contient aucun sommet de  $\sigma_2$  [ES96].
- Dans  $\mathbb{T}^n$ , si  $\sigma$  est un  $(n - 1)$ -simplexe d'une triangulation  $\mathcal{T}$  et  $\sigma_0$  un représentant de  $\sigma$  dans  $\pi^{-1}(\mathcal{T})$ , on dit que  $\sigma$  est localement légal si  $\sigma_0$  est localement légal. Cette notion ne dépend pas du choix du représentant.

*Remarque 2.9.* Dans cette définition, puisque la boule circonscrite à  $\sigma_1$  ne contient aucun des points de  $\sigma$  en son intérieur, la condition de légalité locale revient donc à tester l'inclusion d'un seul sommet dans une boule.

Le lemme suivant justifie la pertinence de la notion de légalité locale :

**Lemme 2.10** ([ES96]). *Une triangulation  $\mathcal{T}$  est de Delaunay si et seulement si tous ses  $(n - 1)$ -simplexes sont localement légaux.*

Dans la suite, on représentera des simplexes par la donnée de leurs sommets (par exemple, un triangle serait noté  $pqr$ , et une arête  $pq$ ).

### 3 Structures de données

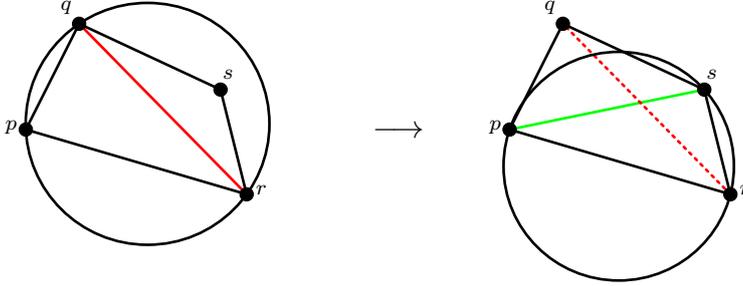
La structure de données retenue pour représenter les triangulations est celle de carte combinatoire [DL14]. Elle permet de représenter des points, des arêtes, des facettes (faces 2-dimensionnelles), et leurs analogues de dimensions supérieures, ainsi que les relations d'adjacence et d'incidence les reliant. La notion élémentaire des cartes combinatoires est celle de **brin** (*dart* en anglais), sorte d'arête orientée entre deux sommets. Pour représenter des triangulations de  $\mathbb{T}^2$  et  $\mathbb{T}^3$ , on utilise une carte combinatoire où on a associé à chaque sommet ses coordonnées dans le domaine fondamental, et à chaque brin la translation du domaine fondamental qui relie ses deux extrémités (par exemple, si un brin relie le sommet  $(0.1, 0.1)$  à lui-même par l'arête  $\pi((0.1, 0.1)(1.1, 2.1))$ , on associera au brin correspondant la translation  $(1, 2)$ ).

## 4 Triangulation de Delaunay sur le tore plat 2D

### 4.1 Algorithme dans le plan euclidien

On présente un algorithme incrémentiel dans le cas du plan euclidien, avant de le généraliser au tore 2D puis au cas 3D. Comme dit précédemment, l'algorithme est basé sur des transformations locales, nommées **bascullements d'arêtes** dans le cas 2D (voir figure 1) [Ber+08].

**Définition 4.1** (Bascullement d'arête dans  $\mathbb{R}^2$ ). Dans une triangulation  $\mathcal{T}$  de  $\mathbb{R}^2$ , si  $qr$  est une arête de  $\mathcal{T}$ , et si les deux triangles incidents  $pqr$  et  $sqr$  sont tels que le quadrilatère  $pqsr$  soit convexe sans trois sommets alignés, alors on définit la triangulation  $\mathcal{T}'$  à partir de  $\mathcal{T}$  où on a remplacé la diagonale  $qr$  du quadrilatère par la diagonale  $ps$ .

FIGURE 1 – Basculement de l'arête illégale  $qr$ .

On peut prouver que si une arête d'une triangulation est localement illégale, alors son quadrilatère environnant est convexe sans trois sommets alignés, on peut donc la basculer :

**Lemme 4.2.** *Si une arête  $qr$  est illégale, alors en reprenant les notations de la définition 4.1, le quadrilatère  $pqsr$  est convexe sans trois sommets alignés.*

Ce résultat bien connu est re-détaillé ici pour mieux introduire les problèmes qui y seront liés dans le cas 3D.

*Démonstration.* (voir schéma figure 2) Pour cela, on montre que  $s$  est dans l'intérieur du cône formé par  $q$ ,  $p$  et  $r$ .

- Soit  $C$  le cercle circonscrit au triangle  $pqr$ .
- Soit  $\alpha$  l'angle entre la droite  $d_1 \stackrel{\text{def}}{=} (pq)$  et la tangente à  $C$  en  $q$ .
- Soit  $\beta$  l'angle entre la droite  $d_2 \stackrel{\text{def}}{=} (pr)$  et la tangente à  $C$  en  $r$ .
- Soit  $H$  le demi-plan contenant  $s_0$  délimité par la droite  $(qr)$ .
- Soit enfin  $D$  l'intersection de  $H$  et du disque intérieur à  $C$ .

$D$  est inclus dans l'intersection de  $H$  avec la zone du plan délimitée par les droites  $d_1$  et  $d_2$ . Puisque  $0 < \alpha$  et  $0 < \beta$ , cette zone est elle-même contenue dans le cône délimité par  $q$ ,  $p$  et  $r$ . Enfin, on sait que  $s$  est contenu dans l'intérieur de  $D$  car  $qr$  est localement illégale.  $s$  est donc bien à l'intérieur du cône.  $\square$

On maintient une liste ListeAretes d'arêtes potentiellement illégales de la manière suivante : lorsqu'on ajoute un point, on le relie aux trois sommets du triangle de la triangulation dans lequel il se trouve, et on rajoute les côtés de ce triangle à ListeAretes. Ensuite, tant que ListeAretes n'est pas vide, on effectue des transformations locales sur les arêtes localement illégales de ListeAretes, et on y ajoute les arêtes que ces transformations ont pu rendre illégales. L'algorithme 1 détaille ce procédé en pseudocode.

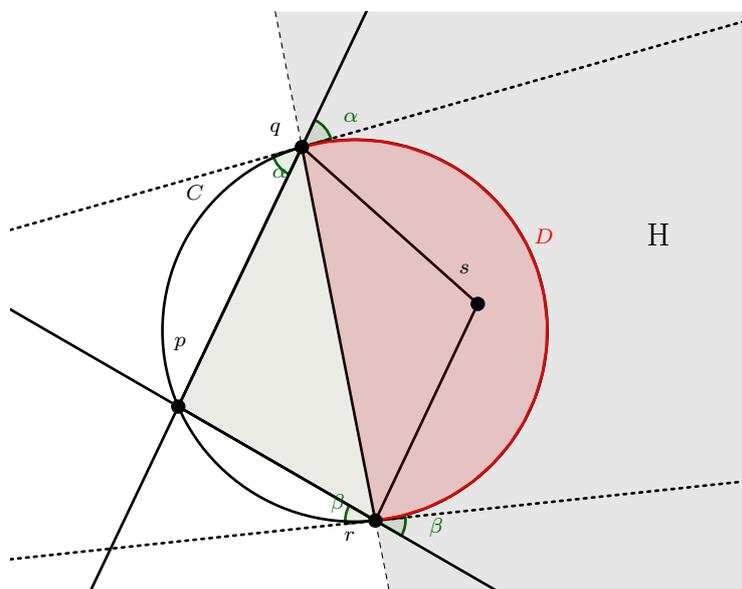


FIGURE 2 – Les triangles incidents à une arête illégale forment un quadrilatère convexe.

La complexité en randomisant l'ordre d'insertion des points est en moyenne  $\mathcal{O}(n \log n)$  pour un ensemble de  $n$  points [Ber+08]. L'initialisation est gérée en considérant un triangle suffisamment grand contenant tous les points qui seront ajoutés ou en considérant un point virtuel à l'infini.

*Remarque 4.3.* L'algorithme ne détaille pas la marche à suivre lorsqu'on doit ajouter un point non pas dans un triangle, mais sur une arête. On appelle ces situations où trois points sont alignés (ou 4 points coplanaires en 3D, etc. en dimensions supérieures) des **cas dégénérés**. Il suffit alors de relier ce point aux sommets opposés des deux triangles incidents à l'arête en question.

La correction de l'algorithme repose sur le fait que l'on va bien basculer toute arête illégale. Dans LEGALISEARETE, on n'ajoute à ListeAretes qu'au plus deux arêtes, quand bien même l'intuition voudrait qu'on y rajoute les 4 côtés du quadrilatère  $pqsr$ ; de même, dans AJOUTPOINT, on n'ajoute pas à ListeAretes les arêtes partant du point venant d'être inséré. On peut en effet prouver que toute arête créée partant du point ajouté restera localement légale jusqu'à l'ajout du prochain point, pas besoin donc de les ajouter à la liste [Ber+08].

---

**Algorithm 1** Delaunay2D

---

```

1: procedure DELAUNAY2D(ListePoints)
2:    $\mathcal{T} \leftarrow \text{TRIANGLEENGLOBANT}(\text{ListePoints})$ 
3:   ListeAretes  $\leftarrow \emptyset$ 
4:   while ListePoints  $\neq \emptyset$  do
5:     Soit  $p \in \text{ListePoints}$ 
6:     AJOUTPOINT( $\mathcal{T}$ , ListeAretes,  $p$ )
7:     ListePoints  $\leftarrow \text{ListePoints} \setminus \{p\}$ 
8:     while ListeAretes  $\neq \emptyset$  do
9:       Soit  $qr \in \text{ListeAretes}$ 
10:      LEGALISEARETE( $\mathcal{T}$ , ListeAretes,  $p$ ,  $qr$ )
11:      ListeAretes  $\leftarrow \text{ListeAretes} \setminus \{qr\}$ 
12:    end while
13:  end while
14:  return  $\mathcal{T}$ 
15: end procedure
16:
17: procedure AJOUTPOINT( $\mathcal{T}$ , ListeAretes,  $p$ )
18:   Soit  $p_1p_2p_3$  le triangle contenant  $p$  dans  $\mathcal{T}$ 
19:    $\mathcal{T} \leftarrow \mathcal{T} \setminus \{p_1p_2p_3\} \cup \{p; pp_1; pp_2; pp_3; pp_1p_2; pp_2p_3; pp_3p_1\}$ 
20:   ListeAretes  $\leftarrow \{p_1p_2; p_2p_3; p_3p_1\}$ 
21: end procedure
22:
23: procedure LEGALISEARETE( $\mathcal{T}$ , ListeAretes,  $p$ ,  $qr$ )
24:   Soit  $sqr$  le triangle adjacent à  $pqr$  dans  $\mathcal{T}$ 
25:    $\triangleright$  La structure de carte combinatoire permet typiquement d'accéder
      à cette relation d'adjacence en  $\mathcal{O}(1)$ .
26:   if  $s$  est dans le disque ouvert circonscrit à  $pqr$  then
27:      $\mathcal{T} \leftarrow \mathcal{T} \setminus \{qr; pqr; sqr\} \cup \{ps; qps; rps\}$ .
28:     ListeAretes  $\leftarrow \text{ListeAretes} \cup \{sq, sr\}$ 
29:   end if
30: end procedure

```

---

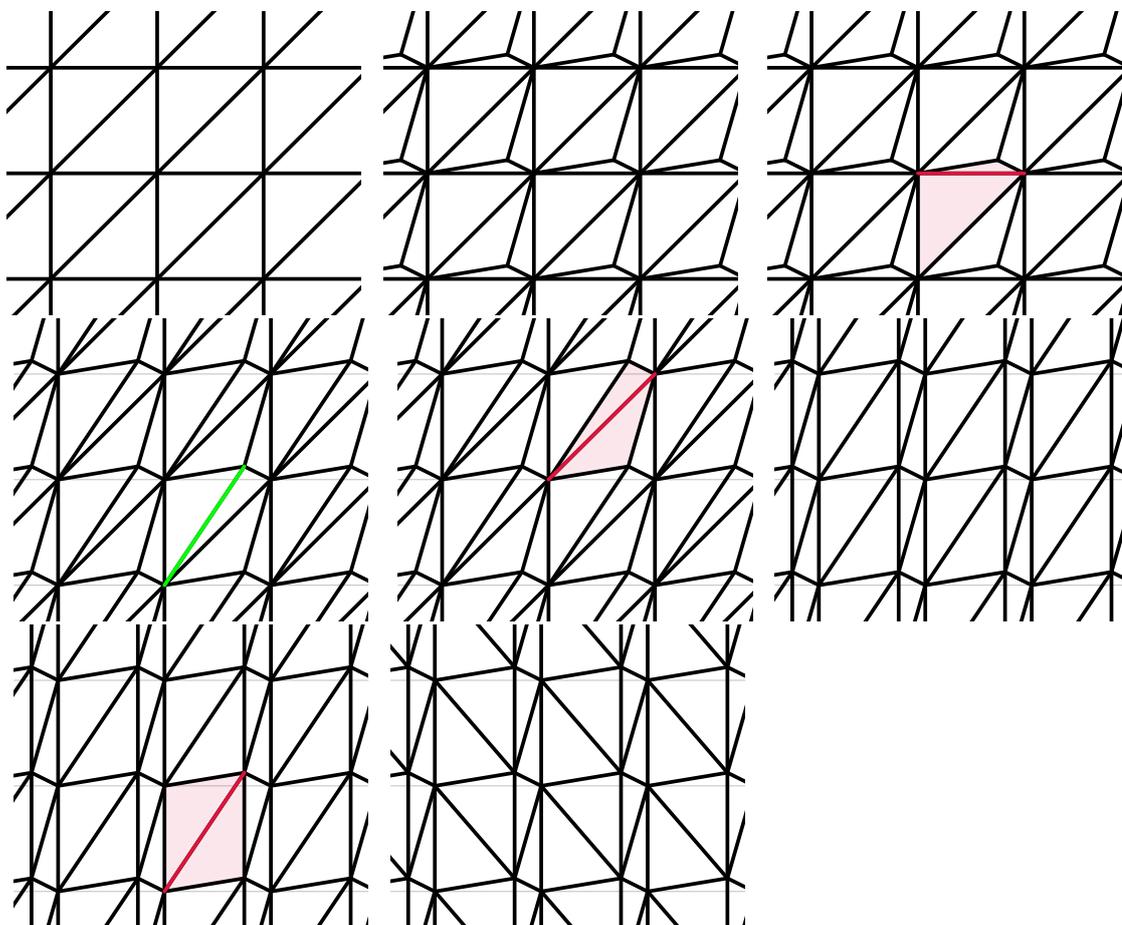


FIGURE 3 – Détail du calcul de la triangulation de Delaunay périodique de l'ensemble de points  $\tilde{\mathcal{P}} = \{\pi(0, 0); \pi(\frac{3}{4}, \frac{1}{8})\}$ . On remarque que l'arête créée par le premier basculement d'arête est détruite par le dernier.

## 4.2 Adaptation au tore 2-dimensionnel

On détaille maintenant l'adaptation de l'algorithme précédent à  $\mathbb{T}^2$ . Les différentes opérations sont intuitivement les mêmes, il faut essentiellement les formaliser dans ce nouveau cadre. Une différence notable remarquée lors de l'implémentation est qu'après un basculement, il faut rajouter les 4 côtés du quadrilatère modifié à la liste des arêtes potentiellement illégales, car les arêtes créées, bien que localement légales, peuvent nécessiter d'être à nouveau détruites pour atteindre une triangulation de Delaunay, comme on peut le voir figure 3. Le schéma principal de l'algorithme reste le même, on change juste les fonctions AJOUTPOINT et LEGALISEARETE pour les adapter au tore. Détaillons ces changements.

**Insertion d'un point** L'insertion du premier point consiste à créer une triangulation du tore composée de deux triangles et de trois arêtes, reliant le seul sommet de la triangulation à ses translatés dans les directions  $(0, 1)$ ,  $(1, 0)$  et  $(1, 1)$ , voir figure 4. Puisque toute triangulation du tore le recouvre entièrement, tout point futur sera contenu dans un triangle ou sur une arête (le cas de l'insertion sur une arête se traite de la même manière que dans le cas non-périodique, et n'est pas non plus détaillé ici). L'ajout d'un point se fait conceptuellement en relevant la triangulation courante  $\tilde{\mathcal{T}}$  dans  $\mathbb{R}^2$ , en y insérant le représentant dans le domaine fondamental du point, et en projetant le changement local dans  $\mathbb{T}^2$ . Le détail est donné en pseudocode : voir algorithme 2.

---

**Algorithm 2** Insertion d'un point dans  $\mathbb{T}^2$

---

```

procedure AJOUTPOINTTORE2D( $\tilde{\mathcal{T}}$ , ListeAretes,  $\tilde{p}$ )
  Soit  $p$  le représentant de  $\tilde{p}$  dans le domaine fondamental
  if  $\tilde{p}$  est le premier point ajouté then
     $p_1 \leftarrow p + (0, 1)$ 
     $p_2 \leftarrow p + (1, 1)$ 
     $p_3 \leftarrow p + (1, 0)$ 
     $\tilde{\mathcal{T}} \leftarrow \pi(\{pp_1; pp_2; pp_3; pp_1p_2; pp_2p_3\})$ 
  else
    Soit  $p_1p_2p_3$  le triangle de  $\pi^{-1}(\mathcal{T})$  contenant  $p_0$ 
     $\tilde{\mathcal{T}} \leftarrow \tilde{\mathcal{T}} \setminus \{\pi(p_1p_2p_3)\} \cup \pi(\{p; pp_1; pp_2; pp_3; pp_1p_2; pp_2p_3; pp_3p_1\})$ 
    ListeAretes  $\leftarrow \pi(\{p_1p_2; p_2p_3; p_3p_1\})$ 
  end if
end procedure

```

---

**Basculement d'arête** De la même manière que pour l'insertion, le basculement se fait en relevant l'arête à basculer dans  $\mathbb{R}^2$ , en la basculant dans ce contexte, puis en reprojétant les triangles affectés sur le tore. Géométriquement, on obtient bien une triangulation du tore, car on a juste modifié l'intérieur d'un quadrilatère avant de le reprojeter, il ne peut donc pas y avoir de conflit avec les autres triangles de la triangulation. On définit comme suit la version périodique de LEGALISEARETE : 3.

La structure de triangulation étant bien conservée par les opérations d'ajout de point et de légalisation, il suffit de montrer que la triangulation obtenue est bien de Delaunay.

**Théorème 4.4.** *L'algorithme incrémentiel de calcul de triangulation de Delaunay sur  $\mathbb{T}^2$  par basculement d'arêtes est correct.*

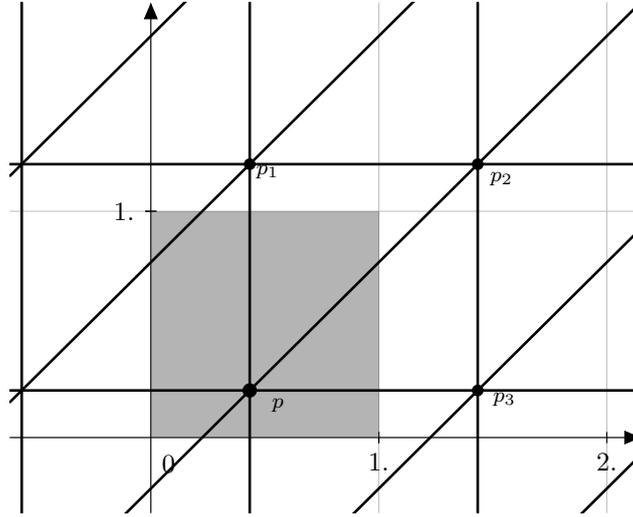


FIGURE 4 – Triangulation de Delaunay d'un seul sommet sur le tore  $\mathbb{T}^2$ . Le domaine fondamental est colorié en gris.

---

**Algorithm 3** Basculement d'arête dans  $\mathbb{T}^2$

---

**procedure** LEGALISEARETETORE2D( $\tilde{\mathcal{T}}$ , ListeAretes,  $\tilde{e}$ )

Soient  $q$  et  $r$  des représentants des extrémités de l'arête  $\tilde{e}$  tels que  $qr$  soit un représentant de  $\tilde{e}$

Soient  $pqr$  et  $sqr$  les deux triangles incidents à  $qr$  dans  $\pi^{-1}(\tilde{\mathcal{T}})$

**if**  $s$  est dans le disque circonscrit à  $pqr$  **then**

▷  $\tilde{e}$  est localement illégale

$\tilde{\mathcal{T}} \leftarrow \tilde{\mathcal{T}} \setminus \pi(\{qr; pqr; sqr; \}) \cup \pi(\{rs; qps; rps\})$

ListeAretes  $\leftarrow$  ListeAretes  $\cup \pi(\{pq; qs; sr; rp\})$

**end if**

**end procedure**

---

*Démonstration.* On peut prouver qu'après un basculement, l'arête créée est localement légale [Ber+08].

On prouve d'abord la correction partielle par récurrence sur le nombre de points :

- Dans le cas d'un seul point, la triangulation renvoyée est de Delaunay par construction.
- Lorsqu'on rajoute un point, on peut faire une observation simple sur les arêtes localement illégales : si les deux triangles incidents à une arête existaient dans la triangulation avant l'ajout du dernier point, alors cette arête est localement légale. Une arête ne peut donc être rendue localement illégale que si un de ses deux triangles incidents a été modifié après l'ajout du point. Or on ne crée ou modifie des triangles que lors de l'insertion d'un point ou d'un basculement d'arête. L'arête créée lors d'un basculement étant localement légale, l'ajout des 4 côtés du quadrilatère à ListeAretes est suffisant pour garantir que toute arête illégale sera basculée durant l'exécution de l'algorithme.

Pour prouver la terminaison, on utilise une fonction de poids qui associe à chaque triangulation un entier, et qui va décroître à chaque basculement. Étant donnée une triangulation  $\tilde{\mathcal{T}}$  de  $\mathbb{T}^2$ , pour tout triangle  $\tilde{\Delta} \in \tilde{\mathcal{T}}$ , on pose  $k(\tilde{\Delta})$  le nombre de sommets de  $\pi^{-1}(\tilde{\mathcal{T}})$  à l'intérieur du disque ouvert circonscrit à un représentant de  $\tilde{\Delta}$  (cette valeur ne dépendant pas du représentant). On pose alors, en notant  $T$  l'ensemble des triangles de  $\tilde{\mathcal{T}}$ ,

$$\mathcal{K}(\tilde{\mathcal{T}}) \stackrel{\text{def}}{=} \sum_{\tilde{\Delta} \in T} k(\tilde{\Delta})$$

Soit  $\tilde{e}$  une arête illégale de  $\tilde{\mathcal{T}}$  et  $\tilde{\mathcal{T}}'$  la triangulation obtenue après avoir basculé  $\tilde{e}$ . On note  $qr$  un représentant de  $\tilde{e}$ ,  $pqr$  et  $sqr$  les triangles incidents à  $qr$  dans  $\pi^{-1}(\tilde{\mathcal{T}})$ , et  $D_1, D_2, D'_1$  et  $D'_2$  les disques ouverts respectivement circonscrits à  $pqr, sqr, qps, rps$  (voir figure 5). En notant  $\mathcal{P}$  l'ensemble des sommets de  $\pi^{-1}(\tilde{\mathcal{T}})$ ,  $A_i = D_i \cap \mathcal{P}$  et  $A'_i = D'_i \cap \mathcal{P}$ , on a la relation suivante :

$$\mathcal{K}(\tilde{\mathcal{T}}') = \mathcal{K}(\tilde{\mathcal{T}}) - |A_1| - |A_2| + |A'_1| + |A'_2|.$$

Il suffit donc de montrer que  $|A_1| + |A_2| > |A'_1| + |A'_2|$ . On utilise pour cela le lemme de géométrie élémentaire suivant (admis) :

**Lemme 4.5.** *L'illégalité de  $qr$  dans  $\pi^{-1}(\tilde{\mathcal{T}})$  implique les inclusions*

$$\begin{aligned} D'_1 \cup D'_2 &\subset D_1 \cup D_2 \\ D'_1 \cap D'_2 &\subset D_1 \cap D_2 \end{aligned}$$

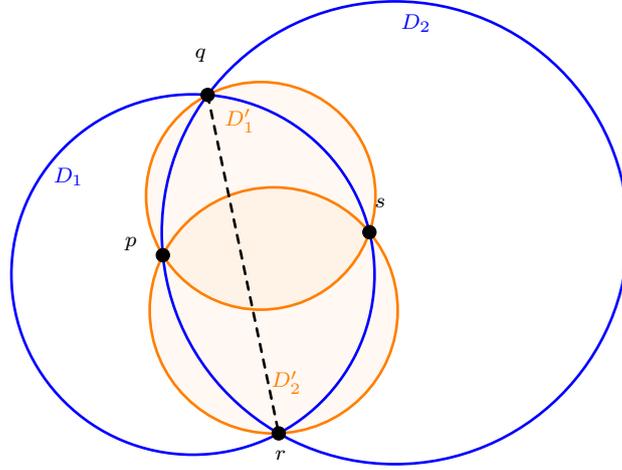


FIGURE 5

$$\begin{aligned} D'_1 \cup D'_2 &\subset D_1 \cup D_2 \\ D'_1 \cap D'_2 &\subset D_1 \cap D_2 \end{aligned}$$

On en déduit directement les mêmes inclusions avec les  $A_i$ , et puisque  $p, s \notin D'_1 \cup D'_2$ , on a même

$$\begin{aligned} A'_1 \cup A'_2 &\subset (A_1 \cup A_2) \setminus \{p; s\} \\ A'_1 \cap A'_2 &\subset A_1 \cap A_2 \end{aligned}$$

Or,  $\{p; s\} \subset A_1 \cup A_2$ , on obtient donc, en passant au cardinal :

$$\begin{aligned} |A'_1| + |A'_2| &= |A'_1 \cup A'_2| + |A'_1 \cap A'_2| \\ &\leq |A_1 \cup A_2| - 2 + |A_1 \cap A_2| \\ &= |A_1| + |A_2| - 2 \end{aligned}$$

On a donc la terminaison.

Puisque l'algorithme termine dans une configuration où tous les arêtes sont localements légales, le lemme 2.10 permet de conclure quant à la correction.  $\square$

## 5 L'algorithme en 3 dimensions

### 5.1 Initialisation

En dimension 3, une triangulation de Delaunay d'un seul point est constituée de 7 arêtes (les projections des segments formés par un représentant du point et ses translatés dans les directions  $(0, 0, 1)$ ,  $(0, 1, 0)$ ,  $(0, 1, 1)$ ,  $(1, 0, 0)$ ,  $(1, 0, 1)$ ,  $(1, 1, 0)$ , et  $(1, 1, 1)$ ) et de 6 tétraèdres (tous incidents à l'arête correspondant à la translation  $(1, 1, 1)$ ).

### 5.2 Les transformations locales

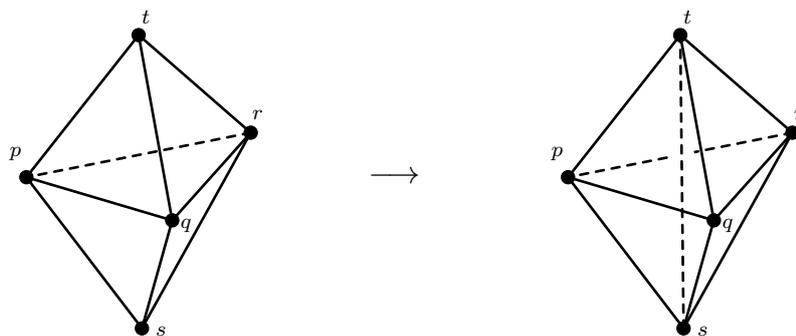
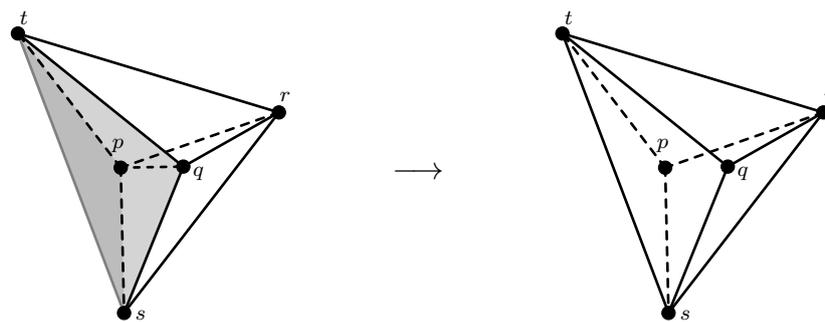
En dimension 3, les simplexes de dimension maximale sont les tétraèdres. Une première difficulté rencontrée pour définir les transformations locales en 3D est l'absence d'un équivalent au lemme 4.2. En effet, en 3D, on peut rencontrer une face localement illégale dont l'union des tétraèdres incidents forme un polyèdre non convexe. On définit alors deux types de transformations locales :

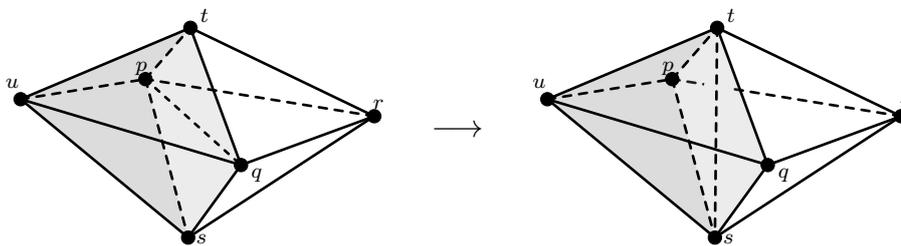
**Définition 5.1.** Soit  $\mathcal{T}$  une triangulation dans  $\mathbb{R}^3$  et  $pqr$  une facette de  $\mathcal{T}$ , et  $pqrs$ ,  $pqrt$  ses deux tétraèdres incidents.

- Si  $pqrs \cup pqrt$  est convexe, alors on dit qu'on peut **basculer**  $2 \rightarrow 3$  la facette  $pqr$ , et la triangulation obtenue après cette transformation est  $\mathcal{T}$  où on a relié les sommets  $s$  et  $t$  et où on a donc coupé  $pqrs \cup pqrt$  en  $pqst$ ,  $qrst$  et  $rpst$  (voir figure 6).
- Dans le cas contraire, si  $pqst$ ,  $qrst$  ou  $rpst$  est un tétraèdre de  $\mathcal{T}$  - on suppose que c'est  $pqst$  sans perdre de généralité à renommage près - alors on dit qu'on peut **basculer**  $3 \rightarrow 2$  l'arête  $pq$ . La triangulation obtenue après cette transformation est  $\mathcal{T}$  où on a remplacé l'arête  $pq$  et ses tétraèdres incidents par la facette  $str$  et les tétraèdres  $pstr$  et  $qstr$  (voire figure 7).
- Si aucun des deux cas précédents ne s'applique, on dit qu'on ne peut pas basculer.

Détecter le type de basculement à effectuer pour supprimer une facette illégale peut se faire en temps constant en utilisant la structure de carte combinatoire. Dans le cas périodique, on définit les mêmes transformations de manière analogue, en relevant la facette à basculer dans  $\mathbb{R}^3$  et en plongeant le changement dans  $\mathbb{T}^3$ , comme pour le cas 2D.

Dans  $\mathbb{R}^3$ , l'algorithme incrémentiel analogue à celui déjà étudié va maintenir une liste de facettes potentiellement illégales, et tenter de les basculer les unes après les autres. S'il ne peut pas basculer une facette illégale, il se

FIGURE 6 – Basculement  $2 \rightarrow 3$  de la facette  $pqr$ FIGURE 7 – Basculement  $3 \rightarrow 2$  de l'arête  $pq$

FIGURE 8 – Basculement  $4 \rightarrow 4$  de l'arête  $pq$ 

contente de la retirer de la liste. La non-basculabilité de certaines facettes d'une triangulation fait que l'algorithme naïf qui, étant donnée une triangulation, bascule toutes les faces illégales possibles, ne va pas forcément arriver à une triangulation de Delaunay [ES96]. Cependant, les auteurs de [ES96] prouvent que l'algorithme incrémentiel, lui, reste correct dans  $\mathbb{R}^3$  (et même dans  $\mathbb{R}^n$ ), en se basant sur la structure de ce qu'ils appellent les **facettes de lien** (dans l'algorithme incrémentiel, pendant l'étape des basculements qui suivent l'insertion d'un point  $p$ , les facettes de lien sont les facettes extérieures de  $\text{star}(p)$ , l'ensemble des simplexes incidents au point  $p$ ). Cependant, dans  $\mathbb{T}^3$ , cet ensemble n'a pas nécessairement les propriétés structurelles exploitées par [ES96] pour prouver la correction de l'algorithme (on peut par exemple trouver des situations où toutes les facettes de la triangulation sont des facettes de lien).

Le développement d'une implémentation de l'algorithme dans  $\mathbb{T}^3$  en C++ avec le package "Combinatorial Maps" de la bibliothèque CGAL nous a permis de trouver des contre-exemples, pour lesquels l'algorithme incrémentiel ne marche pas dans  $\mathbb{T}^3$ . Le premier obstacle rencontré vient du cas où les points  $t$ ,  $s$ ,  $p$  et  $q$  de la figure 6 sont coplanaires lorsqu'on essaie de basculer la facette  $pqr$ . Dès lors, aucun des deux basculements n'est possible. Cela ne pose pas de problème dans [ES96] car les cas dégénérés (4 points coplanaires, entre autres) y sont ignorés. Cependant, dans  $\mathbb{T}^n$ , il est plus difficile d'ignorer ces dégénérescences, car la périodicité de l'espace fait naturellement apparaître ce genre de configurations. Pour palier à ce problème, on a mis au point un nouveau type de basculement, consistant dans un tel cas à chercher deux autres tétrèdres autour de l'arête  $pq$ , notés  $pqtu$  et  $pqsu$ , de manière à ce que l'enveloppe convexe de  $pqrst$  soit un octaèdre. En remplaçant dans cet octaèdre l'arête  $pq$  par l'arête  $ts$ , on passe d'un partitionnement en 4 tétraèdres de l'octaèdre à un autre partitionnement en quatre tétraèdres. Ce basculement est donc appelé **basculement**  $4 \rightarrow 4$  (voir figure 8).

Après avoir fait effectué un quelconque basculement, on rajoute comme dans le cas 2D à une file les facettes ayant pu être rendues localement illégales

par cette transformation.

### 5.3 Résultats empiriques

Avec l’implémentation de ce nouveau basculement, on a finalement pu mettre en évidence des contre-exemples plus profonds où l’algorithme crée une triangulation dans laquelle il reste des facettes localement illégales, mais dont aucune n’est basculable, quel que soit le basculement envisagé (voir annexe B).

Le premier bilan est donc la non-correction de l’algorithme incrémentiel basé sur les transformations locales dans  $\mathbb{T}^3$ . Une question qui se pose alors naturellement est “Existe-t-il, pour tout ensemble de points, un ordre dans lequel effectuer les insertions et/ou les basculements pour arriver de manière sûre à une triangulation de Delaunay?”. Nous avons donc effectué des tests en randomisant l’ordre des basculements, pour des ensembles de points et des ordres d’insertion fixés, et il semble toujours exister une stratégie permettant d’arriver à une triangulation de Delaunay. Plus quantitativement, sur une cinquantaine de couples de points choisis au hasard, en testant pour chaque couple 50 ordres de basculement aléatoires, on atteint plus de 84% de triangulations de Delaunay, et ce pour chaque couple.

## 6 Conclusion

Le bilan global est donc la correction de l’algorithme dans  $\mathbb{T}^2$ , et sa non-correction dans  $\mathbb{T}^3$ . Les pistes à explorer dans le futur comptent entre autres la recherche d’un moyen de trouver à coup sûr une stratégie de basculements permettant d’arriver à un Delaunay (étant donné l’apparente forte probabilité d’y arriver en randomisant), ou encore chercher à faire des basculements purement combinatoires (n’ayant donc pas nécessairement de réalisation géométrique).

Je tiens à remercier chaleureusement mes tuteurs M. D Vincent et Mme. T Monique pour leur accueil, leur soutien et leur encadrement dans cette première expérience de recherche, et également à mes collègues de bureau et à toute l’équipe GAMBLE grâce à laquelle j’ai pu passer un superbe stage.

## A Code utilisé

Les algorithmes ont été implémentés en C++ avec la bibliothèque CGAL (principalement pour ses structures de calcul exact et le package de cartes

combinatoires). Ils sont trouvables à l'adresse suivante : <http://perso.crans.org/edlyr/stage/>

## B Contre-exemple pour l'algorithme dans $\mathbb{T}^3$

Voici le détail d'un contre-exemple à deux points trouvé par notre implémentation pour l'algorithme dans  $\mathbb{T}^3$  :

Initialisation de la triangulation avec le point  $\pi((0, 0, 0))$ .

Insertion du point  $\pi((13/64, 9/64, 33/64))$  dans le tétraèdre

$$\pi((0, 0, 0), (1, 1, -1), (1, 0, -1), (0, 0, -1)).$$

La facette  $\pi((0, 0, 0), (-1, 0, 1), (-1, 0, 0))$  est illégale.

On la bascule  $2 \rightarrow 3$ .

La facette  $\pi((0, 0, 0), (-1, -1, 1), (-1, 0, 0))$  est illégale.

On la bascule  $2 \rightarrow 3$ .

La facette  $\pi((0, 0, 0), (-1, -1, 1), (-1, -1, 0))$  est illégale.

On la bascule  $2 \rightarrow 3$ .

La facette  $\pi((13/64, 9/64, 33/64), (1, 0, 0), (0, -1, 0))$  est illégale.

On bascule  $4 \rightarrow 4$  l'arête  $\pi((0, 0, 0), (13/64, 73/64, 33/64))$ .

La facette  $\pi((0, 0, 0), (0, -1, 0), (0, -1, 1))$  est illégale.

On la bascule  $2 \rightarrow 3$ .

La facette  $\pi((13/64, 9/64, 33/64), (0, 1, 0), (0, 0, 1))$  est illégale et non basculable.

La facette  $\pi((0, 0, 0), (-1, -1, 1), (0, -1, 1))$  est illégale et non basculable.

La facette  $\pi((0, 0, 0), (0, -1, 0), (-1, -1, 1))$  est illégale.

On la bascule  $2 \rightarrow 3$ .

La facette  $\pi((13/64, 9/64, 33/64), (0, 1, 0), (-1, 0, 1))$  est illégale.

On bascule  $4 \rightarrow 4$  l'arête  $\pi((0, 0, 0), (77/64, 9/64, -31/64))$ .

La facette  $\pi((0, 0, 0), (13/64, 9/64, -31/64), (1, 0, -1))$  est illégale.

On bascule  $3 \rightarrow 2$  l'arête  $\pi((0, 0, 0), (-1, 0, 1))$ .

La facette  $\pi((0, 0, 0), (1, 1, -1), (1, 1, 0))$  est illégale.

On la bascule  $2 \rightarrow 3$ .

La facette  $\pi((0, 0, 0), (-1, -1, 1), (-51/64, 9/64, 33/64))$  est illégale.

On bascule  $3 \rightarrow 2$  l'arête  $\pi((0, 0, 0), (-1, -1, 1))$ .

La facette  $\pi((13/64, 9/64, 33/64), (1, 1, 0), (1, 0, 1))$  est illégale.

On la bascule  $2 \rightarrow 3$ .

La facette  $\pi((0, 0, 0), (-51/64, 9/64, 33/64), (0, -1, 1))$  est illégale et non basculable.

La facette  $\pi((13/64, 9/64, 33/64), (0, 1, 0), (0, 0, 1))$  est illégale et non basculable.

La facette  $\pi((0, 0, 0), (0, 1, -1), (-51/64, 73/64, -31/64))$  est illégale et non basculable.

La facette  $\pi((13/64, 9/64, 33/64), (0, -1, 1), (1, -1, 1))$  est illégale et non basculable.

La facette  $\pi((0, 0, 0), (-1, 0, 0), (-51/64, 73/64, -31/64))$  est illégale et non basculable.

La facette  $\pi((0, 0, 0), (1, 0, 0), (13/64, 73/64, -31/64))$  est illégale et non basculable.

La facette  $\pi((0, 0, 0), (13/64, 9/64, -31/64), (0, 1, -1))$  est illégale et non basculable.

La facette  $\pi((0, 0, 0), (13/64, -55/64, 33/64), (0, -1, 1))$  est illégale et non basculable.

La facette  $\pi((13/64, 9/64, 33/64), (0, 0, 0), (0, -1, 1))$  est illégale et non basculable.

La facette  $\pi((0, 0, 0), (13/64, 9/64, 33/64), (0, -1, 1))$  est illégale et non basculable.

La triangulation obtenue n'est pas de Delaunay, et aucune des faces illégales restantes n'est basculable :

$\pi((0, 0, 0), (0, -1, 1), (-51/64, 9/64, 33/64))$

$\pi((0, 0, 0), (0, 1, -1), (13/64, 9/64, -31/64))$

$\pi((13/64, 9/64, 33/64), (0, -1, 1), (1, -1, 1))$

$\pi((0, 0, 0), (13/64, 73/64, -31/64), (0, 1, -1))$

## Références

- [Bow81] Adrian BOWYER. “Computing Dirichlet tessellations”. In : *The Computer Journal* (1981).
- [ES96] Herbert EDELSBRUNNER et N.H. SHAH. “Incremental topological flipping works for regular triangulations”. In : *Algorithmica* (1996).
- [BY98] Jean-Daniel BOISSONNAT et Mariette YVINEC. *Algorithmic Geometry*. Cambridge University Press, 1998.
- [Lee00] John M. LEE. *Introduction to Topological Manifolds*. Springer-Verlag, 2000. ISBN : 978-1-4419-7940-7.
- [Ber+08] Mark de BERG, Otfried CHEONG, Marc van KREVELD et Mark OVERMARS. *Computational Geometry*. Springer, 2008. ISBN : 9783540779735.
- [DL14] G. DAMIAND et P. LIENHARDT. *Combinatorial Maps : Efficient Data Structures for Computer Graphics and Image Processing*. A K Peters/CRC Press, 2014.

- [CT16] Manuel CAROLI et Monique TEILLAUD. “Delaunay triangulations of closed Euclidean d-orbifolds”. In : *Discrete and Computational Geometry* (2016).