

1113

David-Olivier Jaquet

**ENUMERATION COMPLETE  
DES CLASSES DE  
FORMES PARFAITES  
EN DIMENSION 7**

Thèse de doctorat

**ENUMERATION COMPLETE**  
**DES CLASSES DE**  
**FORMES PARFAITES**  
**EN DIMENSION 7**

Thèse de doctorat de **David-Olivier Jaquet**,  
effectuée sous la direction du Professeur **François Sigrist**,  
à l'Institut de Mathématiques et d'Informatique de l'Université de Neuchâtel.

L'examen de soutenance de thèse a eu lieu en présence des professeurs  
**François Sigrist**, professeur à l'Université de Neuchâtel, directeur de thèse,  
**Ulrich Suter**, professeur à l'Université de Neuchâtel,  
**Jacques Martinet**, professeur à l'Université de Bordeaux I,  
**Joseph Oesterlé**, professeur à l'Institut Poincaré (Paris VI).

# IMPRIMATUR POUR LA THÈSE

Énumération complète des classes de formes  
parfaites en dimension 7

de Monsieur David-Olivier Jaquet

---

UNIVERSITÉ DE NEUCHÂTEL

FACULTÉ DES SCIENCES

La Faculté des sciences de l'Université de Neuchâtel  
sur le rapport des membres du jury,

MM. les professeurs F. Sigrist, U. Suter  
J. Martinet (Bordeaux) et J. Oesterlé (Paris)

autorise l'impression de la présente thèse.

Neuchâtel, le 18 juin 1991

Le doyen :

  
Cl. Mermod

## Avant-propos

*Fruit de plusieurs années de formation, résultat de recherches personnelles, une thèse de doctorat est tout à la fois une étape importante et un nouveau départ.*

*J'aimerais remercier aujourd'hui tous ceux qui m'ont accompagné durant cette période enrichissante, tous ceux qui, directement ou indirectement, m'ont permis de réaliser ma thèse de doctorat, soit par les connaissances qu'ils m'ont permis d'acquérir, soit par le soutien qu'ils m'ont apporté, soit par l'enthousiasme qu'ils ont su me transmettre.*

*Mes remerciements s'adressent tout particulièrement à mon directeur de thèse, Monsieur François Sigrist, professeur à l'Université de Neuchâtel. François Sigrist a toujours manifesté à mon égard une très grande disponibilité. Je désire lui faire part ici de ma reconnaissance pour ses conseils, ses suggestions et son précieux soutien. Les échanges d'idées que j'ai eu la chance de partager avec lui représentent, pour moi, un apport essentiel.*

*Je remercie également les trois experts, Messieurs les Professeurs Suter de Neuchâtel, Martinet de Bordeaux et Oesterlé de Paris. Les contacts avec chacun d'eux ont été très enrichissants.*

*L'équipe du Centre de calcul de l'Université de Neuchâtel, son directeur Monsieur Randoald Corfu et, en particulier, Monsieur Jean-Pierre Maradan ont apporté, par leur compétence, un soutien technique indispensable.*

*Ma gratitude s'adresse tout spécialement à mes parents qui n'ont cessé de me stimuler tout au long de mes études.*

*Enfin, j'aimerais faire un clin d'oeil à Daphné qui a partagé avec moi, de l'intérieur, toute l'élaboration de ma thèse, dans ses moments d'incertitude, comme dans les instants privilégiés qui suivent une découverte.*

## 1. Introduction :

La classification des formes parfaites en dimension inférieure à sept est un problème résolu depuis plus de trente ans. En 1831 déjà, Gauss montre que le réseau cubique à faces centrées est l'unique réseau absolument extrême en dimension trois. Dans cette dimension, toute forme parfaite est équivalente à  $A_3$ ; il n'existe donc qu'une seule classe de formes parfaites et ces formes sont absolument extrêmes. Pour les dimensions quatre et cinq, il faut attendre les travaux de Korkine et Zolotareff en 1877. En dimension quatre, on observe deux classes de formes parfaites ( $A_4$  et  $D_4$ ). En dimension cinq, leur nombre s'élève à trois. Quatre-vingts années plus tard, en 1957, Barnes démontre qu'il n'y a que sept classes de formes parfaites à six variables, dont une qui n'est pas extrême, conformément à ce que pensait Voronoï. A la fin de son article, Barnes prévoit que ses méthodes ne permettront certainement pas de traiter la dimension sept, vu la complexité du problème [Ba 3].

La démonstration de Barnes se base sur le seul algorithme général qui garantit l'énumération complète des classes de formes parfaites en dimension  $n$ , à savoir le fameux algorithme de Voronoï [Vo 1]. Sa complexité croît si rapidement en fonction de la dimension, que seule l'utilisation d'ordinateurs permettait d'espérer pouvoir appliquer cet algorithme à la dimension sept.

En 1963, Scott tente de traiter certaines formes, en dimension sept, à l'aide de méthodes semblables à celles de Barnes, afin d'appliquer l'algorithme de Voronoï [Sc 1]. Il y réussit partiellement, mais ses résultats contiennent des erreurs. Jusqu'en 1971, on ne connaît que vingt-deux classes de formes parfaites à sept variables. Les travaux de Stacey permettent d'en découvrir onze nouvelles. Ces travaux sont basés sur des théorèmes de Watson. En fait, Stacey génère un grand nombre de formes parfaites. Mais elle n'arrive pas à décider si deux formes données sont équivalentes ou non. Pour effectuer un tri, elle considère un système de trois invariants; lorsque deux formes coïncident sur ces invariants, elle n'en retient qu'une.

La liste obtenue par Stacey fut considérée comme très vraisemblablement exhaustive. Mais rien ne permettait d'exclure, à priori, que deux formes parfaites non équivalentes coïncident sur ce système de trois invariants. Dès lors, le problème fut de montrer que cette liste de trente-trois formes était effectivement exhaustive.

Dans un article paru en 1988, Conway et Sloane affirment que seules vingt-sept de ces formes ont été traitées dans l'optique de Voronoï et ils qualifient ces résultats d'incertains [CS 2]. Mes programmes, écrits en LISP, m'ont permis de traiter dans un premier temps trente-deux de ces trente-trois formes (pour  $D_7$ , cf [JS 1]). Une version plus complexe, que j'ai développée ultérieurement, a rendu possible le traitement de la dernière, à savoir  $E_7$ .

Mes travaux, comme ceux de Barnes, se basent sur l'algorithme de Voronoï. Par contre, les méthodes utilisées pour caractériser les faces d'un domaine sont entièrement nouvelles.

Voronoï, dans son étude générale de  $D_n$ , emploie certaines astuces très particulières qui permettent d'exhiber une liste suffisante de faces. De même, les travaux de Barnes contiennent très souvent des astuces de ce genre qui, dans un certain sens, localisent le problème et, par conséquent, le simplifient. Dans sa version la plus complexe, mon programme automatise, d'un certain point de vue, cette découverte d'astuces particulières.

Ces mêmes programmes, appliqués aux dimensions inférieures à sept, m'ont permis, entre autre, de confirmer très rapidement les résultats de Barnes et de donner une description plus détaillée de la manière dont s'imbriquent les domaines de Voronoï. En particulier - et ceci est un résultat nouveau - le domaine associé à  $E_6$  possède exactement 38 124 faces qui se répartissent, conformément aux résultats de Barnes, en 11 orbites sous l'action du groupe des automorphismes de  $E_6$ .

L'ordinateur utilisé, une VAX 8530, a travaillé pendant environ cent vingt jours "CPU" pour traiter complètement la dimension sept. Les résultats de ces calculs démontrent le théorème suivant :

### Théorème :

*Il n'existe que trente-trois classes de formes parfaites à sept variables.*

On notera, au passage, que ces calculs ont permis de décrire complètement le domaine associé à  $E_7$ ; ce domaine possède exactement 79 900 912 faces qui se répartissent en 157 orbites sous l'action du groupe des automorphismes de  $E_7$ .

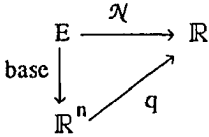
Mes recherches avaient pour but - et le but est atteint - la classification des formes parfaites à sept variables. Quelles sont les motivations d'une telle classification?

Toute forme extrême étant parfaite, on déduit de la connaissance des classes de formes parfaites dans une dimension, celle des classes de formes extrêmes, donc des réseaux les plus denses. En particulier, cela permet de calculer la valeur exacte de  $\gamma_n$ .

Si l'on possède une description détaillée du domaine de Voronoï associé à chaque représentant des classes de formes parfaites, on en déduit un algorithme général de réduction des formes quadratiques définies positives, algorithme décrit précisément par Voronoï.

2. **Définitions :**

Soit E, un espace vectoriel réel de dimension n, muni d'un produit scalaire  $\langle \cdot | \cdot \rangle$  et  $q : \mathbb{R}^n \rightarrow \mathbb{R}$ , une forme quadratique définie positive. A isométrie près, il existe une unique base de E pour laquelle q soit l'application "norme au carré", c'est-à-dire pour laquelle le diagramme suivant soit commutatif :



où  $\mathcal{N}(v) = \langle v | v \rangle$ .

fig. 1

Si Q représente la matrice de Gram de cette base,  $\forall x \in \mathbb{R}^n, q(x) = x^t Q x$ ; on dira que le réseau engendré par les vecteurs de cette base est un réseau associé à q.

Si une autre base de E engendre le même réseau, la matrice S de changement de base appartient à  $GL_n(\mathbb{Z})$ .

Deux réseaux sont équivalents s'ils sont semblables (équivalence "naturelle"). Les réseaux de E associés à q étant isométriques sont donc équivalents. On en déduit une relation d'équivalence entre formes quadratiques définies positives, compatible avec l'équivalence entre réseaux : deux formes q et q' sont équivalentes s'il existe  $S \in GL_n(\mathbb{Z})$ , telle que q soit positivement proportionnelle à q'oS.

Ainsi, on obtient une bijection entre classes de formes quadratiques définies positives et classes de réseaux.

Par définition, le minimum de q (et de Q) est :

$$\min q = \min q(x) = \min x^t Q x = \min Q \quad \text{où } x \in \mathbb{Z}^n, x \neq 0.$$

Les paires de vecteurs  $\pm v \in \mathbb{Z}^n$  vérifiant  $q(v) = \min q$  sont appelées paires de vecteurs minimaux de q (et de Q). Géométriquement, les vecteurs minimaux de q correspondent aux points d'un réseau associé à q, les plus proches de l'origine.

Le nombre  $\text{disc } q = \det Q$  est appelé discriminant de q. La racine de  $\text{disc } q$  est égale au volume d'une maille d'un réseau associé à q.

On définit l'application suivante, dite invariant d'Hermite :

$$\mu : q \mapsto \mu(q) = \frac{\min q}{\sqrt[n]{\text{disc } q}}.$$

$\mu$  est invariante sur les classes de formes quadratiques définies positives; elle est directement liée à la densité des empilements de sphères associés aux réseaux.

On définit  $\gamma_n$ , appelée constante d'Hermite, par  $\gamma_n = \sup \mu(q)$ , lorsque  $q$  varie dans l'ensemble des formes quadratiques définies positives à  $n$  variables. On démontre que  $\mu$  atteint ses maxima; en particulier, la valeur de  $\gamma_n$  est atteinte.

Les formes quadratiques définies positives correspondant aux maxima locaux de la fonction  $\mu$  sont dites extrêmes, celles qui vérifient  $\mu(q) = \gamma_n$ , absolument extrêmes.

Une forme quadratique définie positive est dite parfaite si elle est entièrement caractérisée par la valeur de son minimum et l'ensemble de ses vecteurs minimaux.

Considérons maintenant l'espace des formes quadratiques, définies ou non, à  $n$  variables. Cet espace vectoriel est isomorphe à  $\text{Sym}_n(\mathbb{R})$  qui, muni du produit scalaire  $(X, Y) \mapsto \text{trace}(XY)$ , est un espace euclidien de dimension  $N = \frac{n(n+1)}{2}$ : l'espace de Voronoï, noté  $\mathcal{U}$ .

Soit  $q$  une forme parfaite, indiquons ses paires de vecteurs minimaux:  $\pm v_k$ ,  $k = 1, \dots, s$ . A chaque paire  $\pm v_k$  correspond une forme quadratique positive  $q_k(x)$ :  $q_k: x \mapsto q_k(x) = (v_k^t x)^2$ , et donc un point  $v_k v_k^t$  dans l'espace de Voronoï. Appelons  $d_k$  la demi-droite, issue de l'origine et passant par  $v_k v_k^t$ . L'enveloppe convexe des  $d_k$  est un cône convexe appelé domaine de Voronoï associé à  $q$  (et à  $Q$ ).

On appelle face du domaine tout cône convexe de codimension  $t$  obtenu en intersectant le domaine avec un de ses hyperplans d'appui.

Plus généralement, une d-face sera un cône convexe de dimension  $d \geq 1$  obtenu en intersectant des faces du domaine. Les 1-faces sont les arêtes du domaine.

Si  $q(x) = x^t Q x$ , on définit  $q^{\text{ad}}$  par  $q^{\text{ad}}: x \mapsto q^{\text{ad}}(x) = x^t Q^{\text{ad}} x$  où  $Q^{\text{ad}}$  est la matrice adjointe de  $Q$ . On dira que  $q$  est eutactique si  $q^{\text{ad}}$  peut s'exprimer comme combinaison linéaire à coefficients strictement positifs des  $q_k$ . Géométriquement, si  $q$  est parfaite, cela revient à demander que  $Q^{\text{ad}}$  soit un point intérieur du domaine associé à  $q$ .

Voronoï démontre un critère, maintenant classique, pour les formes extrêmes: *une forme quadratique définie positive est extrême si et seulement si elle est parfaite et eutactique.*

### 3. La fonction d'Hermite $\mu$ :

La recherche des réseaux les plus denses en dimension  $n$  revient à caractériser les maxima de la fonction  $\mu$ . Pour  $n$  fixé, la fonction  $\mu$  est bornée; Minkowski a montré, par exemple, que pour  $n$  donné supérieur à 1,  $\gamma_n$  est inférieur à  $n$ . Il emploie pour cela des méthodes liées à la géométrie des nombres.

On démontre que, modulo les homothétiques, les maxima de  $\mu$  sont isolés et, qu'à équivalence près, il n'y a qu'un nombre fini de formes extrêmes. Historiquement, c'est la recherche de ces classes de formes extrêmes qui a introduit la notion de formes parfaites. La classification des formes extrêmes fut ramenée à la classification des formes parfaites.

On ne connaît les valeurs exactes de  $\gamma_n$  que pour  $n \leq 8$  :

$$\begin{array}{llll} \gamma_1 = 1 & \gamma_2 = \sqrt{\frac{4}{3}} & \gamma_3 = \sqrt[3]{2} & \gamma_4 = \sqrt{2} \\ \gamma_5 = \sqrt[5]{8} & \gamma_6 = \sqrt[6]{\frac{64}{3}} & \gamma_7 = \sqrt[7]{64} & \gamma_8 = 2 \end{array}$$

Les valeurs de  $\gamma_6$ ,  $\gamma_7$  et  $\gamma_8$  furent déterminées par Blichfeldt, voici près de soixante ans :  $\gamma_6$  en 1925,  $\gamma_7$  en 1929 et  $\gamma_8$  en 1935 [Bl 1, 2, 3]. En 1944, Mordell montra à l'aide de l'inégalité qui porte son nom que, de la connaissance de  $\gamma_7$ , on peut déduire celle de  $\gamma_8$  [Mo 1].

En 1966, Watson annonça qu'il était capable de vérifier les calculs de Blichfeldt... et démontra l'unicité des formes absolument extrêmes en dimension 6, 7 et 8 [Wa 1].

Plus récemment, en 1980, Vetchinkin confirma les résultats de Blichfeldt et de Watson dans un article très détaillé [Ve 1].

L'énumération complète des classes de formes parfaites à sept variables a pour corollaire une confirmation immédiate des valeurs de  $\gamma_7$  et  $\gamma_8$ .

#### 4. Généralités sur les convexes :

##### 4.1. Propriétés fondamentales :

Soit  $E$  un espace euclidien et  $l_1, \dots, l_t$  des vecteurs de  $E$ . On définit  $\mathcal{C}$  - cône convexe de  $E$  - par  $\mathcal{C} = \{x \in E \mid l_i \cdot x \geq 0, i = 1, \dots, t\}$ .

##### Théorème 1 :

$\mathcal{C}$  est une variété de dimension maximale dans  $E$  si et seulement si les  $l_i$  sont linéairement positivement indépendants, c'est-à-dire si  $\sum_{i=1}^t \rho_i l_i = 0$ , avec  $\rho_i \geq 0$ , implique  $\rho_i = 0$  pour tout  $i$ .

##### Théorème 2 :

Supposons  $\mathcal{C}$  de dimension maximale; après élimination éventuelle des  $l_i$  pouvant s'exprimer comme combinaison linéaire positive des autres  $l_j$  ( $j \neq i$ ), ceux qui restent définissent les hyperplans d'appui de  $\mathcal{C}$  qui contiennent les faces de  $\mathcal{C}$ , cônes convexes de codimension 1.

##### Théorème 3 :

Supposons  $\mathcal{C}$  de dimension maximale. Si  $(l_i \cdot x = 0 \forall i \Rightarrow x = 0)$ , alors  $\mathcal{C}$  est un cône convexe saillant; il possède des arêtes (représentées par  $a_1, a_2, \dots, a_s$ ), et tout point de  $\mathcal{C}$  peut s'exprimer comme combinaison linéaire positive des  $a_k$ , c'est-à-dire :  $\mathcal{C} = \{x \in E \mid x = \sum_{k=1}^s \rho_k a_k, \rho_k \geq 0\}$ .

On trouvera dans la littérature diverses démonstrations de ces trois théorèmes classiques à propos des cônes convexes. Dans ce qui suit, on supposera toujours que  $\mathcal{C}$  satisfait les hypothèses des théorèmes 2 et 3.

Si  $\pi$ , d'équation  $l \cdot x = 0$ , contient une face de  $\mathcal{C}$ , le théorème 3 garantit que  $\pi$  est l'enveloppe linéaire des représentants des arêtes de  $\mathcal{C}$  qu'il contient. Il existe donc  $J$ , un ensemble d'indices, qui dépend de  $\pi$ , tel que les conditions suivantes (\*) définissent  $l$  à un facteur positif près :

(\*)  $l \cdot a_j = 0$  si  $j \in J$  et  $l \cdot a_j \geq 0$  sinon.

Inversement, s'il existe  $J$ , un ensemble d'indices tel que les conditions (\*) définissent  $l$  à un facteur positif près, l'hyperplan d'équation  $l \cdot x = 0$  est un hyperplan d'appui qui contient une face de  $\mathcal{C}$ .

En effet, en appliquant le théorème 3, on constate que pour tout  $x$  appartenant à  $\mathcal{C}$ ,  $l \cdot x \geq 0$ ; d'autre part, il est clair, sous ces conditions, que l'ensemble des  $x \in \mathcal{C}$  vérifiant  $l \cdot x = 0$  est de dimension maximale dans cet hyperplan.

On appellera facette d'un cône convexe saillant de dimension maximale toute face ou d-face de ce cône.

Soit  $F_i$ , une facette de  $\mathcal{C}$  :  $F_i = \bigcap F_j$  où  $i$  parcourt l'ensemble d'indices  $I$ . Appelons  $E_i$  le plus petit sous-espace de  $E$  contenant  $F_i$ . Dans  $E_i$ ,  $F_i$  est un cône de dimension maximale. Cela a donc un sens de parler des facettes de  $F_i$ .

Montrons que  $F_i \subset E_i$  satisfait les hypothèses du théorème 3.

Décomposons  $E$  en somme directe de  $E_i$  et de  $E_i^\perp$ . Tout vecteur  $l_j$  s'écrit de façon unique comme somme de  $l_j^i \in E_i$  et de  $l_j^{\perp} \in E_i^\perp$ .

Par construction,  $F_i = \{x \in E \mid l_i \cdot x = 0 \text{ si } i \in I \text{ et } l_j \cdot x \geq 0\}$ .

Comme  $F_i$  est de dimension maximale dans  $E_i$ , si  $i$  est dans  $I$ ,  $l_i \cdot x$  vaut 0 pour tout  $x$  dans  $E_i$ . Par conséquent,  $F_i = \{x \in E_i \mid l_j \cdot x \geq 0\} = \{x \in E_i \mid l_j^i \cdot x \geq 0\}$ .

Après élimination des  $l_k^i$  nuls, ou pouvant s'exprimer comme combinaison linéaire positive des autres  $l_j^i$  ( $j \neq k$ ), on trouve un ensemble d'indices  $J$  minimal, non-vide, disjoint de  $I$ , qui caractérise  $F_i$  :

$$F_i = \{x \in E_i \mid l_j^i \cdot x \geq 0, j \in J\} = \{x \in E_i \mid l_j \cdot x \geq 0, j \in J\}.$$

Pour montrer que  $F_i \subset E_i$  satisfait les hypothèses du théorème 3, il suffit de remarquer que, si un point  $x$  de  $E_i$  satisfait  $l_j \cdot x = 0, \forall j \in J$ , ce même point considéré comme point de  $E$  satisfait alors  $l_i \cdot x = 0 \forall i$  et, par conséquent, vaut 0.

La relation "être une facette de" est une relation transitive. Plus précisément :

**Lemme 1 :**

*Toute facette de  $F_i$  est une facette de  $\mathcal{C}$ .*

**Lemme 2 :**

*Toute facette de  $\mathcal{C}$  strictement contenue dans  $F_i$  est une facette de  $F_i$ .*

Les démonstrations sont pour ainsi dire triviales.

**Corollaire :** Toute facette de  $\mathcal{C}$  est entièrement caractérisée par la liste des arêtes de  $\mathcal{C}$  qu'elle contient. Les arêtes d'une telle facette sont exactement les arêtes de  $\mathcal{C}$  qu'elle contient; elle en est l'enveloppe convexe.

Toute arête de  $\mathcal{C}$  s'obtient en intersectant des faces de  $\mathcal{C}$ .

Soit  $a$ , un vecteur représentant une arête de  $C$ ; il existe  $I$ , un ensemble d'indices tel que :  $\{x \in E \mid x = \rho a, \rho \geq 0\} = \bigcap_{i \in I} F_i$  avec  $i \in I$ . Cela revient à dire que les conditions  $l_j \cdot x = 0$  si  $i \in I$  et  $l_j \cdot x \geq 0 \forall j$  définissent le vecteur  $a$  à un facteur positif près.

**Lemme 3 :**

*Un représentant d'une arête de  $C$  ne peut jamais s'exprimer comme combinaison linéaire positive des représentants des autres arêtes de  $C$ .*

**Démonstration :**

Soit  $a$ , un représentant d'une arête de  $C$ . En temps que point de  $C$ ,  $a$  peut s'exprimer comme combinaison linéaire positive des représentants de toutes les arêtes de  $C$  (théorème 3). En utilisant le fait que les conditions  $l_j \cdot x = 0$  si  $i \in I$  et  $l_j \cdot x \geq 0 \forall j$  définissent  $a$  à un facteur positif près, il est facile de voir que le seul coefficient non nul de cette combinaison linéaire est le coefficient devant  $a$ .

□ L. 3

**Lemme 4 :**

*Tout point intérieur de  $C$  (resp. d'une facette de  $C$ ) peut s'exprimer comme combinaison linéaire strictement positive des représentants des arêtes de  $C$  (resp. de cette facette).*

**Démonstration :**

Soit  $x$ , un point intérieur de  $C$  (resp. d'une facette de  $C$ ) et  $a_i, i \in I$ , les représentants des arêtes de  $C$  (resp. de cette facette de  $C$ ).

Si  $\lambda > 0$  est suffisamment petit,  $x' = x - \lambda \sum_{i \in I} a_i$  appartient à  $C$  (resp. à cette fa-

cette de  $C$ ) et peut s'exprimer comme combinaison linéaire positive des  $a_i$ . On en conclut que  $x = x' + \lambda \sum_{i \in I} a_i$  s'exprime comme combinaison linéaire strictement

positive des  $a_i, i \in I$ .

□ L. 4

**4.2. Relèvement d'un cône saillant :**

**4.2.1. Définitions et propriétés fondamentales :**

Considérons un cône convexe saillant  $C$ , de dimension  $d$ , dans un espace euclidien  $E$  de dimension  $s$  ( $s \geq d$ ). On supposera  $C$  donné par  $\mathcal{A} = \{a_1, a_2, \dots, a_s\}$  où  $a_i$  est un vecteur représentant la  $i^{\text{ème}}$  arête de  $C$ . Après renumérotation éventuelle, on peut supposer  $a_1, a_2, \dots, a_d$  linéairement indépendants.

Notons  $E^d$ , l'enveloppe linéaire des  $a_j$ .  $E^d$  est un sous-espace de dimension  $d$  de l'espace euclidien  $E$ . Construisons, maintenant, une suite d'espaces euclidiens emboîtés  $E^d \subset E^{d+1} \subset \dots \subset E^s = E$ , telle que  $E^k$  soit de dimension  $k$  et s'obtienne par adjonction à  $E^{k-1}$  d'un vecteur  $e_k \perp E^{k-1}$ .

Pour tout entier  $k$ ,  $d \leq k \leq s$ , on définit l'application  $r^k$  :

$$r^k : \mathbb{A} \longrightarrow E^k$$

$$a_j \longmapsto a_j \text{ si } 1 \leq j \leq s+d-k; \quad a_j + e_{s+d+1-j} \text{ si } j > s+d-k.$$

Pour simplifier les notations, posons  $a_j^k = r^k(a_j)$  et  $\mathbb{A}^k = r^k(\mathbb{A})$ . Ainsi :

$$\mathbb{A}^d = \mathbb{A}$$

$$\mathbb{A}^{d+1} = \{a_1, a_2, \dots, a_{s-1}, a_s + e_{d+1}\}$$

$$\mathbb{A}^{d+2} = \{a_1, a_2, \dots, a_{s-1} + e_{d+2}, a_s + e_{d+1}\}$$

etc.

$$\mathbb{A}^s = \{a_1, a_2, \dots, a_d, a_{d+1} + e_s, a_{d+2} + e_{s-1}, \dots, a_{s-1} + e_{d+2}, a_s + e_{d+1}\}$$

Dans  $E^k$ , définissons maintenant  $\mathbb{C}^k$ , l'enveloppe convexe des demi-droites issues de l'origine, caractérisées par les éléments de  $\mathbb{A}^k$ . Observons d'abord que les éléments de  $\mathbb{A}^k$  sont linéairement positivement indépendants puisque les éléments de  $\mathbb{A}$  le sont.

En s'inspirant de la démonstration de Voronoï qui, utilisant la notion de domaine corrélatifs, montre que les arêtes d'un domaine associé à une forme parfaite sont parmi les demi-droites dont on prend l'enveloppe convexe, nous allons vérifier que  $\mathbb{C}^k$  est un cône convexe saillant dont les arêtes sont exactement caractérisées par les éléments de  $\mathbb{A}^k$ .

**Démonstration :**

Soit  $R = \{x \in E^k \mid a_i^k \cdot x \geq 0, \forall i\}$ . Par le théorème 1, comme les  $a_i^k$  sont linéairement positivement indépendants,  $R$  est de dimension maximale. Montrons que  $R$  satisfait aussi les hypothèses du théorème 3, c'est-à-dire si  $x \in E^k$  satisfait  $a_i^k \cdot x = 0, \forall i$ ,  $x$  est nul.

Pour cela, décomposons  $E^k$  en somme directe de  $E^d$  et de son complémentaire orthogonal. On écrira  $x$  sous la forme  $x = x_d + x^\perp$ , avec  $x_d$  appartenant à  $E^d$  et  $x^\perp$  à son complémentaire orthogonal. Par construction, les  $a_j^k$  ( $1 \leq j \leq d$ ) appartiennent à  $E^d$  et satisfont  $a_j^k = a_j$ . Par conséquent, pour  $j$  compris entre 1 et  $d$ , l'égalité  $a_j^k \cdot x = 0$  implique  $a_j \cdot x_d = 0$ . Comme, par hypothèse, les  $a_j$  ( $1 \leq j \leq d$ ) forment une base de  $E^d$ , on en déduit  $x_d = 0$ , ou encore  $x = x^\perp$ .

Mais  $x^\perp$  peut s'exprimer comme combinaison linéaire des  $e_i$  ( $d+1 \leq i \leq k$ ).

Donc,  $x^\perp = \sum_{i=d+1}^k y_i e_i$  avec  $e_i \perp e_j$  si  $i \neq j$ .

Choisissons  $j$  tel que  $s+d-k < j \leq s$ . Par construction,  $a_j^k = a_j + e_{s+d+1-j}$ . Dans ce cas, l'égalité  $a_j^k \cdot x = 0$  implique  $y_{s+d+1-j} = 0$ . En observant que la condition  $s+d-k < j \leq s$  équivaut à  $k+1 > s+d+1-j \geq d+1$ , on conclut que  $y_i = 0$  ( $d+1 \leq i \leq k$ ) et, finalement, que  $x = 0$ .

En appliquant le théorème 3 à  $R$ , on déduit que  $R$  possède des arêtes. Si l'on représente ces arêtes par  $l_1, l_2, \dots, l_t$ , on a  $R = \{ x \in E^k \mid x = \sum_{i=1}^t \rho_i l_i, \rho_i \geq 0 \}$ .

Définissons  $R'$  le domaine corrélatif de  $R$ ;  $R' = \{ x \in E^k \mid l_i \cdot x \geq 0, \forall i \}$ .

Les représentants des arêtes d'un cône convexe étant toujours linéairement positivement indépendants, on en conclut que les  $l_i$  sont linéairement positivement indépendants et que, par conséquent,  $R'$  est de dimension maximale.

D'autre part, comme les  $l_i$  représentent les arêtes d'un cône convexe, aucun  $l_j$  ne peut s'exprimer comme combinaison linéaire des autres  $l_i$  ( $i \neq j$ ). Par le théorème 2, on en déduit que  $R'$  possède des faces et que les  $l_i$  sont les vecteurs perpendiculaires aux hyperplans d'appui de  $R'$  qui contiennent ces faces.

Comme  $R$  est un cône convexe de dimension maximale engendré par les combinaisons linéaires positives des  $l_i$ , on en déduit que les  $l_i$  forment un système de générateurs de  $E^k$ . Par conséquent, si  $x \in E^k$  satisfait  $l_i \cdot x = 0, \forall i$ ,  $x$  vaut 0.

Ainsi,  $R'$  satisfait les hypothèses du théorème 3. C'est un cône saillant de dimension maximale; il possède des arêtes.

Il suffit maintenant de montrer que les arêtes de  $R'$  sont exactement caractérisées par les éléments de  $\mathcal{A}^k$ , ce qui démontrera que  $R' = C^k$ .

Montrons d'abord que tout élément de  $\mathcal{A}^k$  caractérise une arête de  $R'$ .

Soit  $a_j^k \in \mathcal{A}^k$ ;  $a_j^k$  est perpendiculaire à un hyperplan d'appui de  $R$ . L'intersection de  $R$  avec cet hyperplan d'appui est une face  $F$ . Comme toute face de  $R$ ,  $F$  est l'enveloppe convexe des arêtes de  $R$  qu'elle contient. Supposons que les  $l_j, j \in J$ , représentent ces arêtes; l'enveloppe linéaire des  $l_j, j \in J$ , étant de codimension 1 dans  $E^k$ , les conditions  $a_j^k \cdot l_j = 0$  si  $j \in J$ ,  $a_j^k \cdot l_j \geq 0 \forall j$ , définissent  $a_j^k$  à un facteur positif près. Cela revient à dire que  $a_j^k$  caractérise une arête de  $R'$ .

Reste à voir que toute arête de  $R'$  appartient à  $\mathcal{A}^k$ .

Soit une arête de  $R'$  caractérisée par  $a$ . Les conditions  $l_i \cdot a = 0$  si  $i \in I$  et  $l_i \cdot a \geq 0 \forall i$ , caractérisent  $a$  à un facteur positif près. Cela revient à dire que  $a$  est perpendiculaire à un hyperplan d'appui de  $R$  et caractérise une face de  $R$ . Par le théorème 2,  $a$  est positivement proportionnel à un élément de  $\mathcal{A}^k$ .

#### 4.2.2. Théorème fondamental :

Pour tout entier  $k$ ,  $d < k \leq s$ , notons  $p^k$  la projection orthogonale de  $E^k$  sur  $E^d$  et  $\pi^k$  la projection orthogonale de  $E^k$  sur  $E^{k-1}$ . Par construction,  $\pi^k(a_i^k) = a_i^{k-1}$  et  $p^k(a_i^k) = a_i$ . On en déduit que  $p^k(C^k) = C$ ,  $\pi^k(C^k) = C^{k-1}$  et  $p^k \circ r^k = \text{identité}$ .

#### Théorème 4 :

*Toute face de  $C^{k-1}$  est projection par  $\pi^k$ , soit d'une face de  $C^k$ , soit de l'intersection de deux faces de  $C^k$ .*

#### Démonstration :

Soit  $F$ , une face de  $C^{k-1}$ .  $F$  est une variété de dimension  $k-2$ . Considérons  $\alpha'$ , l'hyperplan d'appui dans  $E^{k-1}$  qui contient cette face et  $\alpha = (\pi^k)^{-1}(\alpha')$ , l'hyperplan dans  $E^k$  engendré par  $\alpha'$  et  $e_k$ . Notons  $I$ , l'ensemble des indices des représentants des arêtes de  $C^{k-1}$  contenues dans  $F$ . Par le corollaire des lemmes 1 et 2, ces arêtes sont les arêtes de  $F$ .

Tous les points de  $C^k$  sont du même côté de  $\alpha$ , puisque tous les points de  $C^{k-1}$  sont du même côté de  $\alpha'$ .

Notons  $l_i$ , les vecteurs normaux aux hyperplans d'appui de  $C^k$ , et  $l$ , un vecteur normal à  $\alpha$ . On choisira le sens de  $l$  et celui des  $l_i$  de telle sorte que, pour tout  $x$  appartenant à  $C^k$ , on ait  $l_i \cdot x \geq 0$  et  $l \cdot x \geq 0$ .

La variété de  $E^k$  définie par  $l_i \cdot x \geq 0, \forall i$  et  $(-l) \cdot x \geq 0$  vit dans  $\alpha$ . Par le théorème 1, il existe  $\rho \geq 0$  et  $\rho_i \geq 0$ , non tous nuls, tels que :

$$\rho(-l) + \sum \rho_i l_i = 0.$$

Comme les  $l_i$  sont linéairement positivement indépendants,  $\rho$  doit être non nul, et il existe au moins un  $\rho_i$  non nul. Après renumérotation éventuelle, on peut supposer  $\rho_1 > 0$ .

$$\text{Donc, } l = \sum \frac{\rho_i}{\rho} l_i \text{ avec } \rho_1 > 0.$$

Comme les  $a_i^k, i \in I$ , sont contenus dans  $\alpha$ ,  $l \cdot a_i^k = 0$  si  $i \in I$ . Si  $\rho_1 > 0$ , on déduit de ce qui précède que  $l_1 \cdot a_i^k = 0$  si  $i \in I$ . Cela signifie que les  $a_i^k, i \in I$ , sont tous contenus dans la face  $F_1$  et représentent donc des arêtes de cette face. Deux cas sont possibles :

Premier cas : un seul  $\rho_i$  est non nul, disons  $\rho_1$ .

Dans ce cas,  $l$  est positivement proportionnel à  $l_1$ . Toute arête de  $F_1$  étant une arête de  $C^k$ , elle se projette par  $\pi^k$  sur une arête de  $C^{k-1}$  contenue dans  $\alpha'$ , donc une arête de  $F$ . Les arêtes de  $F$  étant caractérisées par les  $a_i^{k-1}, i \in I$ , on en

conclut que les  $a_i^k, i \in I$ , caractérisent toutes les arêtes de  $F_1$ . Par conséquent, par linéarité de  $\pi^k$ ,  $\pi^k(F_1) = F'$ .

Second cas : un second  $\rho_i$  est non nul, disons  $\rho_2$ .

Notons  $F_1$  et  $F_2$ , les faces correspondantes de  $\mathcal{C}^k$ , et posons  $D = F_1 \cap F_2$ ; nous allons montrer que  $\pi^k(D) = F'$ .

$D$  étant une facette de  $\mathcal{C}^k$ , il suffit de voir que les  $a_i^k, i \in I$ , caractérisent exactement les arêtes de  $D$ . Comme toute arête de  $\mathcal{C}^k$  contenue dans  $F_1$  et dans  $F_2$  est une arête de  $D$ , les  $a_i^k, i \in I$ , caractérisent des arêtes de  $D$ . Reste à voir qu'il n'y en a pas d'autres.

A priori,  $D$  est un cône convexe de dimension inférieure ou égale à  $k-2$ . Mais comme  $\alpha'$  est l'enveloppe linéaire des  $a_i^{k-1}, i \in I$ , la dimension de l'enveloppe linéaire des  $a_i^k, i \in I$ , doit être supérieure ou égale à  $k-2$ . Par conséquent, dans ce cas, elle vaut exactement  $k-2$ , et  $D$  est un cône convexe, de dimension  $k-2$ , contenu dans l'enveloppe linéaire des  $a_i^k, i \in I$ .

On en déduit que la projection par  $\pi^k$  de toute arête de  $D$ , non seulement est une arête de  $\mathcal{C}^{k-1}$ , mais est contenue dans  $\alpha'$ ; c'est une arête de  $F'$ . Comme précédemment, les arêtes de  $F'$  étant caractérisées par les  $a_i^{k-1}, i \in I$ , on en conclut que les  $a_i^k, i \in I$ , caractérisent toutes les arêtes de  $D$ .

□ Th. 4

### 4.3. Faces voisines :

#### 4.3.1. Définition et propriété fondamentale :

On dira que deux faces  $F_1$  et  $F_2$  de  $\mathcal{C}$  sont voisines si leur intersection est une facette de codimension 2.

#### Lemme 5 :

*Si  $D$  est une facette de  $\mathcal{C}$  de codimension 2, seules exactement deux faces de  $\mathcal{C}$  contiennent  $D$ .*

#### Démonstration :

Par construction,  $D$  est contenue au moins dans deux faces de  $\mathcal{C}$ .

Considérons  $U$ , l'enveloppe linéaire des éléments de  $D$ , et  $U^\perp$ , son complémentaire orthogonal. Par hypothèse,  $U^\perp$  est de dimension 2, c'est un 2-plan.

Soit  $F_i$ , une face quelconque de  $\mathcal{C}$  contenant  $D$ ,  $\tau_i$ , l'hyperplan d'appui correspondant à  $F_i$ , et  $l_i$ , un vecteur perpendiculaire à  $\tau_i$ . Le sens de  $l_i$  est fixé par la condition  $l_i \cdot x \geq 0, \forall x \in \mathcal{C}$ .

Observons que  $l_i \cdot x = 0, \forall x \in D$ . Par conséquent,  $l_i \cdot x = 0, \forall x \in U$  et  $l_i \in U^\perp$ .

Par le théorème 1, on sait que les  $l_i$  sont linéairement positivement indépendants. Par le théorème 2, on peut supposer qu'aucun  $l_i$  ne puisse s'exprimer comme combinaison linéaire positive des autres  $l_j$  ( $j \neq i$ ).

Dans un 2-plan, seul un ensemble de deux vecteurs, au plus, peut satisfaire simultanément ces deux conditions.

□ L. 5

### Corollaires :

Soit  $D$ , une facette de  $\mathcal{C}$  de codimension 2. Notons  $F$  et  $F'$  les deux seules faces de  $\mathcal{C}$  qui contiennent  $D$ ; par définition d'une facette de  $\mathcal{C}$ , on sait que  $D$  est exactement l'intersection de  $F$  et de  $F'$ .

Aucune autre face de  $\mathcal{C}$  ne peut contenir un point intérieur de  $D$ . En effet, si  $x$  est un point intérieur de  $D$ , il peut s'exprimer comme combinaison linéaire strictement positive des représentants des arêtes de  $D$ . Ainsi, toute face de  $\mathcal{C}$  contenant  $x$  contient toutes les arêtes de  $D$  et, par conséquent,  $D$  tout entier.

### 4.3.2. Connexion :

On dira que deux faces  $F$  et  $F'$  de  $\mathcal{C}$  sont connectées s'il existe une liste de faces de  $\mathcal{C}$  :  $F_1 = F, F_2, \dots, F_l = F'$  avec la propriété que  $F_j$  et  $F_{j+1}$  sont voisines  $\forall j$ .

Remarque : "être connecté" est une relation d'équivalence.

### 4.3.2.1. Théorème fondamental :

#### Théorème 5 :

*Deux faces de  $\mathcal{C}$  sont toujours connectées pour peu que  $\mathcal{C}$ , en tant que variété, soit de dimension  $\geq 2$ .*

#### Démonstration :

Posons  $1 = \sum_{i=1}^s l_i$  et considérons  $\pi$ , l'hyperplan affine d'équation  $1 \cdot x = \sum_{j=1}^s a_j$ .

Le fait que  $1 \cdot a_j > 0, \forall j$ , montre que toute arête de  $\mathcal{C}$  coupe  $\pi$  en un point.

L'intersection de  $\pi$  et de  $\mathcal{C}$  est donc un polyèdre  $\mathcal{P}$  convexe, compact, d'intérieur non-vide, c'est-à-dire un polytope. Les faces de  $\mathcal{P}$  sont en bijection avec celles de  $\mathcal{C}$ . De plus, cette bijection est compatible avec la propriété "être voisines".

On conclut en utilisant le théorème bien connu qui affirme que les faces de  $\mathcal{P}$  sont connectées [Be 1].

□ Th. 5

#### 4.3.2.2. Faces maximales :

Dans ce qui suit, on supposera que  $C$  est un cône convexe saillant possédant  $s$  arêtes.

On dira qu'une facette de  $C$  est maximale si seule une arête de  $C$  n'est pas contenue dans cette facette. Il est clair que toute facette maximale de  $C$  est une face de  $C$ . On parlera donc de face maximale.

On appellera taille d'un cône convexe  $C$  (resp. d'une facette de  $C$ ) la différence entre le nombre d'arêtes contenues dans  $C$  (resp. dans cette facette de  $C$ ) et la dimension de  $C$  (resp. de cette facette de  $C$ ).

Si  $C$  possède  $s$  arêtes,  $C$  possède au plus  $s$  faces maximales. Si  $s = \dim C$ ,  $C$  possède exactement  $s$  faces qui sont toutes maximales. La réciproque est vraie; si  $C$  possède  $s$  faces maximales,  $C$  ne possède pas d'autres faces et  $s = \dim C$  (théorème 7).

#### Théorème 6 :

*Soit  $F$  une face maximale de  $C$ .*

- (i) Elle est voisine de toutes les autres faces de  $C$ .*
- (ii) Il y a bijection naturelle entre les faces de  $C$  différentes de  $F$  et les faces de  $F$ .*
- (iii) Cette bijection est compatible avec la relation "être voisine".*
- (iv) Cette bijection conserve la taille.*
- (v) Cette bijection conserve la propriété "être maximale".*

#### Démonstration :

Soit  $F'$ , une face de  $C$  différente de  $F$ . Considérons  $f = F \cap F'$ .  $f$  est une facette de  $C$  contenue dans  $F$ . Par le lemme 2, c'est une facette de  $F$ . La seule arête de  $F$  hors de  $f$  étant celle de  $C$  qui n'est pas dans  $F$ ,  $f$  est une face maximale de  $F$  et  $\dim f = (\dim F) - 1$ .

(i) L'égalité  $\dim f = (\dim F) - 1$  implique que  $F$  et  $F'$  sont voisines.

(ii)  $f$  est une facette de  $C$  strictement contenue dans  $F$ . C'est une face de  $F$ . Le lemme 5 montre que l'application qui envoie  $F'$  sur  $f = F \cap F'$  crée une bijection entre l'ensemble des faces de  $C$  différentes de  $F$  et l'ensemble des faces de  $F$ .

(iii) Soit  $F_1$  et  $F_2$ , deux faces de  $\mathcal{C}$ , différentes de  $F$ ; posons  $f_1 = F \cap F_1$  et  $f_2 = F \cap F_2$ .  $f_1 \cap f_2$  est une facette de  $\mathcal{C}$  contenue dans  $F_1 \cap F_2$ ; c'est une facette de  $F_1 \cap F_2$ . Vu qu'elle contient toutes les arêtes de  $F_1 \cap F_2$ , excepté celle de  $\mathcal{C}$  qui n'est pas dans  $F$ ,  $f_1 \cap f_2$  est une face maximale de  $F_1 \cap F_2$ . En particulier,  $\dim(f_1 \cap f_2) = \dim(F_1 \cap F_2) - 1$ . On déduit facilement de cette dernière égalité que  $f_1$  et  $f_2$  sont voisines si et seulement si  $F_1$  et  $F_2$  le sont.

(iv) D'une part,  $\dim f = (\dim F) - 1$ ; d'autre part,  $F$  possède exactement une arête de plus que  $f$ . Donc,  $\text{taille } f = \text{taille } F$ .

(v) Comme  $f$  est toujours une face maximale de  $F$  et  $F$  une face maximale de  $\mathcal{C}$ ,  $f$  est une face maximale de  $F$  si et seulement si  $F$  est une face maximale de  $\mathcal{C}$ .

Conséquence :  $F$  possède exactement une face maximale de moins que  $\mathcal{C}$ .

□ Th. 6

On dira que deux faces  $F$  et  $F'$  de  $\mathcal{C}$  sont fortement connectées s'il existe une liste de faces non maximales de  $\mathcal{C}$  :  $F_1 = F, F_2, \dots, F_l = F'$  avec la propriété que  $F_j$  et  $F_{j+1}$  sont voisines  $\forall j$ .

### Théorème 7 :

*Soit  $\mathcal{C}$ , un cône convexe saillant de dimension supérieure ou égale à 3, possédant  $s$  arêtes. Notons  $\psi(\mathcal{C})$ , le nombre de ses faces maximales.*

(i) Si  $\psi(\mathcal{C}) > (\dim \mathcal{C}) - 3$ , alors  $\psi(\mathcal{C}) = \dim \mathcal{C} = s$ .

(ii) Si  $\psi(\mathcal{C}) \leq (\dim \mathcal{C}) - 3$ , alors  $\mathcal{C}$  possède des faces non maximales et les faces non maximales de  $\mathcal{C}$  sont fortement connectées.

**Démonstration :** (par induction sur la dimension de  $\mathcal{C}$ )

**Ancrage :**  $\dim \mathcal{C} = 3$

(i) Si  $\psi(\mathcal{C}) > (\dim \mathcal{C}) - 3 = 0$ ,  $\mathcal{C}$  possède au moins une face maximale. Mais toutes les faces de  $\mathcal{C}$  possèdent exactement deux arêtes. Donc  $s = 3$ , et les trois faces de  $\mathcal{C}$  sont maximales.

(ii) Si  $\psi(\mathcal{C}) \leq (\dim \mathcal{C}) - 3 = 0$ , alors  $\psi(\mathcal{C}) = 0$ . Aucune face n'est maximale et on conclut en se rappelant que les faces de  $\mathcal{C}$  sont toujours connectées.

**Pas d'induction :**  $\dim \mathcal{C} > 3$

Si  $\psi(\mathcal{C}) = 0$ , on conclut comme pour l'ancrage (ii).

On supposera donc que  $\mathcal{C}$  possède au moins une face  $F$  maximale. On a :

$\psi(F) = \psi(C) - 1$  et  $\dim F = (\dim C) - 1 \geq 3$ . Deux cas sont à envisager :

Cas 1 :  $\psi(C) > (\dim C) - 3$ ; dans ce cas,  $\psi(C) - 1 > (\dim C) - 3 - 1 \geq 0$ , ce qui signifie  $\psi(F) > (\dim F) - 3$ .

En appliquant l'hypothèse d'induction à  $F$ , on obtient  $\psi(F) = \dim F = (\text{nombre d'arêtes dans } F) = s - 1$ , puisque  $F$  est maximale.

On en déduit  $\psi(C) = \psi(F) + 1 = (\dim F) + 1 = \dim C$ . En remplaçant  $\dim F$  par sa valeur  $s - 1$ , on obtient  $\psi(C) = \dim C = s$ .

Cas 2 :  $\psi(C) \leq (\dim C) - 3$ ; dans ce cas,  $\psi(C) - 1 \leq (\dim C) - 3 - 1$ , ce qui signifie  $\psi(F) \leq (\dim F) - 3$ .

En appliquant l'hypothèse d'induction à  $F$ , on obtient que  $F$  possède des faces non maximales et que ces faces non maximales sont fortement connectées. Les points (ii) et (v) du théorème 6 permettent alors de conclure que  $C$  possède des faces non maximales. Le point (iii) du théorème 6 montre qu'elles sont fortement connectées.

□ Th. 7

### Corollaires :

- 1) Si  $s > \dim C$ , alors  $\psi(C) \leq (\dim C) - 3$  et  $C$  possède des faces non maximales. Il suffit de remarquer que l'hypothèse  $s > \dim C$  implique  $\dim C \geq 3$ .
- 2) Si  $s > \dim C$ , on peut appliquer l'algorithme suivant pour énumérer les faces de  $C$  :
  - a. Chercher une face non maximale de  $C$ .
  - b. Calculer toutes ses faces voisines.
  - c. Mémoriser toutes les nouvelles faces.
  - d. Pour chaque nouvelle face non maximale, recommencer à partir du point b.

#### 4.4. Graphe des faces d'un cône saillant :

Si  $C$  est un cône convexe saillant de dimension supérieure ou égale à deux, satisfaisant les hypothèses des théorèmes 2 et 3, la notion de faces voisines permet d'interpréter l'ensemble des faces de  $C$  comme un graphe localement fini; les faces jouent le rôle des sommets du graphe, les faces des faces, celui des arêtes.

Le lemme 5 montre que ce graphe est bien défini : chaque arête relie exactement deux sommets! Le théorème 5 montre qu'il est connexe.

Le théorème 6 nous dit qu'à une face maximale de  $C$  correspond un sommet du graphe relié, par des arêtes, à tous les autres sommets. Lorsque le nombre d'arêtes de  $C$  est égal à la dimension de  $C$ , le graphe des faces de  $C$  est complet.

## 5. Domaines de Voronoï :

### 5.1. Domaine associé à une forme parfaite :

On montre aisément qu'une forme est parfaite si et seulement si son domaine est de dimension maximale.

Le domaine d'une forme parfaite a été défini comme l'enveloppe convexe d'un ensemble bien déterminé de demi-droites. Nous allons voir que cet ensemble de demi-droites possède une propriété géométrique de minimalité, à savoir :

#### Lemme 6 :

*Si l'on retire une demi-droite de cet ensemble, l'enveloppe convexe de celles qui restent est toujours modifiée.*

#### Démonstration :

Pour cela, remarquons d'abord que tout  $a = v v^t$ ,  $v \in \mathbb{R}^n$ , fait un angle constant avec la matrice identité. On considère, bien entendu, la métrique définie par la trace. Cet angle satisfait  $\cos \theta = \frac{1}{\sqrt{n}}$ , en particulier,  $\theta \in ] 0, \frac{\pi}{2} [$ .

Considérons l'hyperplan d'équation  $\text{trace } X = n$ . Les intersections de cet hyperplan et des demi-droites dont on prend l'enveloppe convexe sont des points distincts, tous situés sur une sphère centrée en la matrice identité.

Ceci termine la démonstration.

□ L. 6

On déduit facilement de ce qui précède le théorème suivant :

#### Théorème 8 :

*Le domaine de Voronoï associé à une forme parfaite est un cône convexe saillant.*

*Les demi-droites dont on prend l'enveloppe convexe pour définir le domaine sont exactement les arêtes de ce domaine.*

#### Remarque :

Voronoï, dans son article, montre que les arêtes du domaine sont parmi les demi-droites dont on prend l'enveloppe convexe [Vo I].

Contrairement aux apparences, il ne démontre pas ces demi-droites sont exactement les arêtes du domaine. Il lui manque, en fait, le lemme 6 pour conclure.

## 5.2. Domaines voisins et formes voisines :

### 5.2.1. Définitions et propriétés fondamentales :

Soit  $q$  et  $q'$ , deux formes parfaites, et leurs domaines  $C$  et  $C'$  dans l'espace de Voronoï. Si ces deux formes sont positivement proportionnelles, il est clair que  $C = C'$ . Sinon, Voronoï montre que l'intersection de ces deux domaines ne contient aucun point intérieur de  $C$  ou de  $C'$ ; mieux, si  $C \cap C'$  contient un point intérieur d'une facette de  $C$ , cette facette est aussi une facette de  $C'$ . En particulier, si  $C \cap C'$  contient un point intérieur d'une face de  $C$ , cette face est commune à  $C$  et  $C'$ . Dans ce cas, ces deux domaines doivent donc obligatoirement se situer de part et d'autre de l'hyperplan d'appui définissant la face commune. On dira que  $C$  et  $C'$  sont des domaines voisins et que les formes  $q$  et  $q'$  sont des formes voisines.

A chaque face de  $C$  correspond donc, au plus, un domaine voisin.

### 5.2.2. Existence :

Montrons qu'à chaque face de  $C$  correspond toujours un domaine voisin.

Soit  $Q$ , la matrice associée à  $q$ ,  $F$ , une face de  $C$ , et  $\alpha$ , l'hyperplan d'appui d'équation  $l \cdot x = 0$  associé à  $F$ . Le sens de  $l$  est fixé par la condition  $l \cdot x \geq 0, \forall x \in C$ .

Montrons d'abord qu'il existe  $u \in \mathbb{Z}^n$ , tel que  $C$  et  $u^t$  ne soient pas du même côté de  $\alpha$ , ce qui revient à dire que la forme quadratique associée à  $l$  est indéfinie.

Pour cela, considérons  $L$  l'application linéaire associée à la matrice  $l$ . Observons qu'il existe une arête de  $F$ , caractérisée par  $a = v^t$  avec  $v \in \mathbb{Z}^n$ , telle que  $L(v)$  soit non nul. En effet, comme  $L$  est non nulle, son noyau est de dimension inférieure à  $n$ ; par conséquent, l'enveloppe linéaire, dans l'espace de Voronoï, des  $x \cdot x^t$  avec  $x \in \text{Ker}(L)$ , est de dimension inférieure ou égale à  $\frac{n(n-1)}{2}$ . On conclut en remarquant que si  $n > 1$ ,  $\frac{n(n-1)}{2} < \dim F = \frac{n(n+1)}{2} - 1$ . Soit, donc, un vecteur  $v \in \mathbb{Z}^n$ , tel que :

- (a)  $L(v) = l \cdot v$  soit non nul et
- (b)  $v^t l \cdot v = 0$ .

On choisira ensuite  $y$  et  $z$  dans  $\mathbb{Z}^n$  satisfaisant :

- (i)  $y^t l \cdot y > 0$  (penser à une arête de  $C$  qui n'est pas dans  $F$ ),
- (ii)  $z^t l \cdot v < 0$  (par exemple  $z = -L(v)$ ).

On peut supposer que  $y^t l \cdot v \leq 0$  (en remplaçant éventuellement  $y$  par  $-y$ ) et que  $z^t l \cdot z \geq 0$  (sinon,  $u = z$  est solution de notre problème).

Considérons finalement  $w = \lambda y + z$ . En remarquant que  $w^t l \cdot v$  est négatif pour tout  $\lambda$  positif et que  $w^t l \cdot w$  est positif dès que  $\lambda$  est suffisamment grand, on construit notre vecteur  $u$  en posant  $u = (w^t l \cdot w) v - (w^t l \cdot v) w$  pour une certaine

valeur entière de  $\lambda$  suffisamment grande. On vérifiera sans peine que si  $\lambda$  est assez grand,  $u^t l u = - (w^t l w) (w^t l v)^2$  est négatif.

Si  $Q'$  est voisine de  $Q$  relativement à la face  $F$ , on peut supposer sans restreindre la généralité que  $\min Q = \min Q'$ . Du fait que l'enveloppe linéaire des représentants des arêtes de  $F$  est de codimension 1, on déduit que  $Q' - Q$  est proportionnelle à  $l$ . Étudions par conséquent  $Q(\theta) = Q + \theta l$ , où  $\theta$  est un nombre réel. Si  $\theta < 0$ ,  $\min Q(\theta) < \min Q$  et les vecteurs minimaux de  $Q$  associés aux arêtes de  $F$  ne sont plus vecteurs minimaux de  $Q(\theta)$ . Seul, le cas  $\theta \geq 0$  reste donc intéressant.

Lorsque  $\theta > 0$  est suffisamment petit,  $\min Q(\theta) = \min Q$  et les vecteurs minimaux de  $Q(\theta)$  sont exactement ceux correspondant aux arêtes de  $C$  contenues dans  $F$ . Considérons l'ensemble  $\mathfrak{R}$  des valeurs de  $\theta \geq 0$  pour lesquelles  $\min Q(\theta) = \min Q$ .

L'existence d'un vecteur  $u \in \mathbb{Z}^n$  satisfaisant  $1 \cdot (u u^t) < 0$  montre que, si  $\theta$  est assez grand,  $u^t Q(\theta) u < \min Q$ .  $\mathfrak{R}$  est donc borné.

En observant que si  $\min Q(\theta) = \min Q$ , l'intervalle  $[0, \theta]$  est entièrement contenu dans  $\mathfrak{R}$ , on conclut que  $\mathfrak{R}$  est lui-même un intervalle.

La continuité de  $\min Q(\theta)$  montre qu'il est fermé.

$\mathfrak{R}$  est donc de la forme  $[0, \rho]$  pour un certain  $\rho > 0$ .

Il est facile de voir que l'ensemble des valeurs de  $\theta$  dans  $\mathfrak{R}$ , pour lesquelles les vecteurs minimaux de  $Q(\theta)$  sont exactement ceux correspondant aux arêtes de  $F$ , est un ouvert; c'est même l'intérieur de  $\mathfrak{R}$ . Par conséquent, la forme  $Q' = Q + \rho l$  possède au moins un vecteur minimal supplémentaire. L'arête correspondante, n'appartenant pas à  $F$ , doit être extérieure à  $\alpha$ . Le domaine associé à  $Q'$  est alors de dimension maximale dans l'espace de Voronoï. La forme  $Q'$  est parfaite; elle est la voisine de  $Q$  relativement à la face  $F$ .

A ce stade, il est important de remarquer que, bien qu'une face d'un domaine  $C$  n'appartienne qu'à un seul autre domaine  $C'$ , l'hyperplan  $\alpha$  qui contient  $F$  peut contenir des faces d'autres domaines. Ainsi, le même hyperplan  $\alpha$  peut être hyperplan d'appui de plusieurs domaines. La matrice  $l$  normale à  $\alpha$  ne caractérise donc pas, en général, une unique face.

### 5.2.3. Illustration du cas des formes à 2 variables :

Reprenons les notions de formes parfaites et de domaines associés aux formes parfaites dans le cas des formes à 2 variables, c'est-à-dire  $n = 2$ . L'espace des domaines de Voronoï est difficilement visualisable dans le cas général. Seul le cas  $n = 2$ ,  $N = 3$  s'y prête bien. C'est pourquoi il est intéressant d'approfondir cette si-

tuation particulière dans l'optique de Voronoï, afin de mieux saisir la géométrie de l'espace de Voronoï.

Prenons, pour commencer, la matrice identité. Elle est associée à la forme  $x^2 + y^2$ .

Son minimum vaut 1. Ses vecteurs minimaux sont  $\pm \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \pm \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ .

Cette forme n'est pas parfaite. En effet, il existe une infinité de formes quadratiques définies positives qui possèdent ces vecteurs comme vecteurs minimaux.

Une forme quadratique à deux variables est donnée par :

$$(x \ y) \begin{pmatrix} a & b \\ b & c \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}.$$

Si  $\pm \begin{pmatrix} 1 \\ 0 \end{pmatrix}$  est un vecteur minimal, alors  $a = 1$ .

Si  $\pm \begin{pmatrix} 0 \\ 1 \end{pmatrix}$  est un vecteur minimal, alors  $c = 1$ .

Mais  $b$  n'est pas déterminé. On montre facilement que pour tout  $b$  inférieur en valeur absolue à  $\frac{1}{2}$ , la forme correspondante possède exactement ces vecteurs comme vecteurs minimaux. Par contre, lorsque  $b$  vaut  $\frac{1}{2}$ , on obtient une forme qui possède un vecteur minimal supplémentaire. Cette forme est proportionnelle à  $A_2 = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$ . Elle est parfaite. En effet, le vecteur minimal supplémentaire  $\pm \begin{pmatrix} 1 \\ -1 \end{pmatrix}$  introduit la condition  $1 - 2b + 1 = 1$ , condition qui détermine  $b$ .

Le réseau correspondant à cette matrice est engendré par deux vecteurs de même longueur et formant un angle de  $60^\circ$ .

C'est le réseau, bien connu, d'empilement de sphères le plus dense dans le plan.

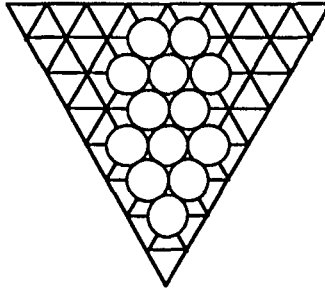


fig. 2

Dans  $\mathbb{R}^3$ , le domaine associé à  $\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$  possède exactement trois arêtes caractérisées par :

$$a_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad (1 \ 0) = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \quad a_2 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, \quad a_3 = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}.$$

Comme  $s = N = 3$ , on calcule les faces du domaine en enlevant à tour de rôle chacune des arêtes; celles qui restent déterminent une face. Par exemple, on obtient  $F_1$  en retirant  $a_1$  comme suit :

Posons  $F_1 = \begin{pmatrix} f_1 & f_2 \\ f_2 & f_3 \end{pmatrix}$ , un vecteur normal à la face cherchée. Donc  $F_1$  est normal à  $a_2$  et à  $a_3$ . Ces conditions s'expriment à l'aide du produit scalaire donné par la trace.

(i)  $F_1 \cdot a_2 = \text{Trace}(F_1 a_2) = 0$ ,

(ii)  $F_1 \cdot a_3 = \text{Trace}(F_1 a_3) = 0$ .

La condition (i) nous donne  $f_3 = 0$ ; la condition (ii) implique  $f_1 - 2f_2 + f_3 = 0$ . Ces équations définissent  $F_1$  à un facteur près. On peut choisir  $F_1$  de telle manière que ses composantes soient entières et premières entre elles, et fixer le sens de  $F_1$  en demandant que  $\text{Trace}(F_1 a_1) > 0$ . On obtient  $F_1 = \begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix}$ .

De manière analogue, on détermine  $F_2 = \begin{pmatrix} 0 & 1 \\ 1 & 2 \end{pmatrix}$  et  $F_3 = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}$ .

Etudions la voisine de  $\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$  relativement à la face  $\begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}$ .

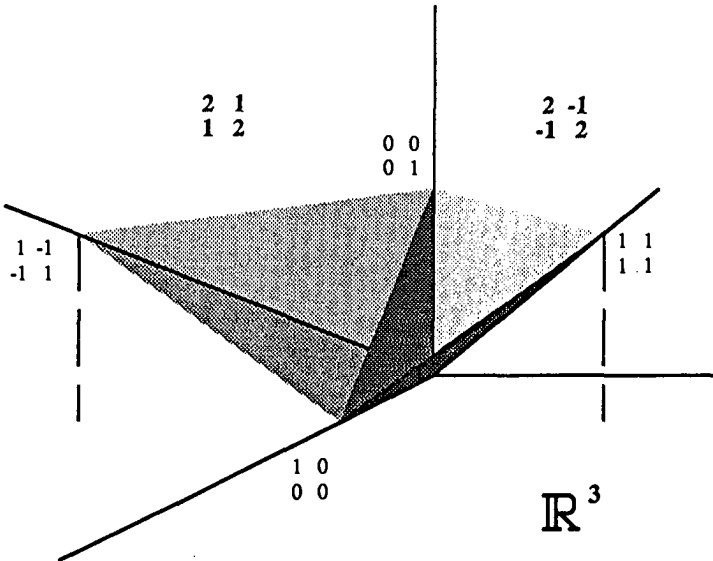


fig. 3

Cette voisine doit s'écrire  $\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} + \rho \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}$ , pour un certain  $\rho > 0$ . La voisine étant unique, il n'existe qu'une seule valeur de  $\rho$  positive qui donne une forme parfaite de même minimum.

On devine que pour  $\rho = 2$ , la forme correspondante  $\begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}$  est parfaite et de minimum 2. C'est donc la voisine cherchée.

La figure ci-dessus (fig. 3) visualise, dans  $\mathbb{R}^3$ , les domaines associés à ces deux formes voisines. Les domaines correspondants sont voisins. Leur face commune admet  $\begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}$  comme vecteur normal. Pour plus de clarté, j'ai indiqué en petits caractères les représentants des arêtes des domaines et en caractères gras les formes correspondant à chacun de ces domaines. On prolongera par l'esprit les faces dont je n'ai dessiné qu'une portion.

En contemplant  $\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} + \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ , on conclut que ces deux formes sont équivalentes.

On peut, de même, calculer les voisines correspondant aux deux autres faces. La voisine correspondant à  $F_1$  est donnée par  $\begin{pmatrix} 6 & 3 \\ 3 & 2 \end{pmatrix}$ , celle correspondant à  $F_2$  par  $\begin{pmatrix} 2 & 3 \\ 3 & 6 \end{pmatrix}$ . Après avoir calculé les voisines de  $A_2$ , on peut déterminer les voisines des voisines de  $A_2$ , etc. La figure 4, ci-dessous, prolonge la figure 3. Les arêtes de tout domaine sont caractérisées par des matrices de la forme  $\begin{pmatrix} a \\ b \end{pmatrix} (a \ b) = \begin{pmatrix} a^2 & ab \\ ab & b^2 \end{pmatrix}$ . Une telle matrice sera représentée dans  $\mathbb{R}^3$  par  $(a^2, \sqrt{2}ab, b^2)$ .

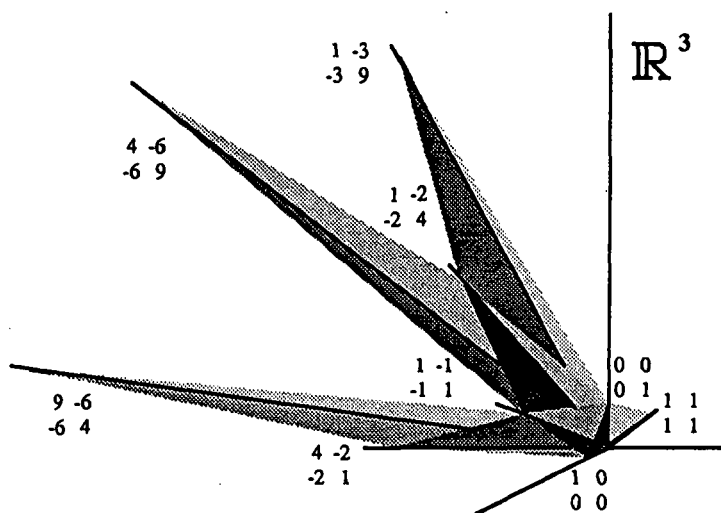


fig. 4

Ce sont ses coordonnées dans la base orthonormée  $\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$ .

Il est facile de voir que les voisines de  $A_2$  sont équivalentes à  $A_2$ , soit en explicitant des matrices de changements de bases qui réalisent l'identification, soit en constatant que le groupe des automorphismes de  $A_2$  est transitif sur les faces du domaine associé à  $A_2$ . Voronoï démontre que, de manière générale, le groupe des automorphismes de  $A_n$  agit transitivement sur les faces du domaine associé à cette forme.  $A_n$  est donnée par  $x_1^2 + x_2^2 + \dots + x_n^2 + (x_1 + x_2 + \dots + x_n)^2$ . Dans les cas  $n = 2$  et  $n = 3$ , les voisines de  $A_n$  sont équivalentes à  $A_n$ , donc, d'après Voronoï, il n'existe qu'une seule classe de formes parfaites dans ces dimensions.

Revenons au cas  $n = 2$ ; en remarquant que  $a^2 b^2 = (a b)^2$ , on conclut que les représentants de toutes arêtes appartiennent à la surface d'équation  $y^2 = 2 x z$ . Il s'agit d'un cône de révolution dont le sommet est à l'origine. La section de ce cône par le plan d'équation  $x + z = 2$  est un cercle de rayon  $\sqrt{2}$  centré en  $(1, 0, 1)$ . Remarquons, au passage, que  $(1, 0, 1)$  correspond à la matrice identité.

La figure 5, ci-dessous, visualise cette section.

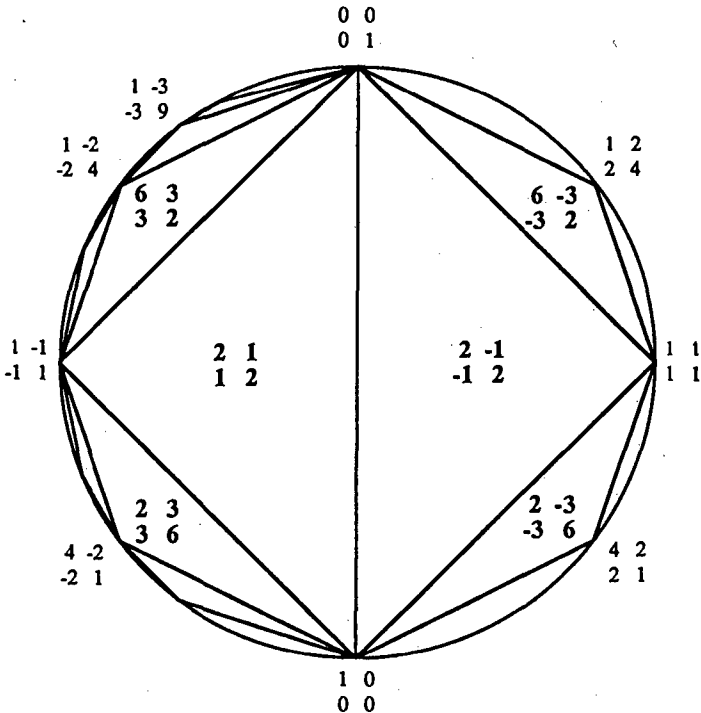


fig. 5

Ici, le chemin reliant deux domaines est unique. En effet, associons à chaque domaine un point intérieur et relierons les points qui appartiennent à deux domaines voisins. Le graphe ainsi obtenu est, en fait, un arbre.

Le sens d'un vecteur normal à une face d'un domaine est fixé par la condition que son produit scalaire avec un représentant d'une arête de ce domaine soit toujours positif ou nul. Pour qu'une forme B soit intérieure à un domaine, il faut et il suffit que  $F \cdot B > 0$  pour tout F, vecteur normal à une face du domaine. On voit facilement que  $A_2$  n'est pas intérieure à son domaine. En effet,  $\text{Tracc} \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} < 0$ . Par contre, en additionnant les représentants des arêtes d'un domaine, on obtient toujours une forme qui est intérieure à ce domaine. Ici, dans chaque cas, on trouve la forme adjointe. Par exemple, la forme adjointe de  $\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$  est  $\begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}$  :

$$\begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}.$$

La forme parfaite  $A_2$  est donc eutactique.

La figure 6, ci-dessous, illustre l'eutaxie de la forme parfaite  $A_2$ .

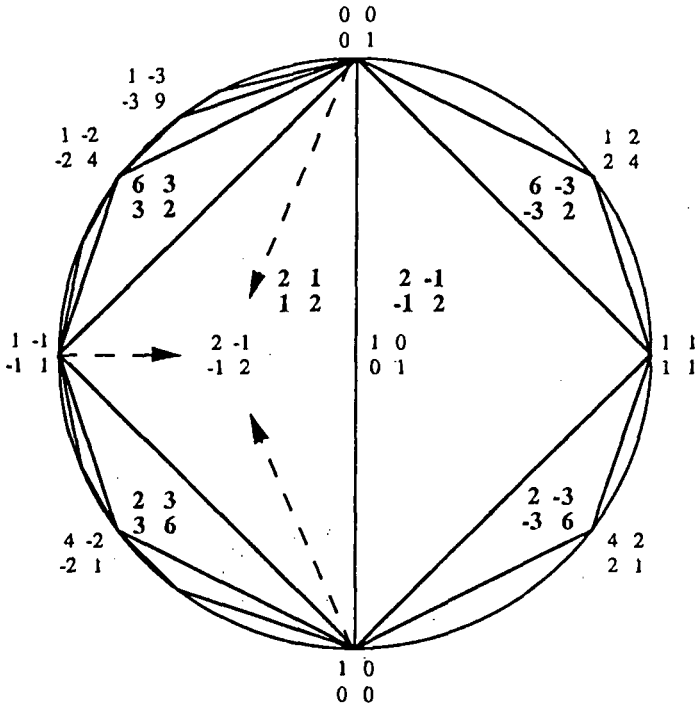


fig. 6

Je rappellerai ici le critère démontré par Voronoï : *une forme est extrême si et seulement si elle est parfaite et euactique.*

Lorsqu'on a une forme parfaite, on peut représenter son domaine dans l'espace de Voronoï; mais on peut aussi la considérer comme un point dans cet espace. On vient de voir qu'en général ce point n'appartient pas au domaine de la forme. Quel est alors le lien géométrique entre deux domaines voisins et les deux formes parfaites voisines correspondantes?

On peut répondre à cette question par un exemple. Partons du domaine associé à  $A_2$  et considérons la face de ce domaine, donnée par  $\begin{pmatrix} 0 & 1 \\ 1 & 2 \end{pmatrix}$ . Cette face appartient également à un autre domaine : le domaine associé à  $\begin{pmatrix} 2 & 3 \\ 3 & 6 \end{pmatrix}$ . Ces deux domaines sont donc voisins et les formes parfaites correspondantes sont voisines. Voronoï nous dit que  $\begin{pmatrix} 2 & 3 \\ 3 & 6 \end{pmatrix}$  doit s'obtenir en additionnant à  $\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$  un multiple positif du vecteur normal à la face, c'est-à-dire un multiple positif de  $\begin{pmatrix} 0 & 1 \\ 1 & 2 \end{pmatrix}$ . Numériquement, on obtient :

$$\begin{pmatrix} 2 & 3 \\ 3 & 6 \end{pmatrix} = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} + 2 \begin{pmatrix} 0 & 1 \\ 1 & 2 \end{pmatrix}.$$

La figure suivante illustre cette relation.

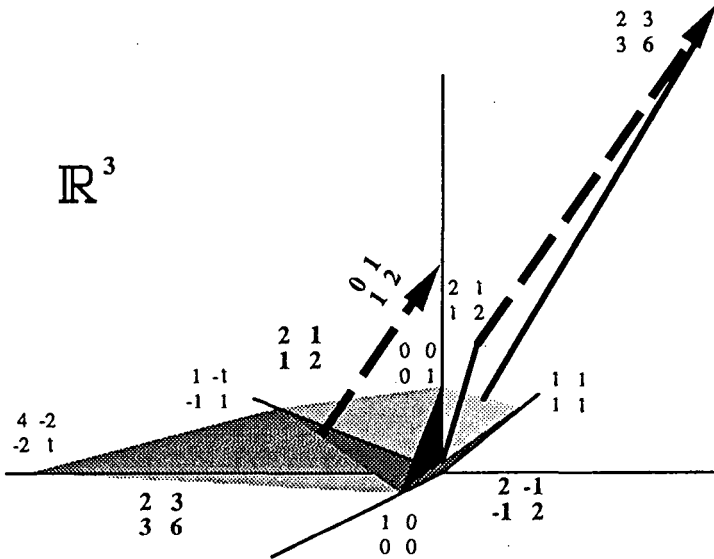


fig. 7

## 6. Graphe de Voronoï :

### 6.1. Définition :

La notion de domaines voisins permet d'interpréter l'ensemble des domaines associés aux formes parfaites comme un graphe localement fini; les domaines jouent le rôle des sommets du graphe, les faces des domaines celui des arêtes. Nous appellerons ce graphe, le graphe de Voronoï.

### 6.2. Connexité du graphe :

Nous allons montrer que le graphe de Voronoï est connexe. La démonstration est essentiellement basée sur le lemme suivant :

#### Lemme 7 :

*Soit  $f$ , la matrice d'une forme quadratique définie positive et  $\beta$ , un nombre réel positif. Il n'existe au plus qu'un nombre fini de matrices  $A$  associées à des formes parfaites de minimum 1 et vérifiant  $f \cdot A \leq \beta$ .*

#### Démonstration :

Comme  $A$  est définie positive, il existe une matrice  $S \in GL_n(\mathbb{R})$ , telle que  $A = S^t S$ . Appelons  $l_k$  le vecteur dont les composantes sont celles de la  $k^{\text{ème}}$  ligne de  $S$ . On montre facilement que  $f \cdot A = \sum_{k=1}^n l_k^t f l_k$ .  $f$  étant définie positive,

chacun des termes  $l_k^t f l_k$  doit être strictement inférieur à  $\beta$ , et la norme de  $S$  est bornée supérieurement par une constante qui ne dépend que de  $f$ .

La relation d'Hermite  $\mu(A) \leq \gamma_n$  montre que le déterminant de  $S$  ne peut devenir arbitrairement petit. Ainsi, par Cramer, la norme de  $S^{-1}$  est aussi bornée supérieurement par une constante qui ne dépend que de  $f$ . On en déduit qu'en valeur absolue, les composantes des vecteurs minimaux de  $A$  sont bornées supérieurement par une constante qui ne dépend que de  $f$ .

Considérons  $\mathcal{E}$ , l'ensemble des vecteurs  $v \in \mathbb{Z}^n$  dont les composantes sont, en valeur absolue, inférieures à cette constante. Cet ensemble est bien évidemment fini. On peut définir une application qui, à chaque matrice  $A$  satisfaisant les hypothèses du lemme, fait correspondre une partie de  $\mathcal{E}$  : l'ensemble des vecteurs minimaux de  $A$ . Vu que les matrices  $A$  sont associées à des formes parfaites de minimum 1, cette application est injective, ce qui permet de conclure.

□ L. 7

Pour démontrer la connexité du graphe de Voronoï, il suffit de montrer que deux domaines quelconques  $\mathcal{C}$  et  $\mathcal{C}'$  sont toujours connectés. Pour cela, considérons  $f$ , un point intérieur de  $\mathcal{C}'$ .  $f$  est toujours la matrice d'une forme quadratique définie positive.

Comme  $f$  est un point intérieur de  $\mathcal{C}'$ , si  $f$  appartient à  $\mathcal{C}$ , alors  $\mathcal{C} = \mathcal{C}'$ . Si, par contre,  $f$  n'appartient pas à  $\mathcal{C}$ , on peut trouver un hyperplan d'appui  $\pi$  de ce domaine, définissant une face de  $\mathcal{C}$ , tel que  $\mathcal{C}$  et  $f$  se situent de part et d'autre de  $\pi$ . On recommence alors le raisonnement en remplaçant  $\mathcal{C}$  par son domaine voisin relativement à la face  $\pi \cap \mathcal{C}$ .

Le lemme garantit que le processus s'arrête après un nombre fini d'itérations. En effet,  $f \cdot A$ , où  $A$  est la matrice de la forme parfaite de minimum 1 associé au domaine testé ( $\mathcal{C}$ ), décroît strictement lors de chaque itération.

**6.3. Actions de  $Gl_n(\mathbb{Z})$  :**

Considérons l'action de  $Gl_n(\mathbb{Z}) \times \mathbb{R}^+$  sur l'ensemble  $\mathcal{P}_n$  des formes parfaites à  $n$  variables :

$$\alpha : (Gl_n(\mathbb{Z}) \times \mathbb{R}^+) \times \mathcal{P}_n \longrightarrow \mathcal{P}_n, \quad (S; \lambda) \cdot q = \lambda q \circ S^{-1}$$

Les orbites de cette action correspondent aux classes d'équivalence des formes parfaites.

**Théorème 9 :**

*L'action  $\alpha$  décompose  $\mathcal{P}_n$  en un nombre fini d'orbites.*

Pour la démonstration, on se référera, par exemple, à [Vo 1] ou à [Oe 1].

Considérons la représentation  $\sigma$  de  $Gl_n(\mathbb{Z})$  sur l'espace de Voronoï  $\mathcal{V}$  :

$$\begin{array}{ccccccc} \sigma : & Gl_n(\mathbb{Z}) & \longrightarrow & Gl(\mathcal{V}) & & & \\ & S & \longmapsto & \sigma_S : \mathcal{V} & \longrightarrow & \mathcal{V} & \\ & & & x & \longmapsto & S \times S^t & \end{array}$$

Si  $a_k = v_k v_k^t$  caractérise une arête du domaine associé à  $q$ ,  $\sigma_S(a_k)$  caractérise une arête du domaine associé à  $q \circ S^{-1}$ . Par linéarité, on conclut que  $\sigma_S$  transforme le domaine associé à  $q$  en le domaine associé à  $q \circ S^{-1} = \alpha((S; 1); q)$ , et crée une bijection entre les faces de ces deux domaines.

Mieux : la permutation des domaines associés aux formes parfaites, induite par  $\sigma_S$ , est compatible avec la relation "être voisins". Par conséquent,  $\sigma$  permet de définir une action de  $Gl_n(\mathbb{Z})$  sur le graphe de Voronoï.

On dira que deux domaines sont équivalents s'ils appartiennent à la même orbite. On montre facilement que deux domaines sont équivalents si et seulement si les formes

parfaites correspondantes sont équivalentes. Il n'existe donc qu'un nombre fini de classes d'équivalence de domaines.

Le stabilisateur d'une forme parfaite par l'action  $\alpha$  est un sous-groupe de  $Gl_n(\mathbb{Z}) \times \{1\}$ . Sa projection sur  $Gl_n(\mathbb{Z})$  est appelée groupe des automorphismes de cette forme. Il faut comprendre "automorphisme" dans le sens d'isométries entières de cette forme.

Observons que le groupe des automorphismes d'une forme parfaite peut également être défini comme le stabilisateur du domaine associé à cette forme, par l'action de  $Gl_n(\mathbb{Z})$  sur le graphe de Voronoï. En effet, soit  $q$ , une forme parfaite, et  $\mathcal{C}$ , le domaine qui lui est associé;  $S \in Gl_n(\mathbb{Z})$  est un automorphisme de  $q$  si et seulement si  $q \circ S^{-1} = q$ , c'est-à-dire  $\sigma_S(\mathcal{C}) = \mathcal{C}$  puisque  $q$  est parfaite.

Pour rendre fidèle l'action du groupe des automorphismes d'une forme parfaite sur l'ensemble des arêtes du domaine associé à cette forme, il suffit de quotienter le groupe des automorphismes par  $\pm$ identité, puisque toute forme parfaite est connectée au sens de Coxeter [Co 1]. J'appellerai le groupe ainsi obtenu le groupe des symétries du domaine.

Finalement, il est clair que le groupe des automorphismes d'une forme parfaite est un groupe fini, puisque, si  $\pm v_1, \pm v_2, \dots, \pm v_s$  sont les vecteurs minimaux de cette forme, son groupe des automorphismes s'injecte de façon évidente dans le groupe des permutations de  $2s$  éléments.

**7. Quelques propriétés des graphes localement finis :**

Soit  $\Gamma$ , un graphe localement fini,  $\mathcal{A}$ , l'ensemble de ses sommets et  $\mathcal{Q}$ , l'ensemble de ses arêtes.

On définit deux applications naturelles :

$$\mathfrak{A} : \mathcal{A} \times \mathcal{A} \longrightarrow \{0, 1\}$$

$$(x; y) \longmapsto \begin{matrix} 1 & \text{si } x \text{ et } y \text{ sont liés par une arête} \\ 0 & \text{sinon} \end{matrix}$$

$$\mathfrak{S} : \begin{matrix} \mathcal{Q} & \longrightarrow & \mathcal{P}(\mathcal{A}) & \text{injective} \\ a & \longmapsto & \{x_a, y_a\} & \text{où } x_a \text{ et } y_a \text{ sont les deux sommets} \\ & & & \text{de } \Gamma \text{ liés par l'arête } a. \end{matrix}$$

Soit un groupe  $G$  opérant sur  $\Gamma$ .  $G$  opère également sur  $\mathcal{A}$  et  $\mathcal{Q}$ , et il y a compatibilité de ces deux actions. Il existe donc deux actions, disons  $\alpha_{\mathcal{A}}$  et  $\alpha_{\mathcal{Q}}$ ,

$$\alpha_{\mathcal{A}} : \begin{matrix} G \times \mathcal{A} & \longrightarrow & \mathcal{A} \\ (g; x) & \longmapsto & g \cdot x \end{matrix} \quad \text{et} \quad \alpha_{\mathcal{Q}} : \begin{matrix} G \times \mathcal{Q} & \longrightarrow & \mathcal{Q} \\ (g; a) & \longmapsto & g \cdot a \end{matrix}$$

satisfaisant les propriétés suivantes de compatibilité :

- (i)  $\mathfrak{A}(x, y) = \mathfrak{A}(g \cdot x, g \cdot y), \forall g \in G \text{ et } \forall x, y \in \mathcal{A}$
- (ii)  $\mathfrak{S}(g \cdot a) = \{g \cdot x_a, g \cdot y_a\}, \forall g \in G \text{ et } \forall a \in \mathcal{Q}$   
ou encore  $\{x_{g \cdot a}, y_{g \cdot a}\} = \{g \cdot x_a, g \cdot y_a\}, \forall g \in G \text{ et } \forall a \in \mathcal{Q}$

Définitions :

On dira que deux sommets  $x$  et  $y$  sont voisins si  $\mathfrak{A}(x, y) = 1$ . On appellera voisin de  $x$ , tout sommet  $y$  de  $\Gamma$  tel que  $\mathfrak{A}(x, y) = 1$ .

Considérons, dans un premier temps,  $\alpha_{\mathcal{A}}$ , l'action de  $G$  sur l'ensemble  $\mathcal{A}$  des sommets du graphe. Cette action partitionne  $\mathcal{A}$  en orbites.

Soit  $O_x = \{x_1, x_2, \dots, x_i, \dots\}$ , une telle orbite, et  $x$ , un sommet quelconque de  $O_x$ . Appelons  $V_x$ , l'ensemble des sommets voisins de  $x$  et  $G_x$ , le stabilisateur de  $x$  dans  $G$ .

On peut considérer la restriction de  $\alpha_{\mathcal{A}}$  à  $G_x \times V_x$ . Cette restriction nous fournit une action  $\alpha_x$  de  $G_x$  sur  $V_x$ . Cette action partitionne  $V_x$  en orbites. On appellera ces orbites, les orbites de voisins de  $x$ ; le nombre d'orbites de voisins de  $x$  est fini puisque, par hypothèse, le graphe est localement fini. Notons  $V_j^x$ , la  $j^{\text{ème}}$  orbite de voisins de  $x$ . Si deux sommets appartiennent à la même orbite  $V_j^x$ , il est clair qu'ils sont également dans la même orbite lorsqu'on regarde l'action de  $G$  sur  $\mathcal{A}$ .

Soit maintenant  $O_x = \{x_1, x_2, \dots, x_i, \dots\}$  et  $O_y = \{y_1, y_2, \dots, y_i, \dots\}$ , deux orbites de sommets, distinctes ou non, par rapport à l'action de  $G$  sur  $\mathcal{A}$ . Pour  $x$ , un sommet quelconque de  $O_x$ , on définit la fonction suivante :

$$f(x, O_y) = \sum_i \chi(V^i_x, O_y) \quad \text{où } \chi(V^i_x, O_y) \text{ vaut } 1 \text{ si } V^i_x \subset O_y, \text{ et } 0 \text{ sinon.}$$

$f(x, O_y)$  compte le nombre d'orbites de voisins de  $x$ , contenues dans  $O_y$ .

### **Théorème 10 :**

*Avec les notations ci-dessus :*

(i)  $f(x, O_y)$  est constante lorsque  $x$  parcourt  $O_x$ .

*On peut donc parler de  $f(O_x, O_y)$ .*

(ii) Avec la remarque du point (i),  $f$  est symétrique, c'est-à-dire

$$f(O_x, O_y) = f(O_y, O_x) \quad \forall O_x, O_y.$$

### **Démonstration :**

(i) Soit  $x$  et  $x'$ , deux sommets de la même orbite  $O_x$ . On doit démontrer que  $f(x, O_y) = f(x', O_y) \quad \forall O_y$ .

Comme  $x$  et  $x'$  sont dans la même orbite, il existe  $g \in G$  tel que  $g \cdot x = x'$ .

Soit  $V^1_x, V^2_x, \dots, V^k_x, \dots, V^q_x$ , les orbites de voisins de  $x$ .

Les translatés par  $g$  des orbites de voisins de  $x$  donnent les orbites de voisins de  $x'$ . On peut donc numérotter les orbites de voisins de  $x'$  de telle manière que  $V^{k'}_{x'} = g \cdot V^k_x \quad \forall k$ .

En observant que  $V^k_x$  et  $V^{k'}_{x'}$  sont contenues dans la même orbite de  $\mathcal{A}$  par rapport à l'action  $\alpha_{\mathcal{A}}$ , on déduit que  $\chi(V^k_x, O_y) = \chi(V^{k'}_{x'}, O_y) \quad \forall k, \forall O_y$ . En particulier,  $f(x, O_y) = f(x', O_y)$  quelle que soit l'orbite  $O_y$  considérée.

(ii) Si  $O_x = O_y$ , il n'y a rien à démontrer. Supposons  $O_x \neq O_y$ .

Remarquons d'abord que la relation "être voisin de" est symétrique. Par conséquent,  $f(O_x, O_y) = 0$  si et seulement si  $f(O_y, O_x) = 0$ . On supposera donc que  $f(O_x, O_y) > 0$ .

Soit  $y_1$  et  $y_2$ , deux sommets voisins de  $x$ , distincts ou non, contenus dans  $O_y$ .

Il existe  $g \in G$  tel que  $y_2 = g \cdot y_1$ . Choisissons un quelconque  $y$  dans  $O_y$ ; on trouvera  $g_i \in G$  tels que  $y = g_i \cdot y_i$  ( $i=1, 2$ ).

Remarque :  $1 = \mathfrak{A}(x, y_i) = \mathfrak{A}(g_i \cdot x, g_i \cdot y_i) = \mathfrak{A}(g_i \cdot x, y)$  signifie que  $g_i \cdot x$  et  $g_2 \cdot x$  sont voisins de  $y$ .

Si  $g_1 \cdot x$  et  $g_2 \cdot x$  sont dans la même orbite de voisins de  $y$ , il existe  $h' \in G_y$ , tel que  $h'(g_1 \cdot x) = g_2 \cdot x$ . Dans ce cas,  $h = (g_2^{-1} h' g_1)$  appartient à  $G_x$ , et envoie  $y_1$  sur  $y_2$ .

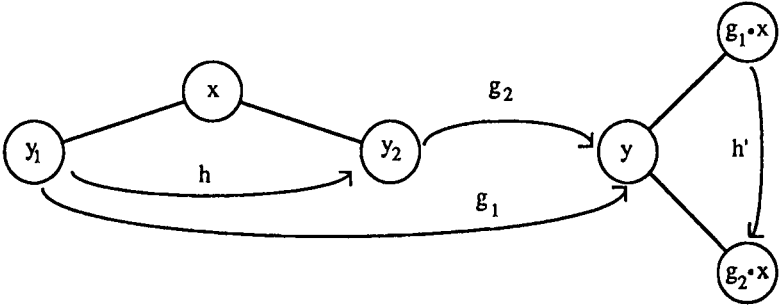


fig. 8

Par conséquent, si  $y_1$  et  $y_2$  appartiennent à des orbites de voisins de  $x$  différentes,  $g_1 \cdot x$  et  $g_2 \cdot x$  doivent aussi appartenir à des orbites de voisins de  $y$  différentes.

Ainsi,  $f(O_x, O_y) = f(x, O_y) \leq f(y, O_x) = f(O_y, O_x)$ .

En échangeant les rôles de  $O_x$  et  $O_y$ , on obtient  $f(O_y, O_x) \leq f(O_x, O_y)$ , ce qui termine la démonstration.

□ Th. 10

**Notations :** Si  $\mathcal{A}(x, y) = 1$ , l'orbite de voisins de  $x$  qui contient  $y$  sera notée  $V_x(y)$ . Si  $a$  est l'arête qui relie  $x$  et  $y$ , on notera  $G_a$ , le sous-groupe de  $G$  qui stabilise  $a$ .

**Lemme 8 :**

Soit  $x$  et  $y$ , deux sommets,  $x \in O_x$  et  $y \in O_y$ . On supposera  $O_x$  et  $O_y$  distinctes.

Si  $\mathcal{A}(x, y) = 1$  et si  $a$  est l'arête qui relie  $x$  et  $y$ , on a :

- (i)  $G_a = G_x \cap G_y$ ;
- (ii)  $|G_x| |V_y(x)| = |G_y| |V_x(y)|$ .

**Démonstration :**

(i) Soit  $g \in G_a$ , c'est-à-dire  $g \cdot a = a$ . La compatibilité des actions de  $G$  sur  $\mathcal{U}$  et de  $G$  sur  $\mathcal{A}$ , nous dit que  $\{x, y\} = \mathfrak{S}(a) = \mathfrak{S}(g \cdot a) = \{g \cdot x, g \cdot y\}$ . Comme  $O_x$  et  $O_y$  sont disjointes, on doit avoir  $g \cdot x = x$  et  $g \cdot y = y$ , ce qui signifie que  $g$  appartient à  $G_x$  et à  $G_y$ . Autrement dit,  $g$  appartient à  $G_x \cap G_y$ .

Inversement, si  $g \in G_x \cap G_y$ ,  $\mathfrak{S}(a) = \{x, y\} = \{g \cdot x, g \cdot y\} = \mathfrak{S}(g \cdot a)$ . Par injectivité de  $\mathfrak{S}$ , on conclut que  $g \cdot a = a$ , c'est-à-dire  $g \in G_a$ .

(ii) On remarquera, tout d'abord, que  $V_x(y)$  et  $V_y(x)$  sont toujours des ensembles finis puisque, par hypothèse, le graphe considéré est localement fini.

Si  $G_x$  et  $G_y$  sont des groupes infinis, il n'y a rien à démontrer. On supposera donc que  $G_x$  est un groupe fini. Par (i),  $G_a$  peut être vu comme le sous-groupe de  $G_x$  qui stabilise  $y$ . En particulier, c'est un groupe fini. Par conséquent,  $|G_x| = |G_a| \cdot |V_x(y)|$ . De même, en interprétant  $G_a$  comme le sous-groupe de  $G_y$  qui stabilise  $x$ , on conclut que  $|G_y| = |G_a| \cdot |V_y(x)|$  et, en particulier, que  $G_y$  est un groupe fini.

En éliminant  $|G_a|$  de ces deux équations, on obtient la relation annoncée.

□ L. 8

### **Lemme 9 :**

Soit  $x$  et  $x'$ , deux sommets appartenant à  $O_x$ . Supposons que  $\mathfrak{A}(x, x') = 1$ , et appelons  $a$ , l'arête qui relie  $x$  et  $x'$ ; alors :

- (i)  $G_x \cap G_{x'} \subset G_a$ ;
- (ii)  $G_x \cap G_{x'}$  est soit égal à  $G_a$ , soit d'indice 2 dans  $G_a$ ;
- (iii)  $|V_x(x)| = |V_{x'}(x')|$ .

### **Démonstration :**

(i) Comme avant, soit  $g \in G_x \cap G_{x'}$ ;

$\mathfrak{S}(a) = \{x, x'\} = \{g \cdot x, g \cdot x'\} = \mathfrak{S}(g \cdot a)$ . L'injectivité de  $\mathfrak{S}$  nous permet de conclure que  $g \cdot a = a$ , c'est-à-dire  $g \in G_a$ .

(ii) Premier cas : Aucun  $h \in G$  n'échange  $x$  et  $x'$ .

Dans ce cas, prenons  $g \in G_a$ ; alors,  $\{x, x'\} = \mathfrak{S}(a) = \mathfrak{S}(g \cdot a) = \{g \cdot x, g \cdot x'\}$ . L'hypothèse qu'aucun élément de  $G$  n'échange  $x$  et  $x'$  permet de conclure qu'obligatoirement,  $g \cdot x = x$  et  $g \cdot x' = x'$ . Donc,  $g$  appartient à  $G_x$  et à  $G_{x'}$ , ce qui signifie que  $g$  appartient à  $G_x \cap G_{x'}$ .

Le point (i) nous permet de conclure  $G_x \cap G_{x'} = G_a$ .

Second cas : Il existe  $h \in G$ , tel que  $x' = h \cdot x$  et  $x = h \cdot x'$ .

Observons que si  $g$  appartient à  $G_x \cap G_{x'}$ ,  $gh$  appartient à  $G_a$ . En effet,

$\mathfrak{S}(gh \cdot a) = \{(gh) \cdot x, (gh) \cdot x'\} = \{g \cdot (h \cdot x), g \cdot (h \cdot x')\} = \{g \cdot x', g \cdot x\} = \{x', x\}$ , ce qui permet de conclure, par injectivité de  $\mathfrak{S}$ , que  $a = (gh) \cdot a$ , c'est-à-dire  $gh \in G_a$ .

Ainsi,  $(G_x \cap G_{x'}) \cup h(G_x \cap G_{x'})$  est contenu dans  $G_a$ .

Pour montrer l'inclusion inverse, considérons  $g$  appartenant à  $G_a$ .

Obligatoirement,  $\{g \cdot x, g \cdot x'\} = \{x, x'\}$ .

Si  $g \cdot x = x$ , alors  $g \cdot x' = x'$  et, par conséquent,  $g \in G_x \cap G_{x'}$ .

Sinon,  $g \cdot x = x'$  et  $g \cdot x' = x$ . Dans ce cas,  $(h^{-1}g) \in G_x \cap G_{x'}$ ; autrement dit,  $g \in h(G_x \cap G_{x'})$ .

Ainsi,  $G_a$  est contenu dans  $(G_x \cap G_{x'}) \cup h(G_x \cap G_{x'})$ .

Finalement, cela démontre que  $G_a = (G_x \cap G_{x'}) \cup h(G_x \cap G_{x'})$ , d'où le fait que  $G_x \cap G_{x'}$  est un sous-groupe de  $G_a$  d'indice 2.

(iii) On regarde  $G_x \cap G_{x'}$  comme le sous-groupe de  $G_x$  qui stabilise  $a$ . La cardinalité de l'orbite  $V_x(x')$  est l'indice de  $G_x \cap G_{x'}$  dans  $G_x$ .

De même, la cardinalité de l'orbite  $V_{x'}(x)$  est l'indice de  $G_x \cap G_{x'}$  dans  $G_{x'}$ .

On conclut en observant que, comme  $x$  et  $x'$  sont tous deux dans  $O_x$ ,  $G_x$  et  $G_{x'}$  sont conjugués.

□ L. 9

## 8. Algorithme de Voronoï :

### 8.1. Description :

Cet algorithme permet de calculer tous les maxima locaux de la fonction  $\mu$  et, en particulier, la valeur de  $\gamma_n$ . On peut vérifier que, modulo les homothéties, tous les maxima locaux de  $\mu$  sont isolés.

L'algorithme de Voronoï ramène le calcul des formes extrêmes à l'énumération complète des classes de formes parfaites. En effet, Voronoï commence par démontrer son fameux théorème : *une forme quadratique définie positive est extrême si et seulement si elle est parfaite et eutactique.*

On rencontre dans la littérature plusieurs démonstrations de ce théorème. A part celle de Voronoï lui-même, je citerai, entre autres, celle de Barnes [Ba 4] basée sur un lemme de Stiemke [Sti 1], celle de Martin Kneser utilisant la géométrie des convexes [Kn 1] et celle, plus générale, d'Anne-Marie Bergé et de Jacques Martinet [BM 2].

Pour obtenir une énumération complète, à équivalence près, des formes parfaites à  $n$  variables, l'algorithme de Voronoï se base, d'une part, sur la connexité du graphe de Voronoï, d'autre part, sur la finitude du nombre de classes d'équivalence des domaines associés aux formes parfaites.

Partant d'un domaine quelconque, on mémorise sa classe d'équivalence et on détermine tous ses domaines voisins. Certains d'entre eux n'appartiennent peut-être pas à la classe de départ. Dans ce cas, on obtiendra une liste de nouvelles classes d'équivalence. On mémorise ces nouvelles classes et, pour chacune d'entre elles, on choisira un représentant. Pour chacun de ces représentants, on déterminera à nouveau ses domaines voisins, domaines qui, à leur tour, caractériseront parfois de nouvelles classes, etc.

La connexité du graphe garantit qu'en répétant ce processus toutes les classes seront atteintes, quel que soit le domaine de départ choisi. La finitude du nombre de classes d'équivalence prouve que l'algorithme s'arrête.

L'énumération de toutes les classes de domaines associés aux formes parfaites (donc de toutes les classes de formes parfaites) est complète lorsque, pour chaque classe connue, les voisins d'un représentant quelconque appartiennent tous à des classes déjà connues.

### 8.2. Symétries :

Dans la pratique, pour déterminer tous les domaines voisins d'un domaine, il faut d'abord calculer toutes les faces de ce domaine. C'est un problème de program-

mation linéaire entière dont la complexité croît très rapidement en fonction du nombre  $s$  de paires de vecteurs minimaux.

Soit  $q$ , une forme parfaite et  $D$ , le domaine qui lui est associé. Le groupe des symétries de  $D$  agit sur les faces de  $D$ . On dira que deux faces sont équivalentes si elle appartiennent à la même orbite. Il est clair que deux faces équivalentes donneront lieu à deux domaines voisins équivalents. Pour appliquer l'algorithme de Voronoï, il suffit de connaître au moins une face dans chaque orbite.

Géométriquement, cela revient à utiliser les symétries des domaines, ce qui simplifie parfois considérablement le problème. On trouvera une illustration de cette simplification dans [JS 1].

### 8.3. Calcul des faces d'un domaine :

Lorsque le nombre d'arêtes d'un cône convexe  $C$  est égal à sa dimension  $d$ , le calcul des faces est instantané. Il suffit, en effet, de retirer à tour de rôle chacune des arêtes; celles qui restent déterminent une face.

Par contre, lorsque le nombre d'arêtes est supérieur à cette dimension, il faut remarquer que, si l'on aborde la recherche des faces d'un point de vue combinatoire (test de tous les  $(d-1)$ -tuples d'arêtes), le problème explose rapidement. C'est là qu'interviennent mes algorithmes de la cascade et de l'explorateur.

#### 8.3.1. Algorithme de la cascade :

##### 8.3.1.1. Description :

Considérons le relèvement de  $C$  (cf 4.2) avec  $d = \dim C = N = \frac{n(n+1)}{2}$ . On supposera que  $C$  possède  $s$  arêtes.

L'algorithme de la cascade se base sur le théorème fondamental du relèvement (th. 4, cf 4.2.2) qui garantit que toute face de  $C^{k-1}$  peut s'obtenir par projection par  $\pi^k$ , soit d'une face de  $C^k$ , soit de l'intersection de deux faces de  $C^k$ .

Le principe de cet algorithme consiste à calculer d'abord les faces de  $C^s$  (sommet de la cascade), puis, à partir des faces de  $C^s$ , à calculer celles de  $C^{s-1}$  (premier saut) en employant le fait que  $C^{s-1} = \pi^s(C^s)$ , puis, à partir des faces de  $C^{s-1}$ , à calculer celles de  $C^{s-2}$  (second saut) en employant le fait que  $C^{s-2} = \pi^{s-1}(C^{s-1})$ , et ainsi de suite, jusqu'au dernier saut qui, à partir des faces de  $C^{N+1}$ , permet de calculer celles de  $C^N$ .<sup>1</sup>

---

<sup>1</sup> Le temps de calcul est ainsi considérablement réduit.

8.3.1.2. **Symétries :**

Dans ce qui suit, on considérera  $\mathcal{C}$ , un domaine de Voronoï associé à une forme parfaite  $q$ .

Un élément du groupe des symétries de ce domaine peut être représenté indifféremment et fidèlement par une matrice dans  $Gl_n(\mathbb{Z})$ , par un automorphisme de l'espace de Voronoï qui conserve le domaine, ou par une permutation des arêtes du domaine.

Par ce qui précède, le groupe des symétries de ce domaine peut être vu comme un groupe  $G$  d'automorphismes de  $E^N$  qui conserve  $\mathcal{C}$ . Le relèvement de ce cône ordonne les arêtes du domaine. Pour  $k = N, \dots, s$ , définissons  $G^k$  par :

$$G^k = \{g \in G \mid g \text{ fixe les } k - N, \text{ dernières arêtes du domaine}\}.$$

Il est clair que  $G = G^N > G^{N+1} > \dots > G^s$ . D'autre part, il y a une représentation canonique de  $G^k$  sur  $E^k \supset E^N$  :

$$\begin{array}{ccc} G^k & \longrightarrow & Gl(E^k) \\ g & \longmapsto & g^k : E^k = E^N \oplus E^\perp \longrightarrow E^k = E^N \oplus E^\perp \\ & \text{avec} & g^k|_{E^N} = g \quad \text{et} \quad g^k|_{E^\perp} = \text{identité.} \end{array}$$

Comme  $g \in G^k$  laisse fixes les  $k - N$  dernières arêtes de  $\mathcal{C}$ ,  $g^k$  laisse fixes les  $k - N$  dernières arêtes de  $\mathcal{C}^k$ . Par construction, seules, exactement, les  $k - N$  dernières arêtes de  $\mathcal{C}^k$  ne sont pas dans  $E^N$ . Ainsi,  $g^k$  est un automorphisme de  $E^k$  qui conserve  $\mathcal{C}^k$ . Cette remarque nous permet de définir le groupe des symétries de  $\mathcal{C}^k$ ; il s'agit de l'image de  $G^k$  par la représentation ci-dessus.

Dans la pratique, calculer  $G^k$  peut être très long. On se contentera en général d'un sous-groupe. Soit donc  $G = G^N = H^N > H^{N+1} > \dots > H^s$  avec, pour tout  $k$ ,  $H^k < G^k$ .

Dans ce qui suit, une telle suite de sous-groupes  $H^k$  sera fixée définitivement; de plus, on représentera  $H^k$  comme un sous-groupe de permutations des arêtes du domaine, ce qui nous permet d'identifier canoniquement  $H^k$  et son image par la représentation ci-dessus.

Par un élément de  $H^k$ , toute facette de  $\mathcal{C}^k$  est transformée en une facette de  $\mathcal{C}^k$  de même dimension. On dira que deux facettes de  $\mathcal{C}^k$  sont équivalentes s'il existe un élément de  $H^k$  qui envoie la première sur la seconde.

Plus généralement, si l'on considère l'ensemble des arêtes de  $\mathcal{C}^k$ , on peut définir une notion d'équivalence entre deux parties de cet ensemble. Deux parties sont équivalentes s'il existe un élément de  $H^k$  qui envoie la première sur la seconde. Comme  $H^k < H^{k-1}$ , on observe que ces équivalences sont compatibles avec la projection  $\pi^k$  qui envoie  $\mathcal{C}^k$  sur  $\mathcal{C}^{k-1}$ .

On dira qu'une partie de l'ensemble des arêtes d'un cône convexe de dimension maximale caractérise une face  $F$  donnée de ce cône si les arêtes de cette partie sont dans  $F$  et si l'enveloppe linéaire des représentants des arêtes de cette partie est un sous-espace de codimension 1.

Ecrivons les orbites de faces de  $\mathcal{C}^k$  sous l'action de  $H^k$  en mettant en évidence un représentant pour chaque orbite :

$(F_1^k, F_{1,2}^k, \dots, F_{1,l_1}^k)$ , représentant  $F_1^k$ ;

$(F_2^k, F_{2,2}^k, \dots, F_{2,l_2}^k)$ , représentant  $F_2^k$ ;

etc.

### **Lemme 10 :**

*Si une face de  $\mathcal{C}^{k-1}$  est caractérisée par la projection des arêtes contenues dans l'intersection de deux faces de  $\mathcal{C}^k$ , elle est équivalente par l'action de  $H^{k-1}$  à une face caractérisée par la projection des arêtes contenues dans l'intersection de deux faces de  $\mathcal{C}^k$ , où la première face est le représentant choisi de son orbite.*

### **Démonstration :**

Soit deux orbites de faces de  $\mathcal{C}^k$ ,  $\mathcal{O}$  et  $\mathcal{O}'$  (distinctes ou non),  $\mathcal{O} = \{F = F_1^k, F_2^k, \dots, F_l^k\}$  et  $\mathcal{O}' = \{F' = F'_1^k, F'_2^k, \dots, F'_{l'}^k\}$ . Soit encore  $F_i^k \in \mathcal{O}$  et  $F'_j^k \in \mathcal{O}'$ , deux faces de  $\mathcal{C}^k$ .

Supposons que les arêtes de  $\pi^k(F_i^k \cap F'_j^k)$  caractérisent une face de  $\mathcal{C}^{k-1}$ . Comme  $F$  et  $F_i^k$  appartiennent à  $\mathcal{O}$ , il existe  $g \in H^k$ , tel que  $F = g \cdot F_i^k$ .

Comme  $g$  appartient à  $H^k < H^{k-1}$ ,  $g$  doit appartenir à  $H^{k-1}$ ; ainsi, affirmer que les arêtes de  $\pi^k(F_i^k \cap F'_j^k)$  caractérisent une face de  $\mathcal{C}^{k-1}$  équivaut à dire que les arêtes de  $g \cdot (\pi^k(F_i^k \cap F'_j^k))$  caractérisent une face de  $\mathcal{C}^{k-1}$ , face qui est équivalente à la première par l'action de  $H^{k-1}$ .

On vérifiera sans peine que  $\pi^k$  et  $g$  commutent, en observant que la permutation des indices des arêtes induite par  $\pi^k$  n'est autre que la permutation identité.

Ainsi, dire que les arêtes de  $g \cdot (\pi^k(F_i^k \cap F'_j^k))$  caractérisent une face de  $\mathcal{C}^{k-1}$ , revient à dire que les arêtes de  $\pi^k(g \cdot (F_i^k \cap F'_j^k))$  caractérisent une face de  $\mathcal{C}^{k-1}$ .

On peut maintenant conclure en remarquant que :

$$g \cdot (F_i^k \cap F'_j^k) = (g \cdot F_i^k) \cap (g \cdot F'_j^k) = F \cap (g \cdot F'_j^k).$$

□

□ L. 10

Ce lemme technique permet d'éviter un nombre parfois considérable de tests de faces potentielles, lorsqu'on effectue un saut dans l'algorithme de la cascade. En effet, il suffit de tester, en les projetant, les facettes obtenues par le procédé suivant :

- on numérote les orbites de faces de  $C^k$  et les éléments de ces orbites;
- on prend la première face de la première orbite et on considère successivement les intersections de cette face avec toutes les faces de  $C^k$ ;
- on prend ensuite la première face de la seconde orbite et on considère successivement les intersections de cette face avec toutes les faces de  $C^k$  appartenant à une orbite dont le numéro est supérieur ou égal à 2;
- on continue avec la première face de la troisième orbite et on considère successivement les intersections de cette face avec toutes les faces de  $C^k$  appartenant à une orbite dont le numéro est supérieur ou égal à 3;
- et ainsi de suite jusqu'à la dernière orbite.

Pour fixer les idées, si  $C^k$  possède  $x$  orbites contenant chacune  $y$  faces, le rapport entre le nombre de cas testés en appliquant le lemme 10 et le nombre de cas testés sans employer ce lemme est donné par  $\frac{x+1}{xy+1}$ , qui est de l'ordre de  $\frac{1}{y}$ .

### 8.3.1.3. Implémentation en LISP :

Un domaine de Voronoï est entièrement caractérisé par les représentants de ses arêtes. Nous avons vu que les arêtes d'un domaine sont exactement les demi-droites dont on prend l'enveloppe convexe pour définir ce domaine. Si  $\pm v_1, \pm v_2, \dots, \pm v_s$  sont les  $s$  paires de vecteurs minimaux de la forme parfaite  $q$  (de minimum 1) correspondant à ce domaine, les  $a_k = v_k v_k^1$  ( $k = 1, \dots, s$ ) caractérisent donc les arêtes du domaine.

Les points, dans l'espace de Voronoï, sont des matrices symétriques  $n \times n$ . Nous allons représenter ces points par des vecteurs de  $\mathbb{R}^N$  où  $N = \frac{n(n+1)}{2}$ . Plus précisément, une matrice symétrique  $A = (a_{ij})$  sera représentée par le vecteur suivant (en notation transposée...) :  $(a_{11} \ a_{12} \dots \ a_{1n} \ a_{22} \ a_{23} \dots \ a_{2n} \ a_{33} \dots \ a_{nn})$ .

Cet isomorphisme d'espace vectoriel transforme le domaine de Voronoï en un cône convexe saillant  $C'$  de  $\mathbb{R}^N$  dont les arêtes sont exactement caractérisées par les vecteurs images des  $a_k$  ( $k = 1, \dots, s$ ). Il est clair que cette transformation envoie toute facette de dimension  $d$  du domaine de Voronoï sur une facette de dimension  $d$  de  $C'$ .

Le calcul des faces du domaine se ramène donc au calcul des faces de  $C'$ .

Pour des raisons de commodité, le produit scalaire utilisé pour les calculs dans  $\mathbb{R}^N$  sera le produit scalaire usuel. Il est évident que le produit scalaire usuel de  $\mathbb{R}^N$  ne correspond pas directement au produit scalaire défini par la trace dans l'espace de

Voronoi, mais cela n'a que peu d'importance. On calculera facilement, lorsque cela est nécessaire, un vecteur normal - au sens du produit scalaire défini par la trace - à une face du domaine, à partir d'un vecteur normal à une face de  $C'$  - au sens du produit scalaire usuel de  $\mathbb{R}^N$ .

Explicitement,  $(b_{11} \ b_{12} \dots \ b_{1n} \ b_{22} \ b_{23} \dots \ b_{2n} \ b_{33} \dots \ b_{nn})$  sera traduit en une matrice  $B = (b'_{ij})$  avec  $b'_{ii} = b_{ii}$  et  $b'_{ij} = \frac{1}{2} b_{ij}$ , si  $i$  est différent de  $j$ .<sup>2</sup>

Notre problème est donc entièrement ramené à un problème de programmation linéaire entière dans  $\mathbb{R}^N$  muni du produit scalaire usuel. Pour ne pas introduire de nouvelles notations, on supposera que les  $a_1, a_2, \dots, a_s$  sont  $s$  vecteurs de  $\mathbb{R}^N$  qui caractérisent les arêtes de  $C'$ . Après renumérotation éventuelle, on peut supposer que les  $N$  premiers vecteurs de cette liste sont linéairement indépendants, puisque la forme est parfaite.

Notons  $\mathfrak{M}$ , la  $N \times s$ -matrice dont les colonnes sont les  $a_j$ . Cette matrice contient toute l'information de notre problème. Par des opérations élémentaires de lignes uniquement, on peut transformer  $\mathfrak{M}$  en une matrice dont le bloc  $N \times N$  de gauche est une matrice identité. Donc, il existe une matrice  $B$  inversible, telle que :

$$B \mathfrak{M} = \begin{pmatrix} 1 & 0 & \dots & 0 & * & \dots & * \\ 0 & 1 & \dots & 0 & * & \dots & * \\ \vdots & & & & \vdots & & \vdots \\ 0 & 0 & \dots & 1 & * & \dots & * \end{pmatrix}.$$

<sup>2</sup> Les différents types d'enregistrement utilisés pour mémoriser les faces sont les suivants :

- type "mat" la face est enregistrée sous forme d'une matrice mémorisant un vecteur normal, dans l'espace de Voronoi, à l'hyperplan d'appui qui lui correspond.
- type "mat-l" comme pour le type "mat", mis à part le choix de la structure de mémoire interne : la matrice est enregistrée sous forme d'un vecteur à  $n$  composantes où chaque composante de ce vecteur est une liste à  $n$  éléments, qui mémorise une ligne de la matrice; cette représentation interne permet d'effectuer des opérations de lignes sur la matrice de manière très performante; par exemple, échanger deux lignes d'une telle matrice revient à échanger deux composantes d'un vecteur!
- type "vec" la face est enregistrée sous forme d'un vecteur à  $N$  composantes, mémorisant un vecteur normal, dans  $\mathbb{R}^N$ , à l'hyperplan d'appui qui lui correspond.
- type "liste" comme pour le type "vec", mis à part le choix de la structure de mémoire interne : le vecteur à  $N$  composantes est remplacé par une liste à  $N$  composantes.
- type "ent" entier dont le  $i^{\text{ème}}$  bit vaut 0 si et seulement si la  $i^{\text{ème}}$  arête du domaine est dans la face.

Cette dernière représentation est la plus économique du point de vue de la place mémoire; d'autre part, c'est celle qui se généralise le mieux lorsqu'on veut mémoriser une facette quelconque de  $C'$  ou relever  $C'$  en vue d'appliquer l'algorithme de la cascade. Lorsqu'une face est représentée par un vecteur normal, le sens du vecteur sera toujours choisi de telle manière que le produit scalaire entre ce vecteur normal et le représentant d'une arête du domaine ne soit jamais négatif.

Diverses fonctions sont programmées pour passer d'une représentation à une autre lorsque cela est nécessaire.

Par exemple, la traduction d'une face de type "liste" en une face de type "mat" est effectuée par la fonction *face-mat*.

Par cette transformation linéaire inversible,  $C'$  est envoyé sur un cône convexe saillant  $C$ , de dimension maximale, dont les arêtes sont caractérisées par les colonnes de  $B\mathfrak{M}$ . Pour les mêmes raisons que tout à l'heure, les faces de  $C'$  sont envoyées bijectivement sur les faces de  $C$ , et, plus généralement,  $B$  crée une bijection entre les facettes de  $C'$  et celles de  $C$ . Si  $v = (v_1, v_2, \dots, v_N)$  est un vecteur normal à une face de  $C$ , on vérifie sans peine que  $vB$  nous donne un vecteur normal à la face correspondante de  $C'$ .

Notre problème de départ est donc ramené au calcul des faces de  $C$ .<sup>3</sup>

Lorsque le nombre d'arêtes du domaine est égal à sa dimension, on devine les vecteurs normaux aux faces de  $C$ . Le vecteur  $e_i = (0, 0, \dots, 0, 1, 0, \dots, 0)$ , vecteur dont toutes les composantes sont nulles, excepté la  $i^{\text{ème}}$  qui vaut 1, est perpendiculaire à la  $i^{\text{ème}}$  face. Les lignes de la matrice  $B$  définissent, dans ce cas, les vecteurs normaux aux faces de  $C'$ .<sup>4</sup>

Dans le cas où le nombre d'arêtes est supérieur à la dimension, on effectue le relèvement de  $C$ , en ayant préalablement réordonné les  $s - N$  dernières arêtes de telle sorte que le calcul des groupes de symétries  $H^k$  soit optimal. Mais on reviendra plus tard sur le choix et le calcul effectif des  $H^k$ .<sup>5</sup>

Pour visualiser le relèvement de  $C$ , on peut contempler la matrice carrée  $s \times s$  suivante, obtenue à partir de la matrice  $B\mathfrak{M}$  :

$$B\mathfrak{M} \longrightarrow \begin{pmatrix} 1 & 0 & \dots & 0 & * & * & \dots & * & * \\ 0 & 1 & \dots & 0 & * & * & \dots & * & * \\ \vdots & & & & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \dots & 1 & * & * & \dots & * & * \\ \begin{pmatrix} 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 1 & 0 \\ \vdots & \vdots & & \vdots & & & & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 1 & 0 & \dots & 0 & 0 \end{pmatrix} \end{pmatrix} \text{matrice } (s - N) \times N$$

<sup>3</sup> Dans le programme, la matrice  $B$  est enregistrée dans la structure  $B-I$ . La matrice  $B\mathfrak{M}$ , n'est pas entièrement mémorisée; en effet, seule sa partie de droite (\*\*), qui contient l'information intéressante, est stockée dans une structure appelée *matmin*.

La procédure qui ordonne le calcul de  $B-I$  et celui de *matmin* est *enumerer-faces*. C'est elle aussi qui gère les différentes étapes de l'algorithme de la cascade et provoque le traitement séparé du cas particulier où le nombre d'arêtes est égal à la dimension du domaine.

<sup>4</sup> La procédure qui traite ce cas particulier et regroupe ces faces en orbites s'appelle *normales-arêtes-faces*.

<sup>5</sup> Le calcul des faces, au sommet de la cascade, est effectué par *init-faces*; ensuite, tout saut dans l'algorithme de la cascade est effectué par *projeter*, mis à part le dernier qui est réalisé par *projeter-naf*.

On appellera cette grande matrice la matrice  $B_{\mathcal{F}}$  augmentée. Elle est de rang maximal. Par construction du relèvement,  $C^k$  est l'enveloppe convexe des arêtes caractérisées par les colonnes de cette matrice, pour peu qu'on ne considère que les  $k$  premières lignes de cette matrice.

Une face étant caractérisée par la liste des arêtes qu'elle contient, elle sera mémorisée dans le cadre de cet algorithme, sous forme d'un entier à  $s$  bits (0 ou 1) dont le  $i^{\text{ème}}$  bit vaut 0 si et seulement si la  $i^{\text{ème}}$  arête de  $C^k$  est dans cette face. Avec cette représentation, l'intersection de deux faces se calcule simplement en appliquant un "ou" logique entre les deux entiers qui les représentent.

On dira qu'un entier caractérise une face si les arêtes correspondant aux bits 0 de cet entier caractérisent cette face. En d'autre terme, cela signifie que cet entier retient suffisamment d'arêtes de cette face. On dira qu'un entier mémorise une face s'il retient exactement toutes les arêtes de cette face. Il peut arriver que deux entiers différents caractérisent la même face; par contre, deux entiers différents (à  $s$  bits significatifs) ne peuvent jamais mémoriser la même face.

Pour appliquer l'algorithme de la cascade, le théorème 4 montre qu'il nous faut une procédure capable de déterminer si un entier donné mémorise une face de  $C^k$ . J'ai développé une procédure, un peu plus générale, qui permet de déterminer, tout aussi rapidement, si un entier donné caractérise une face de  $C^k$ . Il s'agit de vérifier que les représentants des arêtes retenues par cet entier engendrent un hyperplan et que toutes les arêtes de  $C^k$  sont du même côté de cet hyperplan. La méthode standard procède de la manière suivante :

- 1) Construire une matrice dont les colonnes sont les représentants des arêtes retenues par notre entier.
- 2) Vérifier que ces représentants engendrent un hyperplan, c'est-à-dire que le rang de cette matrice est inférieur de 1 au nombre de ses lignes et vaut donc  $k - 1$ .
- 3) Calculer un vecteur normal à cet hyperplan.
- 4) Vérifier que le produit scalaire de ce vecteur avec les représentants des arêtes ne change pas de signe.

L'inconvénient de cette méthode réside dans sa lenteur...

Pour chaque face potentielle à tester, il faut en quelque sorte inverser une matrice  $k \times k$ ; ensuite seulement, on peut décider si l'hyperplan est bien positionné.<sup>6</sup>

La méthode que j'ai développée est beaucoup plus fine. Des tests d'arrêts (pour les mauvais candidats) ont lieu plus rapidement; on peut souvent éliminer une face

<sup>6</sup> Dans la situation qui nous intéresse ( $n=7$ ,  $N=28$ ),  $k$  varie entre 28 et 63.

potentielle avant même de savoir si sa codimension est bonne; on travaille avec des matrices dont les dimensions sont beaucoup plus petites.<sup>7</sup>

Pour comprendre cette méthode, regardons plus attentivement la matrice  $B\eta$  augmentée. Les colonnes de cette matrice sont de trois types, disons (a), (b) et (c). Les  $N$  premières sont de type (a), les  $k - N$  dernières de type (c) et celles qui restent de type (b).

Soit  $F$ , une face de  $C^k$ ; deux cas peuvent se présenter :

- (i)  $F$  contient toutes les arêtes de type (c);
- (ii) une seule arête de type (c) n'est pas dans  $F$ .

En effet, si davantage d'arêtes de type (c) n'étaient pas dans  $F$ , la codimension de  $F$  ne pourrait être égale à 1.

Le cas (ii) se traite théoriquement. Les  $(k-N)$  faces de ce type ne sont, en fait, pas même mémorisées. Il suffit presque de savoir qu'elles existent! Leurs vecteurs normaux sont les  $e_j$  pour  $N < j \leq k$ . Ces faces sont des faces maximales et leurs orbites sont réduites à un élément. Les calculs chercheront uniquement à caractériser les orbites de faces satisfaisant les hypothèses du cas (i). Seule précaution à prendre : lorsqu'on passe de  $C^{k+1}$  à  $C^k$ , la  $(k+1-N)^{\text{ème}}$  avant-dernière arête qui est de type (c) pour  $C^{k+1}$ , devient de type (b) pour  $C^k$ .

La projection par  $\pi^{k+1}$  de l'intersection d'une face maximale de  $C^{k+1}$  ne contenant pas une des  $(k-N)$  dernières arêtes avec une face quelconque de  $C^{k+1}$ , ne peut caractériser qu'une face de  $C^k$  satisfaisant les hypothèses du cas (ii). Par conséquent, la seule orbite de faces de  $C^{k+1}$ , non mémorisée, qui peut intervenir dans le calcul des faces (satisfaisant les hypothèses du cas (i)) de  $C^k$ , est l'orbite à un élément qui correspond à la face maximale de  $C^{k+1}$  ne contenant pas la  $(k+1-N)^{\text{ème}}$  avant-dernière arête et qui satisfait les hypothèses du cas (ii) en dimension  $k + 1$ .

Avant de commencer le calcul des faces de  $C^k$  satisfaisant les hypothèses du cas (i), on complétera donc la liste des orbites mémorisées de faces de  $C^{k+1}$  en lui adjoignant l'orbite à un élément décrite ci-dessus.

Dans le cas (i), une arête au moins de type (a) n'est pas dans la face. Sinon, la face serait un cône de dimension maximale!

Dans la description qui suit, on suppose donné un entier caractérisant une face potentielle  $F$  devant satisfaire les hypothèses du cas (i). Il s'agit de tester si, oui ou non, les arêtes retenues dans cet entier caractérisent effectivement une face.

---

<sup>7</sup> en général inférieures à dix, pour les formes à sept variables.

Les  $(k-N)$  dernières composantes du vecteur normal à cette face potentielle n'ont, en fait, jamais besoin d'être calculées. La  $j^{\text{ème}}$  composante ( $N < j \leq k$ ) du vecteur normal ne peut influencer que le produit scalaire entre ce vecteur normal et un quelconque vecteur caractérisant la  $(s+N+1-j)^{\text{ème}}$  arête, arête de type (c). Cette composante est définie a posteriori, dès que l'on connaît les  $N$  premières composantes du vecteur normal, de telle façon que la  $(s+N+1-j)^{\text{ème}}$  arête soit perpendiculaire au vecteur normal. En réalité, elle n'est jamais calculée explicitement parce que seules les  $N$  premières composantes peuvent jouer un rôle décisif. Pour les mêmes raisons, les arêtes de type (c) n'interviennent plus directement dans le calcul.

<sup>8</sup> Si exactement une arête de type (a) n'est pas retenue dans notre entier, le seul vecteur normal à  $F$  possible (à multiple positif près) coïncide sur ses  $N$  premières composantes avec le vecteur  $e_i$ , où  $i$  est l'indice de l'arête de type (a) exclue. Dans ce cas, la codimension ne pouvant être supérieure à 1, il suffit de vérifier que la  $i^{\text{ème}}$  composante des représentants des arêtes de type (b) retenues dans  $F$  est toujours nulle (vérification que la codimension vaut exactement 1) et que la  $i^{\text{ème}}$  composante des représentants des autres arêtes de type (b) n'est jamais négative (l'hyperplan considéré est un hyperplan d'appui). Si c'est le cas, notre entier caractérise une face  $F$ .

Pour que cet entier mémorise cette face  $F$ , il faut mettre à 0 les bits correspondant aux arêtes de type (b) non encore retenues et dont la  $i^{\text{ème}}$  composante du représentant est nulle. En effet, ces arêtes appartiennent à  $F$ .

Etudions maintenant la situation générale, c'est-à-dire celle où deux arêtes au moins de type (a) ne sont pas retenues dans notre entier.

Les composantes du vecteur normal correspondant aux arêtes de type (a) retenues dans notre entier doivent valoir 0. Les autres doivent être positives ou nulles. L'influence des arêtes de type (a) s'arrête là. Seules les arêtes de type (b) interviendront dans la suite du calcul.

Pour ne pas stocker inutilement des 0, on ne mémorisera pas la totalité des  $N$  premières composantes du vecteur normal à  $F$ , mais uniquement celles qui correspondent à des arêtes de type (a) non retenues. Ce "sous-vecteur" sera noté  $\omega$ . Supposons qu'il y ait  $t$  arêtes de type (a) non retenues dans l'entier à tester;  $\omega$  possède donc exactement  $t$  composantes.

On extrait de  $B\mathfrak{M}$ , <sup>9</sup> une petite matrice  $T$  <sup>10</sup>, dont les lignes sont formées des représentants des arêtes de type (b) retenues dans la face potentielle, représentants dont

<sup>8</sup> Les calculs décrits ci-après sont effectués, dans le programme, par la fonction *face?* ou par une de ses variantes.

<sup>9</sup> *matmin* dans le programme.

<sup>10</sup> *mat-1-t* dans le programme.

on ne prend que les composantes correspondant aux arêtes de type (a) non retenues. On obtient ainsi une matrice à  $t$  colonnes,  $t \geq 2$ .

Pour que la face potentielle ait la bonne dimension, il faut et il suffit que le rang de  $T$  vaille  $(t - 1)$ . En particulier, il est nécessaire que le nombre de lignes de  $T$  soit supérieur ou égal à  $(t - 1)$ . Si cette condition sur les dimensions de  $T$  n'est pas satisfaite, il n'est pas nécessaire de poursuivre le test : l'entier dont on est parti ne caractérise pas une face.

Dans l'exemple qui suit, la matrice  $T$  obtenue est une matrice possédant 6 lignes et 5 colonnes.

**Exemple :** ( $n=4, N=10, 8$  arêtes de type (b))

entier à tester écrit en base 2 :  $0\dots 0\ 1\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 0\ 1\ 0\ 0\ 1\ 0$

(6 arêtes de type (b) sont retenues,  $t = 5$ )

	arêtes de type (a)	arêtes de type (b)	arêtes de type (c)
	0 1 0 0 1 0 1 1 1 0	0 0 0 0 1 0 0 1	0 . . . 0
0			
1		* * * *	* *
0			
0			
1		* * * *	* *
0			
1		* * * *	* *
1		* * * *	* *
1		* * * *	* *
0			

fig. 9

Si  $T$  possède suffisamment de lignes, on cherche  $\omega$  perpendiculaire aux lignes de cette matrice. Je rappelle que les composantes de  $\omega$  doivent être positives ou nulles. Effectuer des opérations de lignes élémentaires sur  $T$  n'affecte pas  $\omega$ . Par des opérations de lignes élémentaires, on essaie de trianguler supérieurement la matrice  $T$  en mettant des 1 sur la diagonale. Si le rang de  $T$  est inférieur à  $t$ , le processus de triangulation doit s'arrêter avant qu'on ait pu traiter la dernière colonne de  $T$ .

Si le processus s'arrête et si  $j$  colonnes ont pu être traitées,  $0 \leq j < t$ , on pose la  $(j+1)^{\text{ème}}$  composante de  $\omega$  égale à 1 et les suivantes (s'il y en a) moralement égales à 0.

Les composantes précédentes de  $\omega$  peuvent maintenant être calculées à l'aide des lignes supérieures de la matrice. Si une composante est négative, on est déjà sûr que l'entier de départ ne caractérise pas une face, bien que l'on ne connaisse pas encore le rang de T et bien que le vecteur normal ne soit pas encore entièrement calculé. Si une composante est nulle, on introduit l'arête correspondante de type (a) dans la face potentielle.

Lorsque les composantes de  $\omega$  sont connues, on introduit dans la face potentielle toutes les arêtes de type (a) correspondant aux dernières composantes de  $\omega$  posées moralement égales à 0.

A ce stade, le rang de T n'est pas encore forcément connu. Si nécessaire, on termine la triangulation de T en ayant effectué préalablement un saut d'une colonne. Cela permet de vérifier que le rang de T vaut effectivement  $t - 1$ , c'est-à-dire que les représentants des arêtes retenues par l'entier à tester engendrent un hyperplan.

Reste à tester  $\omega$  sur les arêtes de type (b) non encore retenues. Le "produit scalaire" de  $\omega$  avec une arête de type (b) non encore retenue se fera bien sûr via les composantes du représentant de cette arête correspondant aux arêtes de type (a) non retenues. On calcule en fait le produit scalaire de ce représentant et du vecteur normal à F.

Si le "produit scalaire" de  $\omega$  avec une de ces arêtes de type (b) est négatif, l'entier ne caractérise pas une face (toutes les arêtes ne sont pas du même côté de l'hyperplan); si le "produit scalaire" de  $\omega$  avec une de ces arêtes de type (b) est nul, cette arête appartient à F et doit donc être introduite dans l'entier caractérisant la face potentielle.

Si aucune obstruction n'a été rencontrée au cours du calcul, l'entier de départ complété mémorise une face F.

A partir de  $\omega$ , il est facile de reconstituer un vecteur normal à cette face. En général, pourtant, cela n'est pas nécessaire; seule la liste complète des arêtes contenues dans la face nous intéresse.

Par contre, lorsqu'on effectue le dernier saut dans l'algorithme de la cascade, il est judicieux de reconstituer un vecteur normal à la face. En effet, la plupart du temps, ce vecteur est réutilisé ultérieurement par d'autres procédures. Par exemple, ce vecteur est indispensable lorsqu'on veut calculer le domaine voisin.<sup>11</sup>

Reprenons maintenant la question des groupes de symétries  $H^k$ .<sup>12</sup>

<sup>11</sup> Cette remarque explique pourquoi le dernier saut dans l'algorithme de la cascade est effectué par une procédure spéciale (*projeter-naf*), alors que les autres sauts sont réalisés par la procédure *projeter*.

<sup>12</sup> Les deux types d'enregistrement utilisés pour mémoriser un élément du groupe des symétries sont les suivants :

Supposons connu un système de générateurs du groupe des symétries de  $\mathcal{C}$ . Notons  $L^N$  l'ensemble de ces générateurs. Si les arêtes de  $\mathcal{C}$  sont ordonnées, on peut définir  $L^k = \{g \in L^N \mid g \text{ fixe les } k - N \text{ dernières arêtes de } \mathcal{C}\}$  et prendre pour  $H^k$  le groupe engendré par les éléments de  $L^k$ .

Pratiquement, on ordonnera les  $s - N$  dernières arêtes en essayant de maximiser  $|L^k|$ .<sup>13</sup>

Lorsqu'une nouvelle face de  $\mathcal{C}^k$  est connue, on calcule son orbite sous l'action du groupe  $H^k$ .<sup>14</sup> Par la suite, lorsqu'on devra tester un entier représentant une face potentielle, on vérifiera d'abord que cette face potentielle n'est pas contenue dans une des faces d'une orbite déjà connue.<sup>15</sup>

La représentation de l'information sous forme d'entiers s'avère très efficace, d'une part, parce que la densité d'information, rapport à la quantité de mémoire utilisée, est très grande et, d'autre part, parce que les opérations sur les bits des entiers jouent avec la représentation interne et physique des données informatiques. Le traitement comparatif des bits de deux entiers peut se faire à l'aide d'une seule instruction logique simulant ainsi une architecture d'ordinateurs vectoriels.

Une illustration des performances liées à cette représentation est donnée par le problème suivant qui peut se formaliser ainsi :<sup>16</sup>

- type "mat" matrice carrée mémorisant un automorphisme de la forme parfaite  $q$ .

- type "ent" entier mémorisant l'action de cet automorphisme sur l'ensemble des arêtes du domaine. Le premier type est avant tout utilisé pour la saisie ou l'affichage d'un automorphisme.

Le deuxième type d'enregistrement est beaucoup plus efficace, par exemple lorsqu'on doit effectivement calculer l'orbite d'une face. Il s'agit, en fait, de mémoriser une permutation des arêtes du domaine. Comme les domaines considérés ( $n \leq 7$ ) possèdent au plus 63 arêtes, 6 bits suffisent toujours pour mémoriser l'indice d'une arête ( $63 < 2^6 = 64$ ).

La numérotation "naturelle" en LISP commence à 0. Les  $s$  arêtes seront numérotées de 0 à  $s - 1$ . Les blocs successifs de 6 bits de notre entier mémoriseront la permutation des arêtes du domaine de la façon suivante : si l'arête numéro  $i$  a pour image l'arête numéro  $j$ , les 6 bits de notre entier à compter du bit  $6i$  (y compris) mémoriseront  $j$  en base 2. Notre entier possèdera 6s bits significatifs.

Lorsque  $s$  vaut 63 ( $6s = 378$ ), on appréciera les performances du langage LISP, langage qui accepte les entiers arbitrairement grands!

La mémorisation d'une symétrie du domaine sous la forme d'un entier présente, entre autres avantages, celui de pouvoir identifier le groupe des symétries du domaine avec le groupe des symétries de  $\mathcal{C}'$  et le groupe des symétries de  $\mathcal{C}$ . Cette représentation, d'autre part, comme je l'ai déjà mentionné, permet de considérer le groupe des symétries de  $\mathcal{C}^k$  comme un sous-groupe du groupe des symétries de  $\mathcal{C}$ .

<sup>13</sup> La procédure qui définit les  $L^k$ , donc les  $H^k$ , s'appelle *init-groupe*; elle fait appel à la procédure *arranger* pour ordonner les  $s - N$  dernières arêtes.

<sup>14</sup> La procédure qui calcule l'orbite d'une face est la procédure *faces-equivalentes*.

<sup>15</sup> La procédure qui vérifie si la face potentielle n'est pas contenue dans une face déjà connue s'appelle *nouveau-p*. La gestion des faces de  $\mathcal{C}^k$  est effectuée dans l'environnement de *gestion-faces* par des procédures internes de cet environnement.

<sup>16</sup> Dans le programme, ce problème est résolu par la fonction *nouveau-p* (fonction interne de *gestion-faces*).

On se donne un ensemble ordonné  $E$  (ensemble d'arêtes dans notre cas) à  $s$  éléments et une liste  $L = \{P_1, P_2, \dots, P_p\}$  de parties de  $E$  (les faces déjà connues par exemple). Soit  $P$  une partie quelconque de  $E$ , est-elle contenue dans un des éléments de la liste  $L$ ?

Une partie de  $E$  sera représentée par un entier à  $s$  bits significatifs, le  $j^{\text{ème}}$  bit valant 0 si et seulement si le  $j^{\text{ème}}$  élément de  $E$  est dans cette partie. Tester si  $P$  est contenu dans  $P_i$  revient à tester si, à tout bit nul de  $P$ , correspond un bit nul de  $P_i$ .

Considérons le tableau logique suivant :

$j^{\text{ème}}$ bit de $P$	$j^{\text{ème}}$ bit de $P_i$	$j^{\text{ème}}$ bit de l'entier-test
0	0	0
0	1	1
1	0	0
1	1	0

fig. 10

Pour effectuer ce test, on appliquera un "et logique" entre le "complémentaire logique" de  $P$  et l'entier  $P_i$ . Si le résultat vaut 0, cela signifie que tous les bits tests sont nuls et, par conséquent, que  $P$  est contenu dans  $P_i$ .

L'instruction LISP qui effectue ce test d'inclusion est la suivante :

*(zerop (logand1 P P<sub>i</sub>)).*

En remarquant que si  $P$  est contenu dans  $P_i$ , les entiers  $P$  et  $P_i$  satisfont  $P \geq P_i$ , on pourra même parfois éviter le test logique *logand1*, grâce à l'instruction LISP suivante :

*(and ( $\geq$  P P<sub>i</sub>))*

*(zerop (logand1 P P<sub>i</sub>)).*

Lorsque le test d'inclusion est fréquent, le choix de cette deuxième instruction est tout à fait justifié, puisqu'en évitant parfois l'opération logique *logand1*, il décharge une partie de la mémoire de travail interne de LISP et entraîne ainsi une diminution de la fréquence d'appel au "garbage collector".

Lorsque  $L$  possède quelques centaines d'éléments, la structure LISP utilisée pour mémoriser  $L$  sera simplement celle d'une liste, les éléments de cette liste étant les entiers représentant les parties de  $E$  contenues dans  $L$ . Le test d'inclusion de  $P$  se fera par un parcours séquentiel de cette liste : on testera successivement l'inclusion de  $P$  dans chacun des éléments de la liste. Le temps d'accès à l'information dépend plus ou moins linéairement de  $p$ , c'est-à-dire du nombre d'éléments de  $L$ .

Mais lorsque la liste  $L$  possède plusieurs milliers d'éléments, le test séquentiel peut être très long. On adoptera dans ce cas une autre représentation interne de l'information contenue dans  $L$ .

Considérons la  $p \times s$ -matrice de bits ( $p$  = nombre d'éléments de  $L$ ,  $s$  = nombre d'éléments de  $E$ ) dont l'élément à la  $j^{\text{ème}}$  ligne et à la  $j^{\text{ème}}$  colonne est égal au  $j^{\text{ème}}$  bit du  $j^{\text{ème}}$  élément de  $L$ . Les éléments de  $L$  correspondent aux lignes de cette matrice. La deuxième façon de mémoriser  $L$  consiste à travailler avec les  $s$  colonnes de cette matrice. Ainsi,  $L$  sera mémorisée à l'aide d'un vecteur  $\xi$  à  $s$  composantes, chaque composante étant un entier avec  $p$  bits significatifs. Lorsqu'une orbite possède 40000 éléments par exemple ( $p = 40000$ ), les entiers mémorisés ont alors pour ordre de grandeur  $2^{40000}$  dont le développement en base dix avoisine les douze mille chiffres. On appréciera à nouveau le fait que LISP travaille avec des entiers arbitrairement grands!

L'avantage principal de cette deuxième représentation est que le temps d'accès à l'information ne dépend presque plus de  $p$ . Le temps nécessaire pour tester l'inclusion d'une partie  $P$  est pour ainsi dire le même si  $p = 20000$  ou si  $p = 40000$ .

Le test d'inclusion de  $P$  s'effectue en appliquant un "ou inclusif logique" sur les composantes de  $\xi$  correspondant aux bits nuls de  $P$ . Le  $j^{\text{ème}}$  bit de l'entier-test ainsi obtenu vaut 0 si et seulement si  $P$  est contenu dans le  $j^{\text{ème}}$  élément de  $L$ .

La fonction LISP *logcount* compte instantanément le nombre de bits égaux à 1, d'un entier donné; pour savoir si  $P$  est contenu dans un élément de  $L$ , il suffira de comparer le nombre de bits égaux à 1 dans notre entier-test avec le nombre d'éléments de  $L$ .

### 8.3.2. Algorithme de l'explorateur :

Lorsque la taille d'un domaine (différence entre son nombre d'arêtes et sa dimension) est grande,<sup>17</sup> l'algorithme de la cascade n'est plus assez efficace.<sup>18</sup> L'utilisation de l'algorithme de l'explorateur se révèle indispensable.

#### 8.3.2.1. Description :

L'algorithme de l'explorateur est avant tout basé sur la notion de faces voisines, notion développée dans le paragraphe 4.3. Cet algorithme se base sur les corollaires du théorème 7 en utilisant, de plus, le groupe des symétries du domaine, ce qui localise considérablement le problème et, par conséquent, le simplifie.

<sup>17</sup> empiriquement supérieure à 8.

<sup>18</sup> En particulier, les domaines associés à  $E_6$  et à  $E_7$  y résistent.

D'une certaine manière, l'algorithme de l'explorateur est un "algorithme de Voronoi" pour les faces d'un domaine, un "algorithme de Voronoi" en profondeur.

La première phase consiste à récolter les faces du domaine déjà connues. Pratiquement, dans les cas qui nous intéressent, on trouve toujours parmi ces faces au moins une face qui n'est pas maximale. Si tel n'était pas le cas, il faudrait en calculer une; mais il est toujours facile de trouver une face, même non maximale; la difficulté est de s'assurer qu'on les a toutes à équivalence près! L'existence de faces non maximales est garantie par le corollaire 1 du théorème 7.

Parmi les faces récoltées, on en retiendra une par classe d'équivalence rencontrée, c'est-à-dire par orbite sous l'action du groupe des symétries du domaine. Pour chaque face retenue non maximale, on calculera l'ensemble de ses faces voisines. Parmi toutes ces faces voisines, certaines, probablement, seront équivalentes à une face déjà rencontrée, d'autres, peut-être, seront nouvelles. On continuera alors l'exploration à partir de ces nouvelles faces.

Plus précisément, pour chaque nouvelle orbite rencontrée de faces non maximales, on choisira un représentant pour lequel on calculera toutes les faces voisines et ainsi de suite, jusqu'à ce que toutes les faces voisines de toutes les faces connues soient équivalentes à une des faces connues.

La connexité du graphe des faces d'un domaine garantit alors qu'on a un représentant de chaque orbite. Le procédé s'arrête puisque le nombre de faces étant fini, le nombre d'orbites, a fortiori, est aussi fini.

Il faut tout de même faire la remarque évidente suivante : si deux faces sont équivalentes, toute face voisine de la première est équivalente à une face voisine de la seconde.

Le problème est maintenant le suivant : étant donné une face  $F$  d'un domaine, calculer ses faces voisines.

L'intersection de  $F$  avec une de ses faces voisines est une facette de codimension 2 du domaine donc, par le lemme 2, une face de  $F$ .

Inversement, si  $f$  est une face de  $F$ , par le lemme 1,  $f$  est une facette de codimension 2 du domaine. Par le lemme 5, seules exactement deux faces du domaine contiennent  $f$ . La face  $F$  en est une; l'autre est une face voisine de  $F$ ; c'est la face voisine de  $F$ , par rapport à  $f$ .

Par conséquent, il s'agit de calculer les faces de  $F$ .

### 8.3.2.2. Symétries :

Considérons  $H$ , le sous-groupe des symétries du domaine qui stabilise la face  $F$ .

H agit sur les faces de  $F$  et les organise en orbites. On dira que deux faces  $f_1$  et  $f_2$  de  $F$  sont équivalentes si elles appartiennent à la même orbite, c'est-à-dire s'il existe  $h \in H$ , tel que  $f_2 = h \cdot f_1$ .

Notons  $F_i$ , la voisine de  $F$  par rapport à  $f_i$  ( $i = 1, 2$ ). Observons que  $h \cdot F_1$  est une face du domaine qui contient  $f_2$ . Comme  $h$  stabilise  $F$ ,  $h \cdot F_1$  est une face différente de  $F$ . Par unicité de la face voisine de  $F$  par rapport à  $f_2$ ,  $F_2 = h \cdot F_1$ . Comme  $h$  est une symétrie du domaine, on conclut que  $F_1$  et  $F_2$  sont deux faces équivalentes.

En résumé, à deux faces équivalentes de  $F$  correspondent deux faces voisines équivalentes; cette remarque nous permet de prendre  $H$  comme groupe des symétries de  $F$ .

Pour appliquer l'algorithme de l'explorateur, il suffit de calculer les faces de  $F$ , à équivalence près.<sup>19</sup>

La description du calcul des faces de  $F$ , à équivalence près, et des faces voisines correspondantes est un cas particulier de la situation générale décrite dans le paragraphe suivant 8.3.3.

### 8.3.3. Fusion de ces deux algorithmes :

#### 8.3.3.1. Description :

L'algorithme de l'explorateur ramène, d'une certaine manière, le calcul des faces d'un domaine au calcul des faces de ses faces. Une face  $F$  d'un domaine  $D$  est aussi un cône convexe saillant. On désire calculer les faces de ce cône convexe  $F$  modulo l'action d'un groupe  $H$  de symétries.

Si la taille de  $F$  est petite,<sup>20</sup> on appliquera comme précédemment l'algorithme de la cascade mais, cette fois-ci, au cône convexe saillant  $F$ . Si, par contre, la taille de  $F$  est trop grande, on fera appel récursivement à l'algorithme de l'explorateur pour calculer les faces de  $F$ .

L'algorithme de l'explorateur appliqué à  $F$  ramène le calcul des faces de  $F$  au calcul des faces des faces de  $F$  et ainsi de suite : si  $f$  est une face de  $F$ , le calcul des faces de  $f$  sera effectué par l'algorithme de la cascade si la taille de  $f$  est petite, ou par l'algorithme de l'explorateur appliqué récursivement à  $f$ , si la taille de  $f$  est encore trop grande.

L'algorithme de l'explorateur contourne les faces maximales. Ainsi, un appel à cet algorithme diminue toujours, non seulement la dimension, mais également la taille

<sup>19</sup> Pour décider si deux faces d'un même domaine sont équivalentes ou non, sans devoir calculer leurs orbites, on fera appel à la fonction *dans-la-meme-orbite*, fonction qui tente de construire un élément du groupe des symétries du domaine, qui envoie la première face sur la seconde (cf 8.6.3).

<sup>20</sup> empiriquement inférieure ou égale à 8.

des cônes convexes saillants sur lesquels on espère pouvoir appliquer l'algorithme de la cascade, ce qui garantit l'efficacité de l'exploration.<sup>21</sup>

L'utilisation récursive de l'algorithme de l'explorateur crée des chaînes de cônes convexes saillants, où chaque chaînon est une face du chaînon précédent :

$C = C_0 \supset C_1 \supset C_2 \supset \dots \supset C_i$ , où pour  $i \geq 1$ ,  $C_i$  est un cône convexe saillant, face de  $C_{i-1}$  et facette de codimension  $i$  de  $C$ .

Nous appellerons une telle chaîne une chaîne inclusive de faces.

Si  $G = H_0$  est le groupe des symétries de  $C$ , on définit inductivement, pour  $i$  compris entre 1 et  $t$ ,  $H_i$ , le groupe des symétries de  $C_i$  par la condition suivante :  $H_i$  est le sous-groupe de  $H_{i-1}$  qui stabilise  $C_i$ . Cette définition des  $H_i$ , comme au paragraphe 8.3.2.2, garantit qu'à deux faces de  $C_i$ , équivalentes par l'action de  $H_i$ , correspondent deux faces de  $C_{i-1}$ , voisines de  $C_i$ , équivalentes par l'action de  $H_{i-1}$ .

### 8.3.3.2. Implémentation en LISP :

Même si, dans la définition généralisée de l'algorithme de l'explorateur, on fait parfois appel récursivement à cet algorithme, il n'est pas nécessaire d'automatiser cette récursivité.<sup>22</sup>

<sup>21</sup> Généralement, mais cela est un résultat tout à fait empirique, le gain sur la taille est important. Si  $C$  représente le domaine de Voronoï d'une forme parfaite dont le nombre de variables est inférieur ou égal à sept, ou une facette d'un tel domaine, la plus grande taille d'une face non maximale de  $C$  dépasse très rarement les  $\frac{2}{3}$  de la taille de  $C$ .

#### Exemples importants :

- (i)  $C =$  domaine associé à  $E_7$  (63 arêtes);  
une face  $F$  de  $C$  contient au plus 46 arêtes;  
taille  $C = 63 - 28 = 35$     taille  $F = 46 - 27 = 19$ .
- (ii)  $C =$  domaine associé à  $D_7$  (42 arêtes);  
à équivalence près, une seule face  $F$  de ce domaine contient plus de 27 arêtes et elle en contient 36;  
taille  $C = 42 - 28 = 14$     taille  $F = 36 - 27 = 9$ .

<sup>22</sup> Lorsqu'on doit calculer les faces d'un cône convexe  $C_i$  défini par une certaine chaîne, le choix de calculer ces faces à l'aide de l'algorithme de la cascade ou, au contraire, en faisant appel récursivement à l'algorithme de l'explorateur, peut être laissé à la libre appréciation de l'utilisateur. J'ai choisi cette option dans l'implémentation de l'algorithme de l'explorateur afin de garder un contrôle permanent des choix importants et de l'état des calculs.

Cette option est également motivée par certaines contraintes liées à des temps de calculs importants. Lorsque la résolution d'un problème demande plusieurs jours, voire plusieurs semaines à un ordinateur, il ne faut pas être tributaire de certains aléas (maintenance des machines et des logiciels, orages, coupures de courant, etc.) qui peuvent interrompre prématurément l'exécution du programme. Autrement dit, je me suis fixé, a priori, la contrainte suivante : mes programmes doivent pouvoir être stoppés à n'importe quel moment (même brutalement) et ensuite relancés sans qu'aucune erreur ne puisse s'introduire et sans que cette manœuvre ne coûte plus d'une heure de temps de calcul.

Lorsque l'on veut traiter l'élément  $C_1$  d'une chaîne inclusive de faces, c'est-à-dire calculer les faces de  $C_1$  à l'aide de l'algorithme de l'explorateur, il faut commencer par définir cette chaîne inclusive de faces.<sup>23</sup>

Si un domaine  $D$  possède une face  $F$  pour laquelle le domaine voisin est équivalent au domaine associé à  $E_n$ , l'équivalence appliquée à  $F$  nous donne une face du domaine associé à  $E_n$ .<sup>24</sup>

Le théorème 10 et sa démonstration montrent que les faces ainsi récoltées caractérisent des orbites distinctes et que ces orbites contiennent toutes les faces du domaine associé à  $E_n$  correspondant aux voisines de  $E_n$  contenues dans une des autres classes connues de formes parfaites. La cardinalité de ces orbites sera calculée en appliquant le lemme 8.

Dans ce cas, on est sûr, a priori, que l'algorithme de l'explorateur ne permettra de découvrir que des faces pour lesquelles le domaine voisin est équivalent au domaine associé à  $E_n$ , ou éventuellement des faces correspondant à des voisines de  $E_n$  appartenant à de nouvelles classes de formes parfaites.<sup>25</sup>

<sup>23</sup> Cette définition est effectuée, dans le programme, par un appel à la procédure *trajectoire*. Parmi les formes parfaites dont le nombre de variables est inférieur ou égal à sept, seules  $E_6$  et  $E_7$  possèdent des domaines qui requièrent l'utilisation de l'algorithme de l'explorateur. C'est pourquoi, dans l'implémentation du programme, lorsqu'une chaîne inclusive de faces est définie, on suppose que  $C_0$  est le domaine associé à  $E_n$  ( $n = 6$  ou  $7$ ).

Suivant la longueur de la chaîne inclusive de faces, le niveau de profondeur d'application de l'algorithme de l'explorateur change. Pour traiter le domaine associé à  $E_6$ , le premier niveau a été largement suffisant. Par contre, pour traiter le domaine associé à  $E_7$ , il a fallu recourir à l'algorithme de l'explorateur jusqu'au cinquième niveau de profondeur, c'est-à-dire appliquer cet algorithme à des facettes de codimension 4 du domaine associé à  $E_7$ .

<sup>24</sup> Un appel à la procédure *recoller-faces-En* permet de récolter les faces du domaine associé à  $E_n$  déjà connues, en parcourant les fichiers d'orbites de faces des domaines associés aux autres formes parfaites.

<sup>25</sup> Au premier niveau, on applique l'algorithme de l'explorateur au domaine associé à  $E_n$  lui-même. Dans ce cas, l'appel à la procédure *trajectoire* n'est pas nécessaire; formellement, on devrait effectuer (*trajectoire* '(1)).

Le préfixe standard pour les noms des procédures qui agissent à ce niveau est "fa-". Par exemple, la gestion des représentants des orbites de faces du domaine associé à  $E_n$  a lieu dans l'environnement défini par *fa-gestion-liste-a-sauver*, environnement dans lequel ils sont numérotés; l'accès à un tel représentant se fait via l'instruction (*fa-c-s* <ifa>), le paramètre <ifa> étant le numéro de ce représentant; la procédure *fa-enumerer-faces* applique l'algorithme de la cascade à une face du domaine associé à  $E_n$  et calcule donc les faces de cette face; la procédure *fa-toutes-les-voisines* calcule toutes les faces voisines d'une face du domaine associé à  $E_n$ .

Dès le second niveau d'application de l'algorithme de l'explorateur, un appel préalable à la procédure *trajectoire* est indispensable.

Pour le second niveau, l'appel sera de la forme (*trajectoire* '(1 <ifa>)) où <ifa> est le numéro d'une face  $F$  de  $E_n$ . Le préfixe standard pour les noms des procédures qui agissent à ce niveau est "fa-inf-".

Un appel à la procédure *recoller-faces* permettra de récolter les faces de  $F$  déjà connues, en parcourant les listes éventuelles d'orbites de faces des autres faces du domaine associé à  $E_n$ : si une face  $F'$  de ce domaine possède une face  $f'$  dont la face voisine correspondante est équivalente à  $F$ , l'équivalence appliquée à  $f'$  nous donne une face de  $F$ .

A nouveau, le théorème 10 et sa démonstration montrent que les faces ainsi récoltées appartiennent à des orbites de faces de  $F$  distinctes.

Pour appliquer l'algorithme de la cascade à une quelconque facette  $F$  de dimension  $d$  du domaine associé à  $E_n$ , il faut, certes, quelque peu adapter l'implémentation décrite en 8.3.1.3; pourtant, l'adaptation est minime.

La gestion des représentants des orbites de faces de  $F$  a lieu dans l'environnement défini par *fa-inf-gestion-liste-a-sauver*, environnement dans lequel ils sont numérotés; l'accès à un tel représentant se fait via l'instruction (*fa-inf-c-s* <*ifa-inf*>), le paramètre <*ifa-inf*> étant le numéro de ce représentant; la procédure *fa-inf-enumerer-faces* applique l'algorithme de la cascade à une face  $f$  de  $F$  et calcule donc les faces de  $f$ ; la procédure *fa-inf-toutes-les-voisines* calcule toutes les faces voisines d'une face  $f$  de  $F$ .

Lorsque l'algorithme de l'explorateur appliqué à  $F$  s'arrête, c'est-à-dire lorsqu'on connaît, à équivalence près, toutes les faces de  $F$ , un appel à la procédure *fa-enumerer-faces-from-inf* permet de traduire cette information de telle façon qu'on ne puisse plus distinguer si les faces de  $F$  ont été calculées à l'aide de l'algorithme de la cascade, ou à l'aide de l'algorithme de l'explorateur. Cette traduction est indispensable si l'on désire pouvoir utiliser ultérieurement la procédure *fa-toutes-les-voisines* qui, à l'aide des faces de  $F$ , calculera toutes les faces voisines de  $F$ .

A partir du troisième niveau d'application de l'algorithme de l'explorateur, l'appel à la procédure *trajectoire* sera pour le  $(i+1)^{\text{ème}}$  niveau (*trajectoire* '(1  $i_1$   $i_2$ ... $i_i$ )) où  $i_j$  signifie qu'on prend pour  $C_j$  la face numéro  $i_j$  de  $C_{j-1}$ . Le préfixe standard pour les noms des procédures qui agissent à ce niveau est "under-".

Il arrive sporadiquement, lorsqu'on travaille à un certain niveau, qu'on doive accéder à des données ci/ou à des procédures des niveaux supérieurs. Par exemple, lorsqu'on travaille au deuxième niveau de profondeur où le préfixe standard est "fa-inf-", il arrive qu'on fasse appel à la procédure (*fa-c-s...*), procédure qui vit au premier niveau, ou même à (*c-s...*) qui vit dans un certain sens au niveau 0. C'est pourquoi, à partir du troisième niveau de profondeur (préfixe standard "under-"), il est nécessaire d'introduire deux autres préfixes relatifs aux niveaux supérieurs : le préfixe "over-" qui correspond exactement au niveau précédent de profondeur, et le préfixe "super-" qui correspond à deux niveaux précédents de profondeur.

Un appel à la procédure *recoller-faces-over* permettra de récolter les faces de  $C_i$  déjà connues, en parcourant les listes éventuelles d'orbites de faces des autres faces de  $C_{i-1}$  : si une face  $F$  de  $C_{i-1}$  possède une face  $f$  dont la face voisine correspondante est équivalente à  $C_i$ , l'équivalence appliquée à  $f$  nous donne une face de  $C_i$ . Pour les mêmes raisons que précédemment, les faces ainsi récoltées caractérisent des orbites de faces de  $C_i$  distinctes.

La gestion des représentants des orbites de faces de  $C_i$  a lieu dans l'environnement défini par *under-gestion-liste-a-sauver*, environnement dans lequel ils sont numérotés; l'accès à un tel représentant se fait via l'instruction (*under-c-s* <*i-under*>), le paramètre <*i-under*> étant le numéro de ce représentant; la procédure *under-enumerer-faces* applique l'algorithme de la cascade à une face  $f$  de  $C_i$  et calcule donc les faces de  $f$ ; la procédure *under-toutes-les-voisines* calcule toutes les faces voisines d'une face  $f$  de  $C_i$ .

Lorsque l'algorithme de l'explorateur appliqué à  $C_i$  s'arrête, c'est-à-dire lorsqu'on connaît à équivalence près toutes les faces de  $C_i$ , un appel à la procédure *over-enumerer-faces-from-under* permet de traduire cette information de telle façon qu'on ne puisse plus distinguer si les faces de  $C_i$  ont été calculées à l'aide de l'algorithme de la cascade, ou à l'aide de l'algorithme de l'explorateur. Cette traduction est indispensable si l'on désire pouvoir utiliser (lors d'un appel ultérieur à la procédure *trajectoire* au niveau précédent de profondeur) la procédure *under-toutes-les-voisines* qui, à l'aide des faces de la  $i^{\text{ème}}$  face de  $C_{i-1}$ , calculera toutes les faces de  $C_{i-1}$  voisines de cette  $i^{\text{ème}}$  face.

Dans l'environnement défini par *fa-gestion-liste-a-sauver*, les faces du domaine associé à  $E_n$  sont mémorisées dans des structures de type "mat", c'est-à-dire sous forme matricielle. Dans les environnements définis par *fa-inf-gestion-liste-a-sauver* et *under-gestion-liste-a-sauver*, les représentants des orbites de faces ne sont plus des faces du domaine associé à  $E_n$  et sont mémorisées dans une structure de type "ent", en tant que facettes du domaine associé à  $E_n$ .

Tout d'abord, donnons une définition d'un vecteur  $v$  normal à une face  $f$  de  $F$ , qui facilitera cette adaptation, tout en étant suffisante pour les besoins effectifs des calculs. On dira qu'un vecteur  $v$ , dans l'espace de Voronoï, est normal à une face  $f$  de  $F$  si  $v$  est perpendiculaire à toute arête de  $f$  et si le produit scalaire de  $v$  avec un représentant d'une arête de  $F$  qui n'est pas dans  $f$  est toujours positif. La définition naturelle, plus stricte, aurait exigé en plus que  $v$  appartienne au plus petit sous-espace qui contient  $F$  mais, comme nous le verrons plus loin, cette restriction n'est pas utile.

On définit  $C'$  comme en 8.3.1.3, mais en ne retenant que les arêtes du domaine associé à  $E_n$  contenues dans  $F$ . Pour simplifier les notations, on supposera que  $F$  contient exactement  $s$  arêtes;  $s$  est donc inférieur au nombre d'arêtes du domaine associé à  $E_n$ . Le calcul des faces de  $F$  est ramené au calcul des faces de  $C'$ . Après renumérotation éventuelle, on peut supposer que les représentants des  $d$  premières arêtes de  $C'$  sont linéairement indépendants, puisque  $F$  est de dimension  $d$ .

On construit, comme précédemment, la  $N \times s$ -matrice  $\mathfrak{M}$  qui contient toute l'information du problème. Par des opérations élémentaires de lignes uniquement, on peut transformer  $\mathfrak{M}$  en une matrice dont le bloc  $d \times d$ , en haut à gauche, est une matrice identité et dont les  $(N - d)$  dernières lignes ne contiennent que des 0. Donc, il existe une matrice  $B$  inversible telle que :

$$B\mathfrak{M} = \begin{pmatrix} 1 & 0 & \cdots & 0 & * & \cdots & * \\ 0 & 1 & \cdots & 0 & * & \cdots & * \\ \vdots & & & & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 & * & \cdots & * \end{pmatrix} \begin{matrix} d \text{ premières lignes} \\ \\ \\ \end{matrix}$$

$$\begin{pmatrix} 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \end{pmatrix} \begin{matrix} N-d \text{ dernières lignes} \\ \\ \\ \end{matrix}$$

On définit  $C$ , comme tout à l'heure, à l'aide des colonnes de  $B\mathfrak{M}$ , ce qui ramène, à nouveau, notre problème de départ au calcul des faces de  $C$ . D'autre part, si  $v = (v_1, v_2, \dots, v_N)$  représente un vecteur normal à une face de  $C$ , on vérifie sans peine que  $vB$  nous donne un vecteur normal à la face correspondante de  $C'$ , vecteur qui se traduira instantanément en un vecteur normal (au sens où nous l'avons défini dans ce paragraphe) à la face correspondante de  $F$ . Les  $N - d$  dernières composantes de  $v$  n'ont aucune importance; on peut, par exemple, les fixer à 0, ce qui nous autorise à ne considérer que les  $d$  premières lignes de la matrice  $B$  lors du calcul de  $vB$ .

La suite se déroule exactement comme en 8.3.1.3 pour peu qu'on remplace  $N$  par  $d$  et qu'on efface les  $(N - d)$  dernières lignes de  $B\mathbb{N}$ , celles qui ne contiennent que des 0.<sup>26</sup>

Lorsque deux facettes du domaine associé à  $E_n$ ,  $f$  et  $f'$ , sont équivalentes en tant que facettes de ce domaine, l'équivalence envoie les faces de la première sur les faces de la seconde. La connaissance des faces de  $f$  évite alors le calcul des faces de  $f'$  à l'aide d'un des deux algorithmes généraux.<sup>27</sup>

Etant donné que le groupe des symétries de  $f$  peut dépendre du choix de la chaîne qui conduit jusqu'à  $f$ , même si  $f$  et  $f'$  sont équivalentes, leurs groupes de symétries ne sont pas toujours comparables. En particulier, deux faces de  $f$ , équivalentes par l'action du groupe des symétries de  $f$ , ne correspondent pas toujours à deux faces de  $f'$ , équivalentes par l'action du groupe des symétries de  $f'$ .

Pour calculer les faces de  $f'$  à l'aide de celles de  $f$ , on traduira donc toutes les faces de  $f$  (indépendamment de leur appartenance ou non à une même orbite) et on recalculera les orbites de faces de  $f'$  en appliquant le groupe des symétries de  $f'$ .<sup>28</sup>

<sup>26</sup> Lorsqu'on applique l'algorithme de la cascade à une facette  $F$  du domaine associé à  $E_n$ , bien que  $F$  soit mémorisée dans une structure de type "ent" en tant que facette du domaine associé à  $E_n$ , les faces de  $F$  (resp. des cônes obtenus en relevant  $F$ ) sont mémorisées dans des structures de type "ent" en tant que faces de  $F$  (resp. faces des cônes obtenus en relevant  $F$ ), c'est-à-dire comme sous-ensembles des arêtes de  $F$  (resp. des cônes obtenus en relevant  $F$ ). Ce codage dépend de la manière dont sont ordonnées les arêtes de  $F$ . Les paires de vecteurs minimaux de  $E_n$  correspondant à des arêtes de  $F$  sont mémorisées dans une liste, comme propriété de  $F$ , sous le nom de *minimaux*. Leur ordre dans cette liste définit la numérotation des arêtes de  $F$ .

Le codage d'une facette  $f$  du domaine associé à  $E_n$  en une facette  $f'$  d'une facette  $F$  de ce domaine est effectué par les fonctions *coder-ifa*, *coder-over*, *coder-under*, suivant la codimension de la facette  $F$ .

La traduction inverse d'une facette  $f'$  d'une facette  $F$  du domaine associé à  $E_n$  en une facette  $f$  du domaine associé à  $E_n$  s'effectue en utilisant l'information contenue dans un entier mémorisé comme propriété de  $F$  sous le nom de *conversion*.

Le nombre de bits significatifs de cet entier vaut exactement six fois le nombre d'arêtes de  $F$ . Les six bits à compter du bit 6j (y compris) mémorisent le numéro de la  $j^{\text{ème}}$  arête de  $F$  vue comme une arête du domaine associé à  $E_n$ .

Lorsqu'on trouve une nouvelle face  $f$  d'une facette  $F$  du domaine associé à  $E_n$ , on mémorisera son orbite par l'action du groupe des symétries de  $F$ , comme propriété de  $f$ , sous le nom de *orbite*. Cela permettra ultérieurement de tester si une autre face de  $F$  est nouvelle ou si, au contraire, elle est équivalente à une face de  $F$  déjà rencontrée. Les éléments d'une orbite de face de  $F$  seront mémorisés dans des structures de type "ent", en tant que facettes de  $F$  et non pas en tant que facettes du domaine associé à  $E_n$ , ce qui économise de la mémoire : les entiers qui codent ces facettes possèdent ainsi un nombre plus petit de bits significatifs. Bien que ces divers codages exigent certaines conversions, l'économie de mémoire qui s'ensuit justifie pleinement leur utilisation.

De même, les éléments du groupe des symétries d'une facette  $F$  seront mémorisés dans des structures de type "ent", via leur action sur les arêtes de  $F$  et non pas sur les arêtes du domaine associé à  $E_n$ : le calcul des orbites des faces de  $F$  ne s'en trouvera que facilité.

<sup>27</sup> Le calcul des faces d'une facette  $f$  du domaine associé à  $E_n$  peut prendre plusieurs heures lorsque la taille de cette facette est grande, voire même plusieurs jours si la taille de cette facette exige que le calcul de ses faces s'effectue à l'aide de l'algorithme de l'explorateur.

<sup>28</sup> Il peut arriver que deux facettes  $f$  et  $f'$  d'une facette  $F$  du domaine associé à  $E_n$  soient équivalentes par l'action du groupe des symétries du domaine associé à  $E_n$  bien qu'elles ne soient pas équivalentes par l'action du groupe des symétries de  $F$ . C'est pourquoi le préfixe "pseudo-" est employé pour ca-

Afin de pouvoir être éventuellement réutilisées ultérieurement, les faces de  $f$  doivent être mémorisées.<sup>29</sup>

L'exemple suivant est un cas particulier important à ne pas oublier. Lorsqu'une face  $f$  de  $F$  est maximale, il n'est pas nécessaire de calculer ses faces pour appliquer l'algorithme de l'explorateur à  $F$ . Mais lorsque toutes les faces de  $F$  sont connues, il est trivial, a posteriori, de calculer les faces de  $f$  : ce sont les intersections de  $f$  avec les autres faces de  $F$ . Si la taille de  $f$  est grande, on effectuera ce calcul afin de retenir les faces de  $f$ . En effet, il peut arriver qu'une face  $f$  maximale de  $F$  soit équivalente, par l'action du groupe des symétries du domaine associé à  $E_n$ , à une face  $f'$  d'une facette  $F'$ , sans pour autant que  $f$  soit maximale dans  $F$ .

Reste à décrire l'algorithme qui calcule les faces voisines dans le cadre de l'algorithme de l'explorateur.<sup>30</sup>

Soit un cône  $D$ , une face  $F$  de  $D$  et une face  $f$  de  $F$ . Le but est de calculer et de caractériser  $F'$ , l'unique face de  $D$  voisine de  $F$  par rapport à  $f$ . On supposera que  $D$  est une facette d'un domaine  $C$ , ou éventuellement  $C$  lui-même, et que  $C$  vit dans  $\mathbb{R}^N$ .<sup>31</sup> On a donc :  $N = \dim C \geq \dim D = (\dim F) + 1 = (\dim F' + 1) = (\dim f) + 2$ .

Supposons connus  $n_F$  et  $n_f$  des vecteurs normaux respectivement à  $F$  et à  $f$ , au sens défini dans ce paragraphe; ces vecteurs  $n_F$  et  $n_f$  satisfont les conditions ci-dessous, où  $a$  caractérise une arête quelconque de  $D$  :

$$\begin{aligned} n_F \cdot a &= 0 & \text{si } a \in F \\ n_F \cdot a &> 0 & \text{si } a \in D, a \notin F \\ n_f \cdot a &= 0 & \text{si } a \in f \\ n_f \cdot a &> 0 & \text{si } a \in F, a \notin f. \end{aligned}$$

Le produit scalaire utilisé est le produit scalaire usuel de  $\mathbb{R}^N$ .

Je rappelle que  $n_F$  et  $n_f$  n'appartiennent pas forcément à  $L(D)$ , le plus petit sous-espace de  $\mathbb{R}^N$  qui contient  $D$ . Le produit scalaire de  $n_F$  (resp.  $n_f$ ) avec les vecteurs caractérisant des arêtes de  $C$  qui ne sont pas dans  $D$  (resp. dans  $F$ ) peut être quelconque (négatif, positif ou nul); cela n'a pas d'importance.

On cherche à calculer  $F'$  et  $n_{F'}$  un vecteur normal à  $F'$  satisfaisant :

$$n_{F'} \cdot a = 0 \text{ si } a \in F' \text{ et } n_{F'} \cdot a > 0 \text{ si } a \in D, a \notin F'.$$

caractériser les procédures qui cherchent ou utilisent cette équivalence par l'action du groupe des symétries du domaine associé à  $E_n$ .

<sup>29</sup> Pour ne pas submerger le disque dur de l'ordinateur (!), on ne retiendra les faces de  $f$  que si la taille de  $f$  est suffisamment grande (en fait supérieure à cinq). La gestion des facettes retenues, ainsi que de leurs faces, est essentiellement effectuée dans le programme par les procédures dont le préfixe est "g-d-" (signifiant "grand-delta", c'est-à-dire grande différence entre le nombre d'arêtes et la dimension).

<sup>30</sup> Dans le programme, la procédure qui effectue ce calcul s'appelle *chercher-faces-voisines*.

<sup>31</sup>  $D$ ,  $F$ ,  $f$  et  $F'$  seront mémorisées dans des structures de type "cni", en tant que facettes de  $C$ .

A nouveau,  $n_F$  n'a pas besoin d'appartenir à  $L(D)$ .

Décomposons  $\mathbb{R}^N$  en somme directe de  $L(D)$  et de son complémentaire orthogonal  $L(D)^\perp$ . Tout vecteur  $n$  peut être décomposé comme somme de  $n^*$  et de  $n^\perp$  où  $n^*$  est un vecteur de  $L(D)$  et  $n^\perp$  un vecteur de  $L(D)^\perp$ .

Représentons la situation dans un 2-plan de  $L(D)$  orthogonal à  $f$ .

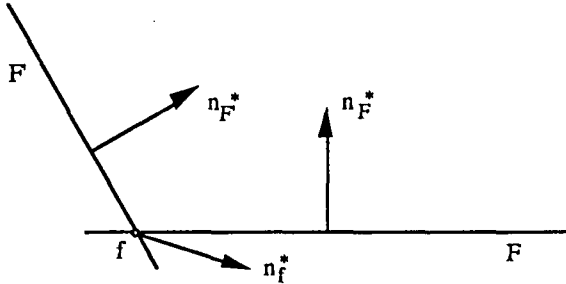


fig. 11

Les vecteurs  $n_F^*$  et  $n_f^*$  ne sont pas parallèles car, si  $a$  caractérise une arête de  $F$  qui n'appartient pas à  $f$ , on observe que  $n_f^* \cdot a > 0$ , alors que  $n_F^* \cdot a = 0$ . Par conséquent, le vecteur  $n_F^*$  peut s'écrire comme combinaison linéaire de  $n_f^*$  et  $n_F^*$ . En observant de plus que le vecteur  $a$  ci-dessus satisfait  $n_F^* \cdot a > 0$ , on déduit que le coefficient devant  $n_f^*$ , dans cette combinaison linéaire, est positif; après normalisation éventuelle, on peut supposer que ce coefficient vaut 1.

Il existe donc un nombre  $\beta$ , tel que  $n_F^* = n_f^* + \beta n_F^*$ . Il est clair que le produit scalaire entre  $n_F^*$  et un vecteur caractérisant une arête quelconque de  $f$  est nul, quelle que soit la valeur de  $\beta$ . D'autre part,  $F$  et  $F'$  n'ont comme arêtes communes que celles qui sont dans  $f$ .

Regardons les arêtes de  $D$  qui ne sont pas dans  $F$ .  $F'$  en contient au moins une. Soit  $a'$ , un vecteur caractérisant une telle arête. Ce vecteur satisfait  $n_F^* \cdot a' = 0$ . D'autre part, le produit scalaire entre  $n_F^*$  et un vecteur caractérisant une arête quelconque de  $D$  qui n'est pas dans  $F$  doit toujours être supérieur ou égal à zéro. On tire de ces deux conditions :

$$\beta = \frac{-n_f^* \cdot a'}{n_F^* \cdot a'} = \frac{-n_f^* \cdot a'}{n_f^* \cdot a'} \quad \text{et} \quad \beta \geq \frac{-n_f^* \cdot a}{n_F^* \cdot a} = \frac{-n_f^* \cdot a}{n_f^* \cdot a} \quad \text{quel que soit le vecteur } a \text{ caractérisant}$$

une arête de  $D$  qui n'est pas dans  $F$ .

On pose donc  $\beta = \max \frac{-n_f^* \cdot a}{n_f^* \cdot a}$  où le vecteur  $a$  caractérise à tour de rôle chacune des arêtes de  $D$  qui n'est pas dans  $F$ .

L'ensemble des arêtes de  $F$  est réunion de deux ensembles : l'ensemble des arêtes de  $f$  et l'ensemble des arêtes de  $D$  qui n'appartiennent pas à  $F$ , caractérisées par un vecteur  $a$  vérifiant  $\frac{n_f \cdot a}{n_f \cdot a} = \beta$ . Pour terminer, on définit  $n_F$  par l'égalité suivante :  
 $n_F = n_f + \beta n_F$ .

#### 8.4. Calcul automatique des voisins :

##### 8.4.1. Description :

Reprenons les notations du chapitre 5.

Nous supposons données la matrice  $Q$  d'une forme parfaite et  $l$ , une matrice mémorisant un vecteur normal à une face  $F$  du domaine associé à  $Q$ . On cherche à déterminer  $Q'$  la voisine de  $Q$  relativement à  $F$ .

Nous avons vu qu'il existe une valeur de  $\rho$  positive telle que  $Q' = Q + \rho l$ , que le minimum de  $Q$  est égal au minimum de  $Q + \theta l$  si et seulement si  $\theta$  appartient à l'intervalle  $[0, \rho]$  et que la forme  $Q'$  possède au moins un vecteur minimal  $v \in \mathbb{Z}^n$  qui n'est pas minimal pour  $Q$ .

En contemplant l'égalité  $v^l Q(\rho) v = v^l Q v + \rho v^l l v = \min Q$  et en se rappelant que  $v^l Q v > \min Q$ , on déduit que  $v^l l v$  est négatif.

Par conséquent, si  $\theta \geq \frac{v^l Q v}{-v^l l v}$ , la forme  $Q(\theta)$  n'est plus définie positive.

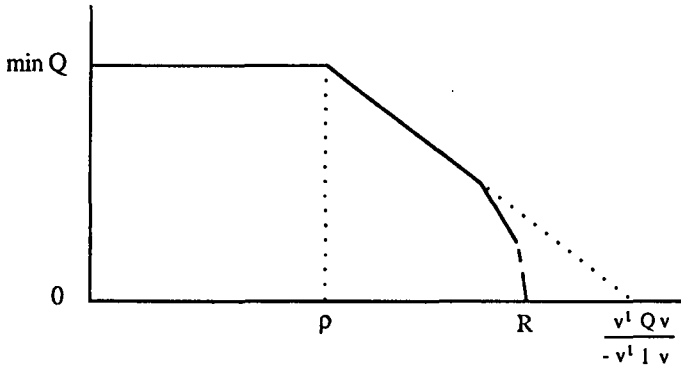


fig. 12

Il est facile de voir que si  $Q(\theta')$  est définie positive pour une certaine valeur positive de  $\theta'$ ,  $Q(\theta)$  est définie positive pour toute valeur de  $\theta$  appartenant à l'intervalle  $[0, \theta']$ . Il existe donc une valeur limite  $R$  pour  $\theta$ , telle que  $Q(\theta)$  soit définie positive si  $\theta$  appartient à l'intervalle  $[0, R[$ , et indéfinie si  $\theta$  est supérieur à  $R$ .

Le graphe de  $\min Q(\theta)$  sur l'intervalle  $[0, R[$  est une courbe continue, concave, composée de segments (fig. 12).

#### 8.4.2. Implémentation en LISP :

L'algorithme qui décrit ce calcul se décompose en deux parties.<sup>32</sup> La première partie consiste à trouver une valeur de  $\theta$  strictement comprise entre  $\rho$  et  $R$ , c'est-à-dire pour laquelle  $Q(\theta)$  est définie positive et  $\min Q(\theta) < \min Q$ .<sup>33</sup> La seconde partie utilise cette valeur de  $\theta$  pour déterminer la valeur exacte de  $\rho$ .<sup>34</sup>

<sup>32</sup> La procédure permettant le calcul automatique de  $Q'$  s'appelle *voisine*.

<sup>33</sup> Première partie :

Le programme fait d'abord appel à la fonction *grand-R* qui rend une valeur de  $\theta$  supérieure à  $\rho$ . La fonction *grand-R* construit, en fait, une suite  $\theta_1, \theta_2, \dots$  satisfaisant  $\theta_1 = 1$  et  $\theta_i = 3 \theta_{i-1}$ , et s'arrête dès que  $Q(\theta_i)$  n'est plus définie positive, ou que le minimum de  $Q(\theta_i)$  est strictement inférieur au minimum de  $Q$ .

Ensuite, la fonction *osciller* fournit une valeur de  $\theta$  strictement comprise entre  $\rho$  et  $R$ . La fonction *osciller* construit récursivement une suite d'intervalles emboîtés de la forme  $[a_i, b_i]$  avec  $a_1 = 0$  et  $b_1 =$  valeur rendue par la fonction *grand-R*. Elle arrête sa recherche dès que  $Q(\frac{a_i + b_i}{2})$  est définie positive et de minimum strictement inférieur au minimum de  $Q$ .

Le principe est le suivant :

- si  $Q(\frac{a_i + b_i}{2})$  n'est pas définie positive, on continue la recherche avec  $a_{i+1} = a_i$  et  $b_{i+1} = \frac{a_i + b_i}{2}$  ;
- sinon,
  - si le minimum de  $Q(\frac{a_i + b_i}{2})$  est égal au minimum de  $Q$ , on continue la recherche avec  $a_{i+1} = \frac{a_i + b_i}{2}$  et  $b_{i+1} = b_i$  ;
  - sinon, on arrête la recherche; la valeur  $\frac{a_i + b_i}{2}$  est une solution de notre problème.

<sup>34</sup> Seconde partie :

On suppose connue une valeur de  $\theta$  strictement comprise entre  $\rho$  et  $R$ . Il existe alors au moins un vecteur  $v \in \mathbb{Z}^n$  satisfaisant  $v^t Q(\theta) v < \min Q$ . Ce vecteur  $v$  ne peut pas être un vecteur minimal de  $Q$ . On calcule une nouvelle valeur pour  $\theta$  de telle manière que  $v^t Q(\theta) v$  soit égal au minimum de  $Q$  : on remplace  $\theta$  par  $\frac{\min Q - v^t Q v}{v^t 1 v}$ .

Pour cette nouvelle valeur de  $\theta$ , si  $\min Q(\theta) < \min Q$ , on recommence; sinon  $\theta$  est égal à la valeur de  $\rho$ , et  $v$  est un vecteur minimal de  $Q'$  qui n'est pas minimal pour  $Q$ .

La procédure récursive *minimiser* effectue le travail décrit dans cette deuxième partie. Elle commence, en fait, par chercher les paires de vecteurs  $\pm v \in \mathbb{Z}^n$  satisfaisant  $v^t Q(\theta) v \leq \min Q$ . Si elle trouve un vecteur pour lequel cette inégalité est stricte, elle poursuit sa recherche récursivement, après avoir calculé la nouvelle valeur de  $\theta$ .

### 8.5. Identification automatique des voisines :

Géométriquement, en terme de réseaux, il s'agit, en fait, de résoudre automatiquement (et si possible rapidement!) le problème suivant :

Soit deux réseaux de  $\mathbb{R}^n$  donnés par les matrices de Gram de deux de leurs bases respectives; ces deux réseaux sont-ils équivalents ?

Dans ce qui suit, on utilisera plutôt le langage des formes quadratiques.

#### 8.5.1. Description :

Soit deux formes quadratiques définies positives données par les matrices  $A$  et  $B$  qui leur sont associées. Ces deux formes sont équivalentes si et seulement si il existe un nombre réel positif  $\lambda$  et une matrice  $S$  dans  $GL_n(\mathbb{Z})$ , tels que  $A = \lambda S^t B S$ .

Après normalisation éventuelle, on peut supposer  $A$  et  $B$  de même minimum et de même déterminant. Notre problème consiste à exhiber, si elle existe, une matrice  $S$  à coefficients entiers, telle que  $A = S^t B S$ .

Considérons  $v_1, v_2, \dots, v_p$  ( $v_i \in \mathbb{Z}^n$ ), un quelconque système de générateurs de  $\mathbb{Z}^n$ . Nous allons voir que  $S$  existe si et seulement si on peut trouver  $w_1, w_2, \dots, w_p$  ( $w_i \in \mathbb{Z}^n$ ) tels que  $v_i^t A v_j = w_i^t B w_j$  ( $\forall i, j$ ).

Si  $S$  existe, il suffit de poser  $w_i = S v_i$ ,  $i = 1, 2, \dots, p$ .

Pour montrer la réciproque, considérons  $V$  (resp.  $W$ ), la matrice dont les colonnes sont les  $v_i$  (resp. les  $w_i$ ).

L'hypothèse devient  $V^t A V = W^t B W$ .

Dire que les  $v_i$  forment un système de générateurs de  $\mathbb{Z}^n$  revient à dire qu'il existe une matrice  $U$  à  $p$  lignes et  $n$  colonnes, à coefficients entiers, telle que le produit  $V U$  soit égal à la matrice identité.

Posons  $S = W U$ .  $S$  est à coefficients entiers et vérifie  $A = S^t B S$ .

D'un point de vue algorithmique, étant donné  $v_1, v_2, \dots, v_p$ , un système bien choisi de générateurs de  $\mathbb{Z}^n$ , la recherche des  $w_1, w_2, \dots, w_p$  sera une recherche en arbre avec une très forte heuristique, avant tout basée sur la notion de spectre.

#### Notion de spectre :

Soit  $A$ , la matrice d'une forme quadratique définie positive. Si  $\pm x, \pm y$  sont deux paires de vecteurs à coefficients entiers, on définit leur spectre relatif par :

$$\langle \pm x, \pm y \rangle_A = \frac{1}{\min A} |x^t A y|.$$

Soit  $E$ , un ensemble fini, non-vide, de paires  $\pm y$  de vecteurs à coefficients entiers; on définit le spectre de  $\pm x$  par rapport à  $E$  par :

$\text{spectre}_A(\pm x, E) = ((a_1 n_1) (a_2 n_2) \dots (a_k n_k))$  où la liste  $((a_1 n_1) (a_2 n_2) \dots (a_k n_k))$  satisfait les conditions suivantes :

- pour toute paire  $\pm y$  dans  $E$ , il existe  $a_i$ , dans cette liste, tel que  $a_i$  soit égal à  $\langle \pm x, \pm y \rangle_A$ ;
- pour tout  $i$ ,  $n_i$  est positif et compte le nombre de paires  $\pm y$  dans  $E$  satisfaisant  $\langle \pm x, \pm y \rangle_A = a_i$ ;
- la suite des  $a_i$  est strictement croissante.

Si  $S$  appartient à  $GL_n(\mathbb{Z})$ ,  $S(E)$  désignera l'ensemble des paires  $\pm Sy$  où  $\pm y$  appartient à  $E$ . Pour  $r \geq 1$ , on notera  $\mathfrak{B}(A, r)$ , l'ensemble des paires  $\pm y$  satisfaisant  $\langle \pm y, \pm y \rangle_A \leq r$ ;  $\mathfrak{B}(A, r)$  est toujours un ensemble fini, non-vidé.

Considérons maintenant  $A$  et  $B$ , deux matrices associées à des formes quadratiques définies positives. S'il existe  $S \in GL_n(\mathbb{Z})$ , telle que  $A = S^t B S$ , on a l'identité spectrale suivante :  $\langle \pm x, \pm y \rangle_A = \langle \pm Sx, \pm Sy \rangle_B$ . On en déduit que  $\text{spectre}_A(\pm x, E) = \text{spectre}_B(\pm Sx, S(E)) \quad \forall x, \forall E$ , et que  $S(\mathfrak{B}(A, r)) = \mathfrak{B}(B, r) \quad \forall r \geq 1$ .

Revenons au problème du calcul des  $w_i$ , les  $v_i$  étant connus.

Considérons  $\mathfrak{B}(A, r)$  contenant tous les  $\pm v_i$ , avec  $r$  minimal pour cette propriété. Par ce qui précède, on voit que les  $\pm w_i$  doivent appartenir à  $\mathfrak{B}(B, r)$ , ce qui limite déjà considérablement les choix possibles pour  $w_i$ .

Mieux; comme  $\text{spectre}_A(\pm v_i, \mathfrak{B}(A, r)) = \text{spectre}_B(\pm w_i, \mathfrak{B}(B, r))$ , on voit que l'ensemble des images possibles, a priori, pour  $v_i$  est réduit à l'ensemble des  $x$  tels que  $\pm x \in \mathfrak{B}(B, r)$  et  $\text{spectre}_B(\pm x, \mathfrak{B}(B, r)) = \text{spectre}_A(\pm v_i, \mathfrak{B}(A, r))$ .

La valeur de  $r$  dépend des  $v_i$ . Dans la pratique, il est très avantageux de choisir les  $v_i$  de façon à minimiser  $r$ .

La situation particulière qui nous intéresse est l'identification des formes parfaites en dimension inférieure ou égale à sept. Dans ces dimensions, quelle que soit la forme parfaite, on peut toujours extraire de ses vecteurs minimaux une base de  $\mathbb{Z}^n$ . Cette propriété facilite considérablement les calculs.

Cela montre, en particulier, qu'on peut prendre pour  $r$  la valeur minimale, c'est-à-dire 1. L'heuristique sera donc basée sur le spectre des vecteurs de cette base par rapport à  $\mathfrak{B}(A, 1)$ , c'est-à-dire par rapport à l'ensemble des vecteurs minimaux de  $A$ .

Par abus de langage, on appellera spectre d'un vecteur minimal, son spectre par rapport aux autres vecteurs minimaux de la forme, spectre calculé à l'aide de cette forme justement.

L'heuristique se traduit ici de la manière suivante :  $w_i$  doit être un vecteur minimal de  $B$  dont le spectre est égal au spectre de  $v_i$ .

### 8.5.2. Implémentation en LISP :

Les bases retenues pour les formes parfaites en dimension inférieure ou égale à sept ont été choisies en fonction du représentant mémorisé pour chacune des classes.<sup>35</sup>

Lorsqu'on désire identifier une forme parfaite B avec l'un des représentants des classes déjà connues, on cherche d'abord le numéro probable de la classe de B en calculant certains invariants : valeur du déterminant lorsque le minimum vaut 2 (normalisation), nombre de vecteurs minimaux, ensemble des spectres des vecteurs minimaux.<sup>36</sup> Ensuite seulement, on essaie d'identifier B avec le représentant A de cette classe d'équivalence à l'aide d'une recherche en arbre avec forte heuristique.<sup>37</sup>

Soit  $v_1, v_2, \dots, v_n$  une base de  $\mathbb{Z}^n$  formée de vecteurs minimaux de A. On tente en fait de découvrir un n-tuple  $(w_1, w_2, \dots, w_n)$  de vecteurs minimaux de B dont la matrice de Gram par rapport à B est égale à la matrice de Gram des  $v_i$  par rapport à A.

La liste mémorisée l-m des vecteurs minimaux de B ne contient qu'un représentant par paire de vecteurs opposés. Un vecteur  $w_i$  sera représenté par deux éléments :

- l'indice de son représentant dans l-m,
- le signe de  $w_i$  par rapport à ce représentant (+1 signifie que  $w_i$  est égal à ce représentant; -1 signifie que  $w_i$  est l'opposé de ce représentant).

On commence par calculer pour chaque  $v_i$ , la liste  $l_i$  des indices des vecteurs de l-m dont le spectre est égal au spectre de  $v_i$ . L'indice du représentant de  $w_i$  devra appartenir à  $l_i$ . Ce sont les listes  $l_i$  qui contiennent l'heuristique.

On peut toujours supposer, sans perdre de généralité, que le signe de  $w_1$  est +1 : S est solution de notre problème si et seulement si -S est solution de notre problème.

Chaque sommet de l'arbre de recherche mémorise un n-tuple  $(w_1, w_2, \dots, w_n)$  partiellement rempli.

La gestion des sommets de l'arbre de recherche se fait à l'aide d'une pile de type L.I.F.O. (Last In, First Out). Au départ, on introduit dans la pile, en parcourant la liste d'indices  $l_1$ , les sommets correspondant aux différentes images possibles pour le premier vecteur de base. Ensuite, la boucle principale de recherche est la suivante :

<sup>35</sup> Ces bases sont de trois types :

- type 1      base :  $c_1, c_2, \dots, c_n$   
                   $S = W = (w_1, w_2, \dots, w_n)$
- type 2      base :  $c_1, c_2, \dots, c_{n-1}, -c_1 + c_n$   
                   $S = (w_1, w_2, \dots, w_1 + w_n)$
- type 3      base :  $c_1, c_2, \dots, c_{n-1}, -c_1 - c_2 + c_n$   
                   $S = (w_1, w_2, \dots, w_1 + w_2 + w_n)$

On pourra donc ainsi facilement calculer S à partir des  $w_i$ . La procédure qui effectue ce calcul s'appelle *mat-chgt-base-type*.

<sup>36</sup> La fonction qui, dans le programme, effectue cette recherche s'appelle *candidat-equi*.

<sup>37</sup> Cette deuxième recherche est effectuée par la fonction *chercher-equivalence*.

- si la pile est vide, la recherche a échoué;  
sinon,
  - on extrait le sommet en tête de pile;
  - si ce sommet correspond à un n-tuple de  $w_i$  complètement rempli, la recherche a abouti;  
sinon,
    - on calcule tous les sommets successeurs de ce sommet et on les introduit en tête de pile.

Reste à décrire le calcul des successeurs d'un sommet.

Supposons que ce sommet corresponde à un n-tuple de  $w_i$ , partiellement rempli, c'est-à-dire où sont mémorisées les valeurs associées aux  $(k-1)$  premiers  $w_i$ .

Il s'agit de calculer les valeurs possibles pour  $w_k$ . Les candidats sont construits en parcourant la liste d'indices  $l_k$ . Pour chaque élément de  $l_k$ , on engendre deux valeurs pour  $w_k$ , celle correspondant au signe positif et celle correspondant au signe négatif. Pour chacune de ces valeurs, on compare la matrices de Gram de  $w_1, \dots, w_k$  et celle de  $v_1, \dots, v_k$ . Plus précisément, si  $v_i^t A v_k = w_i^t B w_k$  pour tout  $i$  compris entre 1 et  $k-1$ , cette valeur de  $w_k$  définit un sommet successeur.

Ce dernier test, couplé avec l'heuristique, permet d'éviter en grande partie le "backtracking" intrinsèquement lié à une recherche en arbre, d'où la performance du programme.<sup>38</sup>

## **8.6. Variantes de l'algorithme décrit au paragraphe 8.5 :**

### **8.6.1. Calcul automatique du groupe des automorphismes :**

#### **8.6.1.1. Description :**

Un élément du groupe des automorphismes de  $A$  est, d'une certaine manière, une matrice  $S$  qui identifie  $A$  avec elle-même :  $A = S^t A S$ .

Pour connaître le groupe des automorphismes de  $A$ , il est suffisant de le connaître modulo  $\pm$ identité. Pour calculer ce groupe, il suffit de modifier la boucle principale de l'algorithme décrit en 8.5 dans le sens suivant :

- si la pile est vide, la recherche est terminée; tous les éléments du groupe des automorphismes de  $A$  sont connus modulo  $\pm$ identité;

<sup>38</sup> Il suffit, en général, d'une dizaine de secondes environ pour exhiber  $S$ , si elle existe, et d'une trentaine de secondes pour démontrer, dans le cas contraire, que  $S$  n'existe pas.

sinon,

- on extrait le sommet en tête de pile;
- si ce sommet correspond à un  $n$ -tuple de  $w_i$  complètement rempli, il caractérise un élément du groupe cherché; on retient cet élément et on poursuit la recherche;

sinon,

- on calcule tous les sommets successeurs de ce sommet et on les introduit en tête de pile.

### 8.6.1.2. Implémentation en LISP :

Le groupe des automorphismes d'une forme parfaite peut être très gros. On ne retiendra donc en général qu'un système de générateurs de ce groupe.<sup>39</sup>

### 8.6.2. Calcul automatique du stabilisateur d'une face :

Le stabilisateur d'une face du domaine associé à une forme parfaite  $A$  est un sous-groupe du groupe des symétries de ce domaine, donc un sous-groupe du groupe des automorphismes de  $A$  modulo  $\pm$ identité.

Supposons que la face considérée est donnée par une matrice  $F$ . On peut reprendre tel quel l'algorithme de 8.6.1 en testant de plus, avant de le retenir définitivement, chaque élément  $S$  qui vient d'être calculé, afin de vérifier que  $S^t F S = F$ .

Remarquons que, dans cette situation, on peut affiner l'heuristique mémorisée dans les listes d'indices  $l_j$ .

Notons  $E$ , l'ensemble des paires de vecteurs minimaux de  $A$  correspondant à des arêtes contenues dans  $F$ . Dans l'algorithme général,  $j$  appartient à  $l_j$  si et seulement si le spectre du  $j^{\text{ème}}$  vecteur de  $l-m$  (liste mémorisée des vecteurs minimaux de  $A$ ) est égal au spectre de  $v_i$ , le  $i^{\text{ème}}$  vecteur de base. Ici, on exigera de plus que le spectre par rapport à  $E$  de la paire de vecteurs minimaux de  $A$ , représentée par le  $j^{\text{ème}}$  élément de  $l-m$ , soit égal au spectre de  $\pm v_i$  par rapport à  $E$ .<sup>40</sup>

<sup>39</sup> La procédure *chercher-sys-gen* calcule automatiquement un système de générateurs du groupe des automorphismes d'une forme parfaite  $A$  modulo  $\pm$ identité. Elle fait appel pour cela à la fonction *chercher-new-auto*, variante de la fonction *chercher-equivalence*.

<sup>40</sup> Dans le programme, la procédure qui calcule automatiquement le stabilisateur d'une face d'un domaine s'appelle *calculer-groupe*. Elle fait appel à la fonction *chercher-new-auto-face*, variante de la fonction *chercher-equivalence*.

Au départ, une dizaine de secondes sont nécessaires au programme pour définir un environnement adapté à la recherche des éléments du groupe; ensuite, en moyenne, trois éléments de ce groupe sont calculés chaque seconde.

### 8.6.3. Tests d'équivalence entre facettes d'un même domaine :

Il s'agit ici de pouvoir décider rapidement si deux facettes données,  $F_1$  et  $F_2$ , du domaine associé à une forme parfaite  $A$  sont équivalentes, c'est-à-dire si  $F_1$  et  $F_2$  sont dans la même orbite de facettes par l'action du groupe des symétries du domaine. On désire pouvoir faire ce test sans expliciter l'orbite d'une de ces facettes.<sup>41</sup>

Notons  $E_i$ , l'ensemble des paires de vecteurs minimaux de  $A$  associés aux arêtes du domaine contenues dans  $F_i$ .

Les facettes  $F_1$  et  $F_2$  sont équivalentes si et seulement si il existe une matrice  $S$  représentant un élément du groupe des symétries du domaine vérifiant  $S(E_1) = E_2$ . Il s'agit d'exhiber, si elle existe, une telle matrice  $S$ .

En particulier,  $E_1$  et  $E_2$  doivent contenir le même nombre d'éléments.

Comme précédemment, la condition  $S(E_1) = E_2$  permet d'affiner l'heuristique.

Dans l'algorithme général,  $j$  appartient à  $l_i$  si et seulement si le spectre du  $j^{\text{ème}}$  vecteur de  $l-m$  est égal au spectre de  $v_i$ , le  $j^{\text{ème}}$  vecteur de base. Ici, on exigera, de plus, que le spectre par rapport à  $E_2$  de la paire de vecteurs minimaux de  $A$  représentée par le  $j^{\text{ème}}$  élément de  $l-m$ , soit égal au spectre de  $\pm v_i$  par rapport à  $E_1$ .

Finalement, la boucle principale de recherche sera la suivante :

- si la pile est vide, la recherche a échoué;
- sinon,
  - on extrait le sommet en tête de pile;
  - si ce sommet correspond à un  $n$ -tuple de  $w_i$  complètement rempli, on calcule l'élément  $S$  du groupe qui lui correspond;
    - si  $S(E_1) = E_2$ , la recherche a abouti;
    - sinon,
      - on poursuit la recherche;
  - on calcule tous les successeurs de ce sommet et on les introduit en tête de pile.

<sup>41</sup> Dans le programme, on distingue deux cas.

Si  $F_1$  et  $F_2$  sont des faces du domaine, mémorisées sous forme matricielle, le test d'équivalence peut se faire à l'aide de la fonction *dans-la-meme-orbite* qui vit dans l'environnement défini par la procédure *init-dans-la-meme-orbite*.

Si  $F_1$  et  $F_2$  sont deux facettes du domaine, mémorisées dans des structures de type "ent", l'équivalence entre  $F_1$  et  $F_2$  sera testée par la fonction *pseudo-equivalent* qui vit dans l'environnement défini par la procédure *init-pseudo-equivalent*.

Il faut, en général, entre dix et vingt secondes pour décider si deux facettes données sont équivalentes ou si, au contraire, elles appartiennent à des orbites de facettes distinctes.

9. Conclusion :

Les calculs que j'ai effectués sur ordinateur démontrent qu'il n'y a que trente-trois classes de formes parfaites en dimension sept, mais les résultats de ces calculs contiennent des informations beaucoup plus précises.

Ils permettent, par exemple, de connaître le nombre exact d'orbites de voisines d'une certaine forme parfaite, qui sont contenues dans une classe donnée; ils décrivent ainsi complètement l'évaluation de la fonction  $f$  du théorème 10 sur les classes d'équivalence des domaines de Voronoï.

Les tableaux suivants résument cette information. Par le théorème 10, on sait qu'ils doivent être symétriques.

Dimension 2 :

det. (min = 2)	1		1
3	3	1	1

fig. 13

Dimension 3 :

det. (min = 2)	1		1
4	6	1	1

fig. 14

Dimension 4 :

det. (min = 2)	1		1	2
4	12	1	1	1
5	11	2	1	1

fig. 15

Dimension 5 :

det. (min = 2)	1		1	2	3
4	21	1	2	1	1
81 / 16	15	2	1		
6	15	3	1		

fig. 16

Dimension 6 :

det. (min = 2)	4	1 2 3 4 5 6 7
3	36	1 3 1 3 2 1 1
243 / 64	27	2 1 1 1 1 1 1
4	30	3 1 1 1 1 1 1
81 / 16	22	4 2 1 1 1 1 1
343 / 64	21	5 1 1 1 1 1 1
351 / 64	21	6 1 1 1 1 1 1
7	21	7 1 1 1 1 1 1

fig. 17

Dimension 7 :

det. (min = 2)	4	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33
2	63	1 11 1 6 4 10 6 18 10 1 8 9 1 3 7 1 3 6 4 1 2 2 7 2 3 3 3 3 5 6 2 4 1
8192 / 2187	28	2 1
243 / 64	36	3 6 3 1 3 3 2 1
4	42	4 4 1
4	36	5 10 1 1 6 1 7 2 2 4 1 3 1 1 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1
1024 / 243	32	6 6 3 1 2 2 1
135 / 32	34	7 18 3 7 2 23 2 4 5 5 1 2 2 1 1 1 1 3 2 3 3 1 4 1 1 1 1 1 1 1 1
9 / 2	32	8 10 2 1 2 1 2 1
71344 / 15625	28	9 1
147 / 32	32	10 8 1 1 2 4 1 2 2 2 1
1120 / 243	30	11 9 1 1 4 5 2 4 1
10240 / 2187	28	12 1
4805 / 1024	28	13 3 1
10136 / 2187	28	14 7 1 1 3 1 5 1
38809 / 8192	28	15 1
128 / 27	30	16 3 1 1 1 2 1
10528 / 2187	28	17 6 1 1 1 2 1
2465 / 512	18	18 4 1 2 1
392 / 81	28	19 1
10658 / 2187	28	20 2 1
3584 / 729	28	21 2 1
315 / 64	30	22 7 1 1 1 5 1
10780 / 2187	28	23 2 1 1 2 1
10806 / 2187	28	24 3 1 2 1
11008 / 2187	28	25 5 1 1 3 1
81 / 16	30	26 3 1
81 / 16	30	27 3 1
81 / 16	30	28 5 1 1 4 1
171 / 32	28	29 6 1
343 / 64	28	30 2 1
45 / 8	28	31 4 1
189 / 32	28	32 1
8	28	33 1

fig. 18

Finalement, cette classification démontre également le théorème suivant qui confirme et généralise plusieurs théorèmes dus à Watson :

**Théorème :**

*Tout réseau parfait de dimension inférieure ou égale à sept possède une base formée de vecteurs minimaux.*

La description détaillée que j'ai obtenue des domaines de Voronoï en dimension inférieure ou égale à sept, peut être consultée à l'Université de Neuchâtel. De plus, un exemplaire complet et entièrement commenté des programmes LISP qu'il m'a fallu développer pour obtenir ces résultats est également disponible et déposé au même endroit. Le fascicule qui regroupe ces programmes comprend plus de 460 pages.

Finalement, j'aimerais signaler que la tendance actuelle vise à une généralisation de l'algorithme de Voronoï, ou plutôt de son domaine d'application.

L'algorithme de l'explorateur s'intègre bien dans ce contexte, puisqu'il donne une généralisation, aussi bien du domaine d'application de l'algorithme de Voronoï (facettes d'un domaine), que de l'algorithme lui-même, en lui permettant d'agir récursivement, en profondeur.

10. Bibliographie :

- Ba 1. Barnes, E.S. *Note on extreme forms.*  
Can. J. Math. 7, pp. 150 - 154; (1955).
- Ba 2. Barnes, E.S. *The perfect and extreme senary forms.*  
Can. J. Math. 9, pp. 235 - 242; (1957).
- Ba 3. Barnes, E.S. *The complete enumeration of extreme senary forms.*  
Phil. Trans. R. Soc. Lond. A 249, pp. 461 - 506; (1957).
- Ba 4. Barnes, E.S. *On a theorem of Voronoï.*  
Proc. Camb. phil. Soc. 53, pp. 537 - 539; (1957).
- Ba 5. Barnes, E.S. *Criteria for extreme forms.*  
J. Aust. math. Soc. 1, pp. 17 - 20; (1959).
- BM 1. Bergé, A.-M. & Martinet, J.  
*Sur la constante d'Hermite (étude historique).*  
Sém. de Th. des Nombres de Bordeaux 2, Exposé 8,  
pp. 8-01 - 8-15; (1985-1986).
- BM 2. Bergé, A.-M. & Martinet, J.  
*Sur un problème de dualité lié aux sphères en géométrie des nombres.*  
J. of Number Theory, Vol. 32, No. 1, pp. 14 - 42; (1989).
- Bc 1. Berger, M. *Géométrie 3 / convexes et polytopes. polyèdres réguliers, aires et volumes.*  
Cedic, Fernand Nathan, (publié avec le concours du C.N.R.S.); (1978).
- Bl 1. Blichfeldt, H. F. *On the minimum value of positive real quadratic forms in 6 variables.*  
Bull. Am. math. Soc. 31, 386; (1925).
- Bl 2. Blichfeldt, H. F. *The minimum value of quadratic forms, and the closest packing of spheres.*  
Math. Annalen 101, pp. 605 - 608; (1929).
- Bl 3. Blichfeldt, H. F. *The minimum values of quadratic forms in six, seven and eight variables.*  
Math. Z. 39, pp. 1 - 15; (1935).
- Br 1. Brooks, Rodney A. *Programming in Common LISP.*  
John Wiley & Sons, Inc.; (1985).
- CS 1. Conway, J. H. & Sloane, N. J. A.  
*Sphere-packings, lattices and groups.*  
Grundlehren der mathematischen Wissenschaften 290,  
Springer-Verlag; (1988).
- CS 2. Conway, J. H. & Sloane, N. J. A.  
*Low-dimensional lattices. III. Perfect forms.*  
Proc. R. Soc. Lond. A 418, pp. 43 - 80; (1988).
- Co 1. Coxeter, H. S. M. *Extreme forms.*  
Can. J. Math. 3, pp. 391 - 441; (1951).

- DEC 1. Digital Equipment Corporation.  
*System Access Guide.*  
*Implementation and Extensions to Common LISP.*  
*Object Reference Manual.*  
 VAX LISP/VMS, Digital Equipment Corporation;  
 (1986, 1987, 1989).
- GS 1. Guy, L. & Stecic J.R. *Common LISP : The Language.*  
 Digital Press, 465 pages; (1984).
- JS 1. Jaquet, D.-O. & Sigrist, F.  
*Formes quadratiques contiguës à  $D_7$ .*  
 C. R. Acad. Sci. Paris, t.309, Série I, pp. 641 - 644; (1989).
- Ja 1. Jaquet, D.-O. *Domaines de Voronoï et algorithme de réduction des formes quadratiques définies positives.*  
 Sém. de Th. des Nombres de Bordeaux 2, pp. 163 - 215; (1990).
- Ja 2. Jaquet, D.-O. *Classification des réseaux dans  $\mathbb{R}^7$  (via la notion de formes parfaites).*  
 Astérisque, Soc. Math. de France; (sous presse).
- Kn 1. Kneser, M. *Two remarks on extreme forms.*  
 Can. J. Math. 7, pp. 145 - 149; (1955).
- KZ 1. Korkinc, A. & Zolotareff, G.  
*Sur les formes quadratiques.*  
 Math. Annalen 6, pp. 336 - 389; (1873).
- KZ 2. Korkinc, A. & Zolotareff, G.  
*Sur les formes quadratiques positives.*  
 Math. Annalen 11, pp. 242 - 292; (1877).
- La 1. Larmouth, J. *The enumeration of perfect forms.*  
 Dans "Computers in number theory" (A. O. L. Atkin & B. J. Birch editors), pp. 237 - 239; New York : Academic Press; (1971).
- Mo 1. Mordcill, L. J. *Observation on the minimum of a positive quadratic form in eight variables.*  
 J. Lond. math. Soc. 19, pp. 3 - 6; (1944).
- Oc 1. Ocsterlé, J. *Empilements de sphères.*  
 Sém. N. Bourbaki 42, Exposé 727, Vol. 1989-90; (1990).
- Sc 1. Scott, P. R. *On perfect and extreme forms.*  
 Thesis, Department of Mathematics, Univ. of Adelaide; (1963).
- Sc 2. Scott, P. R. *On perfect and extreme forms.*  
 J. Aust. math. Soc. 4, pp. 56 - 77; (1964).
- Sc 3. Scott, P. R. *The construction of perfect and extreme forms.*  
 Can. J. Math., Vol. 18, pp. 147 - 158; (1966).
- Se 1. Serre, J.-P. *Cours d'arithmétique.*  
 2<sup>e</sup> édition, Presses Univ. de France, Paris; (1977).

- Si 1. Sigrist, F. *Formes quadratiques encapsulées.*  
Sém. de Th. des Nombres de Bordeaux 2, pp. 425 - 429; (1990).
- Su 1. Sushbaev, S. Sh. *Voronoi neighborhood of perfect form  $\phi_{15}(x_1, x_2, \dots, x_7)$ .*  
Vop. Vychisl. prikl. Mat. 77, pp. 48 - 56; (1985).
- St 1. Stacey, K. C. *The enumeration of perfect quadratic forms in seven variables.*  
Ph. D. Dissertation, University of Oxford; (1973).
- St 2. Stacey, K. C. *The enumeration of perfect septenary forms.*  
J. Lon. math. Soc. (2) 10, pp. 97 - 104; (1975).
- St 3. Stacey, K. C. *The perfect septenary forms with  $\Delta_4 = 2$ .*  
J. Aust. math. Soc. (A) 22, pp. 144 - 164; (1976).
- Sti 1. Stiemke, E. *Über positive Lösungen homogener linearer Gleichungen.*  
Math. Annalen 76, pp. 340 - 342; (1915).
- Ve 1. Vetchinkin, N. M. *Uniqueness of the classes of positive quadratic forms on which the values of Hermite constants are attained for  $6 \leq n \leq 8$ .*  
Trudy mat. Inst. imeni V. A. Steklova 152, pp. 34 - 86; (1980).  
(traduction anglaise dans Proc. Steklov Inst. Math. (3), pp. 37 - 95; (1982)).
- Vo 1. Voronoï, G. *Sur quelques propriétés des formes quadratiques positives parfaites.*  
J. reine angew. Math. 133, pp. 97 - 178; (1908).
- Wa 1. Watson, G. L. *On the minimum of a positive quadratic form in  $n (\leq 8)$  variables (verification of Blichfeldt's calculation).*  
Proc. Camb. phil. Soc. 62, 719; (1966).
- Wa 2. Watson, G. L. *On the minimal points of perfect septenary forms.*  
Mathematika 16, pp. 170 - 177; (1969).
- Wa 3. Watson, G. L. *Integral quadratic forms.*  
Cambridge, University Press; (1970).
- Wa 4. Watson, G. L. *On the minimal points of a positive quadratic form.*  
Mathematika 18, pp. 60 - 70; (1971).
- Wa 5. Watson, G. L. *The number of minimum points of a positive quadratic form.*  
Dissertations Math. 84, pp. 1 - 46; (1971).
- Wa 6. Watson, G. L. *The number of minimum points of a positive quadratic form having no perfect binary section with the same minimum.*  
Proc. Lond. math. Soc. (3) 24, pp. 625 - 646; (1972).

<i>Avant-propos</i> .....	5
1. Introduction.....	7
2. Définitions.....	9
3. La fonction d'Hermite $\mu$ .....	11
4. Généralités sur les convexes.....	12
4.1. Propriétés fondamentales.....	12
4.2. Relèvement d'un cône saillant.....	14
4.2.1. Définitions et propriétés fondamentales.....	14
4.2.2. Théorème fondamental.....	17
4.3. Faces voisines.....	18
4.3.1. Définition et propriété fondamentale.....	18
4.3.2. Connexion.....	19
4.3.2.1. Théorème fondamental.....	19
4.3.2.2. Faces maximales.....	20
4.4. Graphe des faces d'un cône saillant.....	22
5. Domaines de Voronoï.....	23
5.1. Domaine associé à une forme parfaite.....	23
5.2. Domaines voisins et formes voisines.....	24
5.2.1. Définitions et propriétés fondamentales.....	24
5.2.2. Existence.....	24
5.2.3. Illustration du cas des formes à 2 variables.....	25
6. Graphe de Voronoï.....	32
6.1. Définition.....	32
6.2. Connexité du graphe.....	32
6.3. Actions de $Gl_n(\mathbb{Z})$ .....	33
7. Quelques propriétés des graphes localement finis.....	35
8. Algorithme de Voronoï.....	40
8.1. Description.....	40
8.2. Symétries.....	40
8.3. Calcul des faces d'un domaine.....	41
8.3.1. Algorithme de la cascade.....	41
8.3.1.1. Description.....	41
8.3.1.2. Symétries.....	42
8.3.1.3. Implémentation en LISP.....	44

8.3.2. Algorithme de l'explorateur.....	54
8.3.2.1. Description.....	54
8.3.2.2. Symétries.....	55
8.3.3. Fusion de ces deux algorithmes.....	56
8.3.3.1. Description.....	56
8.3.3.2. Implémentation en LISP.....	57
8.4. Calcul automatique des voisines.....	64
8.4.1. Description.....	64
8.4.2. Implémentation en LISP.....	65
8.5. Identification automatique des voisines.....	66
8.5.1. Description.....	66
8.5.2. Implémentation en LISP.....	68
8.6. Variantes de l'algorithme décrit au paragraphe 8.5.....	69
8.6.1. Calcul automatique du groupe des automorphismes.....	69
8.6.1.1. Description.....	69
8.6.1.2. Implémentation en LISP.....	70
8.6.2. Calcul automatique du stabilisateur d'une face.....	70
8.6.3. Tests d'équivalence entre facettes d'un même domaine.....	71
9. Conclusion.....	72
10. Bibliographie.....	75
Table des matières.....	79