Comment compter les trous dans une meule de fromage suisse : ou, l'homologie pour les gourmands

Sara Derivière, Anik Trahan
Tomasz Kaczynski

Résumé

Nous présentons une approche gourmande de l'homologie, motivée dès le départ par des problèmes concrets que l'on se propose de résoudre par la suite grâce à cet outil mathématique puissant (qu'est l'homologie !) assisté par des programmes informatiques, facilement manipulables par l'utilisateur non-informaticien, voir non-mathématicien.

1 Introduction

L'homologie est une spécialité de la topologie algébrique. La topologie est la science qui étudie les propriétés géométriques invariantes d'un objet quand celui-ci est étiré, tordu ou rétréci de manière continue. L'algèbre étudie les propriétés des ensembles munis d'une structure algébrique (groupes, anneaux, lois de composition, ...). Ainsi, la topologie algébrique est une branche des mathématiques où l'algèbre générale est utilisée dans l'étude des espaces topologiques[1]. Cet article est plutôt basé sur une sous-branche de la topologie algébrique : la topologie computationnelle. Celle-ci se concentre sur les applications informatiques et algorithmiques. Tout ceci est très impressionnant, mais pourquoi s'intéresser à une chose apparemment si barbare ? C'est ce à quoi se propose de répondre la Section 2. Une fois que vous aurez compris tout le génie de l'homologie et ce qu'il vous permettra de faire, vous aurez hâte d'en apprendre d'avantage, d'où le besoin de définir certains objets dans la Section 3 que nous manipulerons dans la Section 4. Tout ceci nous permettra de definir, plus ou moins rigoureusement, l'homologie des ensembles dans la Section 5. Finalement, dans la Section 6, vous serez en mesure de compter le nombre de trous dans une meule de fromage suisse, de calculer le nombre de cratères sur la lune, de prédire si tel ou tel labyrinthe est réalisable, et bien plus encore... Vous pourrez ainsi impressionner petits et grands lors de votre prochain souper familial !

[1] Une définition plus rigoureuse de la topologie algébrique ainsi qu'une présentation des recherches possibles dans ce vaste domaine se trouve sur la page web du GRTC : http ://www.dmi.usherb.ca/ kaczyn/grtc/index.html

2 L'homologie, pour quoi faire ?

Ainsi donc, les calculs d'homologie seraient utiles ? Oui, nous l'affirmons ! Voici quelques exemples concrets. Imaginons un instant que vous vouliez résoudre un labyrinthe. Vous pourriez vous demander, avant de commencer à vous atteler à cette tache, si le labyrinthe en question admet bien une solution. Par exemple, dans l'exemple de la Figure 1, existe t-il un chemin permettant de joindre le côté inférieur gauche au côté supérieur droit ?

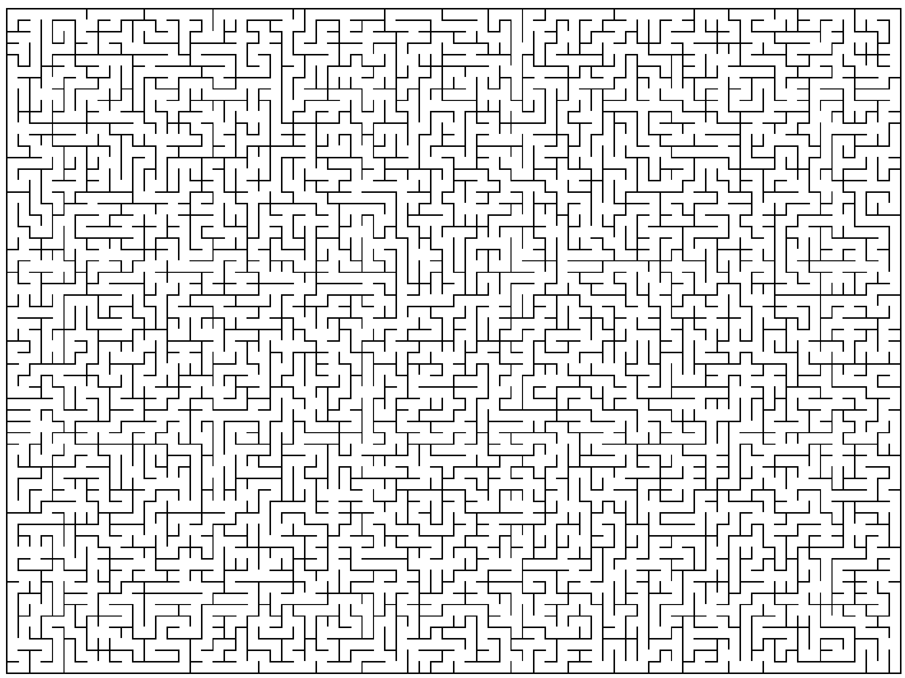

FIG. 1 – La souris pourra t-elle jamais atteindre le morceaux de fromage ?

Imaginons à présent qu'un inconnu frappe à votre porte, un soir d'hiver enneigé, vous montre la photo de la Figure 2 et vous promet un voyage à Miami Beach si vous lui donnez le nombre exact de cratères ? (Presque) rien n'est impossible...

Et si vous étiez un riche industriel, fabriquant des pièces électroniques, voir Figure 3, vous aimeriez donner à vos distributeurs l'assurance de la qualité de vos pièces.

Et bien sûr, une question primordiale lors d'un bon repas, quel morceau de fromage choisir ? Lequel a le moins de trous, et donc le plus de succulente saveur

FIG. 2 – Combien de cratères sont visibles sur cette partie de la lune ?

à dévorer ? Vous pourrez répondre à toutes ces questions grâce aux simples calculs d'homologie. Incroyable mais vrai !

3 Approche cubique

L'homologie se propose d'attribuer, à tout objet géométrique, une suite de groupes abéliens[2] dont les dimensions vont répondre à nos questions ! Par exemple, si l'on dénote par X notre objet d'étude et $H_0(X),\ H_1(X),\ H_2(X), \ldots$ la suite de groupes abéliens calculée, les dimensions de $H_0(X),\ H_1(X)$ et $H_2(X)$ déterminent respectivement le nombre de composantes connexes, le nombre de boucles et le nombre de chambres vides (de cavités) dans X, voir Figure 4.

Pour calculer ces groupes abéliens, nous devons considérer des images ou des données représentés par des ensembles cubiques dont voici la définition (voir par exemple [4] et [2] pour plus de détail).

[2]Un groupe abélien est une structure algébrique dont les éléments peuvent être additionnés, soustraits, multipliés par des scalaires entiers, mais la division par des scalaires n'est pas permise.

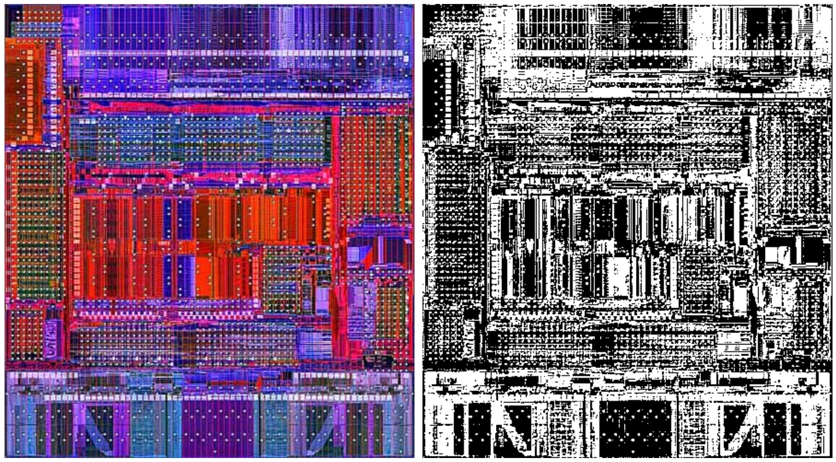

FIG. 3 – Tous les circuits imprimés ont-ils été reproduits conformément à l'original ? Manque t-il des connections ? Voici les questions que l'on peut se poser en regardant la piéce de gauche, plus facilement comparable avec une autre pièce dans sa représentation en noir et blanc.

$$\begin{array}{ccc} \text{Objet géométrique} & & \text{groupes abéliens} \\ X & \mapsto & H_*(X) = \{H_k(X)\} \end{array}$$

$$\begin{aligned} \dim H_0(X) &= \#\ \text{composantes connexes de } X \\ \dim H_1(X) &= \#\ \text{boucles de } X \\ \dim H_2(X) &= \#\ \text{chambres vides dans } X \end{aligned}$$

FIG. 4 – Récapitulatif

Définition 1 Un *ensemble cubique* est une union finie de produits d'intervalles. Dans notre cas, nous considérons uniquement des intervalles de longueurs un :

$$\bigcup_{\text{finie}} [a_1, a_1 + 1] \times [a_2, a_2 + 1] \times \cdots [a_n, a_n + 1], \quad a_i \in \mathbf{Z}.$$

Dans ce cas, un intervalle simple est un segment (de longueur un), le produit de deux intervalles un carré, de trois intervalles un cube, ...

3.1 Motivation pour l'approche cubique

Mais pourquoi utiliser cette approche cubique ? Il existe différents cadres mathématiques permettant de définir et calculer l'homologie des ensembles. Les plus utilisés sont l'approche cubique (que nous privilégions ici) et l'approche simpliciale. En voici les deux principales raisons :

Motivation mathématique Le produit cartésien de deux simplexes (éléments de bases de l'homologie simpliciale) n'est pas nécessairement un simplexe, alors que le produit de cubes est toujours un cube :

$$\begin{array}{rcl} \text{simplex} \times \text{simplex} & \neq & \text{simplex} \\ \text{cube} \times \text{cube} & = & \text{cube} \end{array}$$

Motivation informatique Comme nous l'avons laisser entendre précedemment, nous allons utiliser l'ordinateur et des programmmes informatiques pour calculer efficacement les homologies. Et comme chacun sait, une image sur un écran est en fait une collection de pixels, donc de petits carrés : c'est un ensemble cubique !

$$X = \bigcup_{\text{finie}} n\text{-pixels}$$

Les programmes que nous utiliserons par la suite s'appliquerons en général sur des fichiers dits *cubiques* et portant l'extension `.cub`.

4 Programmation

Nous présentons dans cette section le programme et les commandes dont nous avons besoin pour transformer un fichier image `.bmp` en fichier cubique `.cub` :

$$\text{fichier image} \quad \mapsto \quad \text{fichier.cub}$$

Les programmes que nous utiliserons dans toute la suite proviennent d'une bibliothèque de programmes librement accessible sur le net appelée CHomP[3].

Le programme qui permet d'obtenir un fichier cubique à partir d'un fichier `.bmp` est `bmp2pset`. Chaque commande est défini lorsque l'on tape uniquement son nom. Par exemple, le mode d'emploi de cette commande retourné par le programme est montré à la Figure 5.

Par exemple, le fichier cubique du labyrinthe le la Figure 1, obtenu suite à l'appel :

`bmp2pset maze.bmp maze.cub`,

est affiché à la Figure 6.

5 Définition de l'homologie

Cette section, la plus mathématique, permet enfin de définir exactement l'homologie (voir [1] pour une approche plus rigoureuse des complexes cubiques).

[3] Page web de Computational Homology Project : http ://www.math.gatech.edu/ chomp/

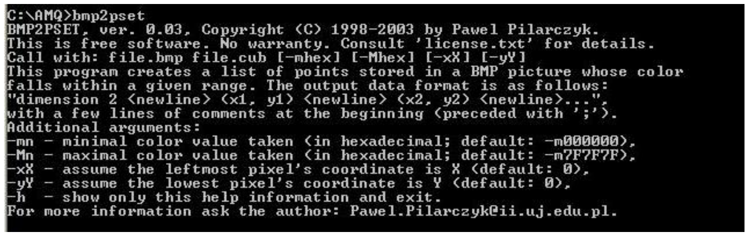

FIG. 5 – Le mode d'emploi de chacune des commandes est très utile.

5.1 Complexe cubique

Définition 2 *Un* cube élémentaire Q *est un produit cartésien d'intervalles de longueur un ou dégénérés (réduit à un point) :*

$$Q = I_1 \times I_2 \times \cdots \times I_n \subset \mathbf{R}^n, I_j = [a, a+1] \text{ ou } I_j = [a] := [a, a], \ a \in \mathbf{Z}$$

Les cubes pleins *sont des produits cartésiens d'intervalles élémentaires non-dégénérés :* $I_j = [a, a+1] \ \forall_j$.

Remarque 3 *Les pixels sont des* cubes pleins *de dimension 2.*

Définition 4 Un *ensemble cubique* X est une union finie de cubes élémentaires,

$$X := \bigcup_{\text{finie}} \{\text{cubes élémentaires}\}.$$

On note $C_k(X)$ le *groupe abélien libre* engendré par l'ensemble $\mathcal{K}_k(X)$ de k-cubes dans X. $C_k(X)$ est appelé le groupe des k-chaînes, ces éléments sont de la forme :

$$c = \sum \alpha_i Q_i$$

5.2 Opérateur de la frontière cubique – idée

Nous définissons à présent un opérateur (appelé *opérateur de la frontière* ou *opérateur de bord*) entre deux groupes de chaînes de dimensions consécutives. Pour développer l'intuition de cet opérateur, regardons l'exemple suivant

Exemple 5 Considérons le carré Q (qui est un par définition un cube élémentaire) de la figure 7. Intuitivement le bord de ce cube est la somme de chacun de ces quatre côtés. Il faut toutefois prendre garde au sens des vecteurs. Selon l'orientation classique de la base et si l'on parcourt le bord du carré dans le sens contraire des aiguilles d'une montre, A_2 et B_1 sont parcourus positivement

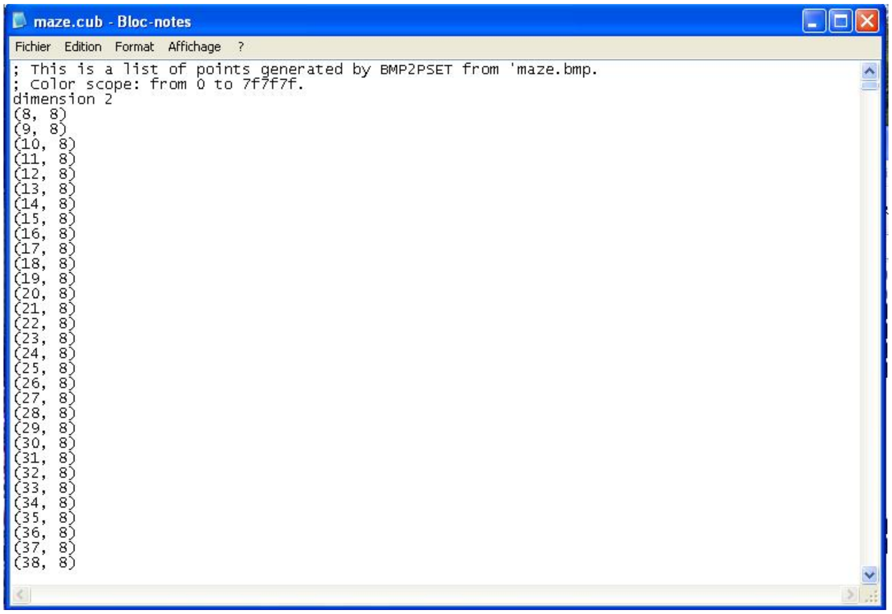

FIG. 6 – Le début du fichier cubique correspondant au labyrinthe de la Figure 1.

tandis que B_2 et A_1 sont parcourus négativement. Ainsi l'opérateur de bord appliqué à Q, noté ∂Q est :

$$\partial Q = A_2 + B_1 - B_2 - A_1$$

Définition 6 *Frontière et homologie cubique* De manière générale et non-rigoureuse, la frontière d'un cube de dimension k est la somme alternée[4] de tous les cubes de dimension $k-1$, c'est à dire de toutes ces faces qui le composent :

$$\partial Q = \sum \pm \ (k-1)\text{-faces de } Q.$$

De plus, on peut vérifier que la composition de deux opérateurs de frontière correspond à l'opérateur nul :

$$\partial \circ \partial = 0.$$

Définition 7 Un *complexe cubique $C(X)$* est la donnée de groupes de chaînes de toutes les dimensions et des opérateurs de bords entre chaque groupe de chaînes consécutif :

$$C(X) := \{C_k(X), \partial_k\}$$

[4]La règle d'alternance du signe est trop technique pour cette présentation, nous référons à [4], chap. 2 pour la formule exacte.

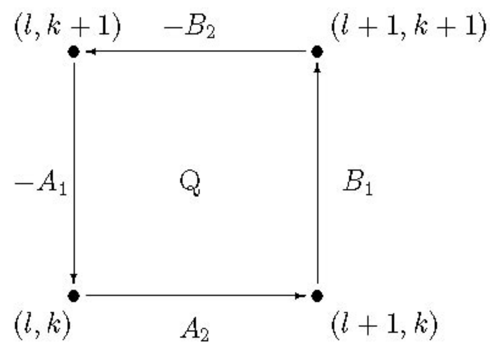

FIG. 7 –

Étant donné un complexe cubique $C(X)$, on peut définir l'homologie de X de dimension k, pour tout $k \geq 0$. Pour cela, on utilise la notion de cycles et bords.

Définition 8 Pour tout $k > 0$, on note ∂_k l'opérateur de frontière entre $C_k(X)$ et $C_{k-1}(X)$. On note respectivement $Z_k(X)$ et $B_k(X)$ le noyau de ∂_k et l'image de ∂_{k+1}. Une k-chaîne de $C_k(X)$ est appelé un *k-cycle* si son bord est nul, c'est-à-dire si $\partial_k z = 0$. Ainsi, par définition, l'ensemble de tous les k-cycles est exactement $Z_k(X)$.

Une k-chaîne z de $C_k(X)$ est appelé un *k-bord* s'il existe une $(k+1)$-chaîne c tel que $\partial_{k+1} c = z$. L'ensemble de tous les k-bords est noté $B_k(X)$. Puisque $\partial^2 = 0$, alors $B_k(X)$ est un sous-groupe de $Z_k(X)$.

Deux cycles c_1 et c_2 de $Z_k(X)$ sont *homologues* si $c_2 - c_1$ est un bord. Il s'agit d'une relation d'équivalence. Finalement, le *$k^{ième}$ groupe d'homologie* de X est le groupe quotient

$$H_k(X) := \frac{Z_k(X)}{B_k(X)} = \frac{\text{cycles}}{\text{bords}}.$$

Intuitivement, l'homologie detecte les cycles qui ne sont pas triviaux, dans le sens qu'ils ne sont pas des bords de chaînes de dimensions plus élevées.

Pour plus de détails sur l'homologie, consultez par exemple le livre [3].

6 Exemples de calcul d'homologie

Nous avons vu dans la Section 4 comment créer un fichier cubique avec l'extension `.cub` à partir d'un fichier image `.bmp`. A présent, nous transformons le fichier cubique en complexe de chaîne grâce à la commande `cubchain` sur lequel nous pourrons calculer l'homologie. La commande exacte est la suivante :

cubchain maze.cub maze.chn

Ensuite, on calcule l'homologie par l'appel de la fonction `homchain` :

homchain maze.chn

Cependant, le programme possède une seconde commande permettant de calculer l'homologie directement sur le fichier cubique : `homcubes`. Dans certains cas, cette dernière est plus avantageuses car beaucoup plus rapide. A l'appel de ces deux commandes, le programme nous retourne toutes les valeurs de $H_k(X)$ dans l'ordre jusqu'à la dernière non-nulle, ainsi que le temps qui a été nécessaire au calcul. Par exemple, la Figure 8 montre les deux résultats obtenus suite à l'appel de ces deux fonctions sur un image représentant le chiffre huit.

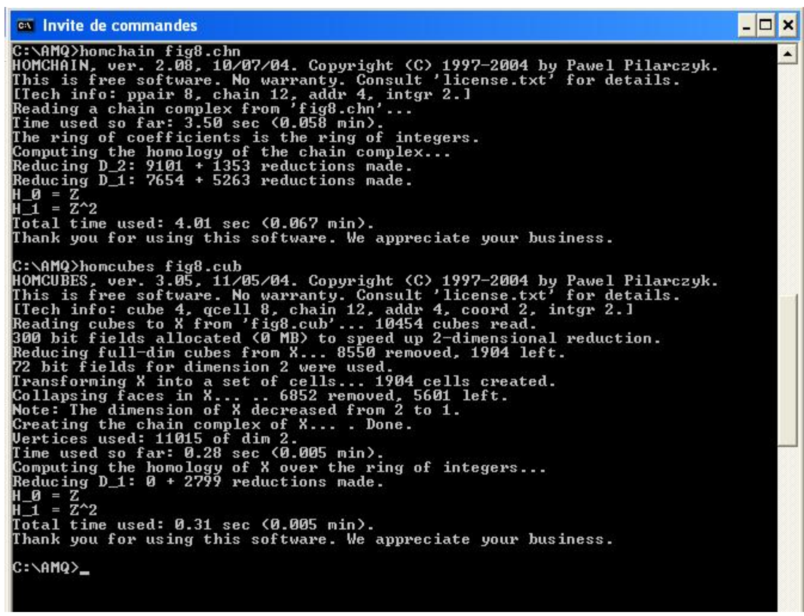

FIG. 8 – Résultat obtenu suite à l'appel des deux fonctions `homchain` et `homcubes`.

Le résultat obtenu est sans surprise. En effet, le chiffre 8 est formé d'une seule composante connexe et de deux boucles, la dimension de H_0 doit donc être 1, celle de H_1 égale à 2 et toutes les homologies de dimensions supérieures doivent être nulles, ce qui correspond exactement aux résultats retournés. Cependant, `homcubes` a effectué le calcul en 0.31 secondes alors qu'il a fallut plus de 4 secondes pour obtenir le même résultat avec `homchain`!

Nous voulons à présent calculer le nombre de trous dans la meule de fromage de la Figure 9 appelée `slicing.cub`. Cette image a été obtenue à partir du fichier cubique par l'appel de la fonction `showcubes` :

```
showcubes slicing.cub
```

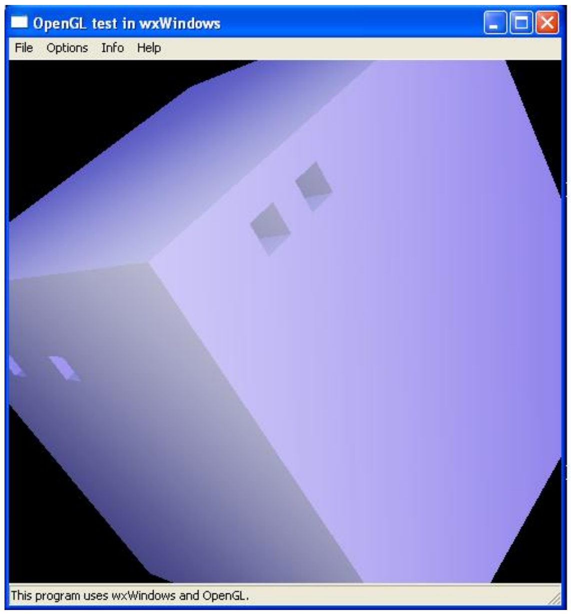

FIG. 9 – Combien de trous dans ce fromage ?

Nous calculons donc l'homologie de ce fromage grâce à l'appel :

```
homcubes slicing.cub
```

dont le résultat est montrée à la Figure 10.

D'après ce calcul, on en conclut que ce morceau de fromage (un seul morceau parce que $H_0(X) = Z$ dont la dimension est 1) a exactement 6 cycles ($H_1(X) = Z^6$) et 3 cavités à l'intérieur même du fromage, donc invisible sans trancher le fromage ($H_2(X) = Z^3$). On peut ensuite vérifier ces résultat en découpant le fromage en tranche par la commande suivante :

```
cubslice slicing.cub tranche
```

puis en regardant tranche par tranche ce que l'on obtient. Par exemple la première tranche de ce morceau est montrée à la Figure 11 grâce à la commande :

```
showcubes tranche1.cub
```

7 Conclusion

Ainsi, les mathématiques et l'homologie nous ont permis de résoudre rapidement des problèmes divers et variés. Chacun de vous est maintenant en mesure

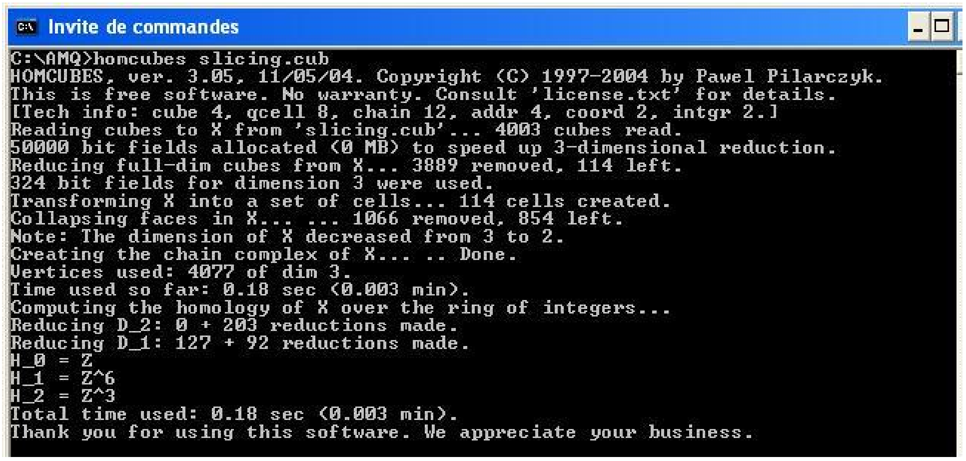

FIG. 10 – Homologie du fromage.

de calculer sans erreur le nombre de trou dans une meule de fromage suisse, et ça, qui l'eut crû...

Nous vous invitons à présent a venir vous amuser avec la biblitohèque de programmes CHomP pour apporter une réponse à vos nombreuses questions existentielles !

Références

[1] J. Blass, H. Wolsztynski : : Cubical Polyhedra and homotopy.

[2] R. Ehrenborg, G. Hetyei : Generalization of Baxter's Theorem and cubical homology. J. Combinatorial Theory, Series A **69**, 233-287, 1995.

[3] P.J. Hilton, S. Wylie : Homology Theory. Cambridge. University Press, Cambridge, 1960.

[4] T. Kaczynski, K. Mischaikow, M. Mrozek : Computational Homology. Appl. Math. Sci. Series **157**, Springer-Verlag, New York 2004.

Département de mathématiques
Université de Sherbrooke,
Sherbrooke (Québec), Canada J1K 2R1
{sara.deriviere, tomasz.kaczynski}@usherbrooke.ca

Department of Computer Science
Bishop's University
Sherbrooke (Québec), J1M 1Z7, Canada
anik.trahan@usherbrooke.ca

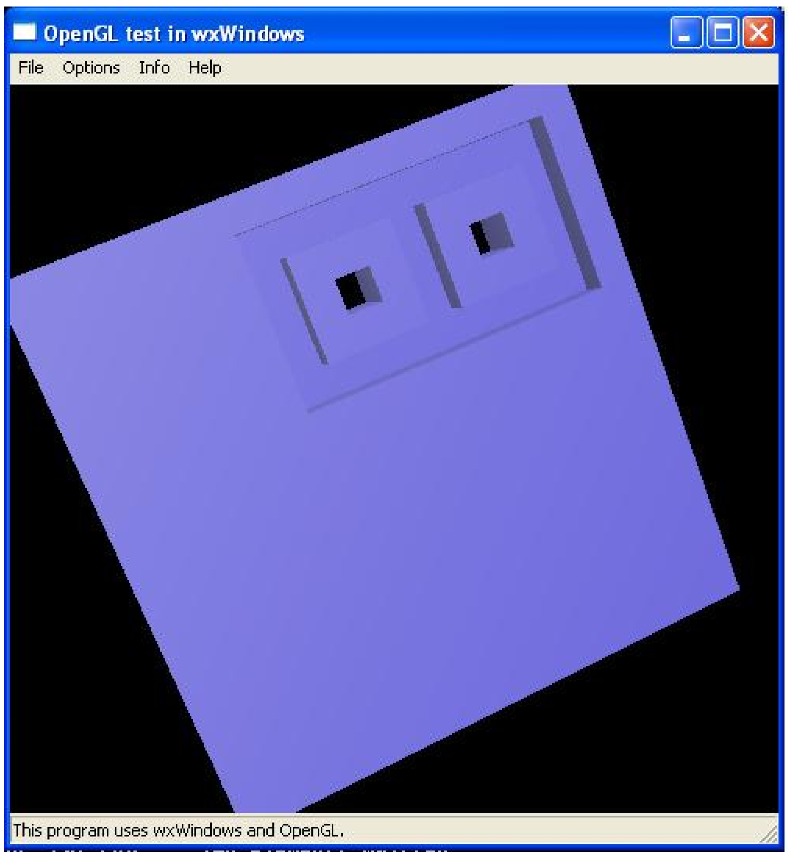

FIG. 11 – La première tranche du fromage. On ne pouvait pas voir ces trous, invisibles de l'exterieur mais détéctés par le programme `showcubes` !