

Une nouvelle représentation pour les ensembles géométriques

George Tzoumas¹, Jean-Marc Cane¹, Arnaud Kubicki¹, Dominique Michelucci¹, Sebti Foufou²

¹ CNRS UMR 5158, Le2i, Université de Bourgogne, France

² Qatar University, Qatar

Résumé

Nous présentons une nouvelle représentation pour les objets géométriques et une application de cette représentation. Elle permet de représenter non seulement les opérations booléennes sur des primitives CSG (unions, intersection, complément) mais aussi des opérations plus complexes comme des projections, des sommes de Minkowski et des extrusions. Le système algébrique obtenu est résolu par un solveur utilisant l'arithmétique d'intervalle. Nous montrons comment le solveur peut être optimisé pour cette application spécifique et comment les algorithmes topologiques existants peuvent bénéficier de cette nouvelle représentation, ce qui étend leur portée. We

present a new representation for geometric sets and an application of this representation. It allows us to describe not only boolean operations of CSG primitives (union, intersection, complement), but also more complicated sets such as projections, Minkowski sums and extrusions. The algebraic system is solved by interval arithmetic. We show how the interval solver can be tuned for this particular application and how existing algorithms for topology computation can benefit from the new representation, allowing them to deal with more types of sets.

Mots-clés : Modélisation géométrique, ensemble géométrique, arithmétique d'intervalle, CSG, projection

1. Introduction

Dans ce travail, on propose une nouvelle représentation pour les ensembles géométriques qui est en général suffisante pour couvrir une large variété d'ensembles qui apparaissent en modélisation géométrique, comme les constructions CSG (constructive solid geometry), les projections, les extrusions et les sommes de Minkowski. Etant donné l'espace de travail d -dimensionnel \mathbb{R}^d , le terme d'*ensemble géométrique* réfère à certains sous-ensembles A de \mathbb{R}^d . Une grande famille de ces ensembles sont des ensembles semi-algébriques qui peuvent être décrit par une formule booléenne sans quantificateur avec des atomes $P < 0$, $P = 0$, $P \in \mathcal{P}$ où \mathcal{P} est un ensemble fini de polynômes. Ces ensembles peuvent être combinés avec les opérateurs \neg, \vee, \wedge (signifiant le complémentaire A^C , la réunion \cup et l'intersection \cap), nous permettant de décrire les constructions CSG.

Les travaux précédents étudient les propriétés topologiques des ensembles semi-algébriques. Dans [Col75], un

algorithme appelé *Décomposition algébrique cylindrique* est présentée qui partitionne l'ensemble étudié en sous-ensembles semi-algébriques homéomorphes à des boîtes ouvertes, et une triangulation homéomorphe à l'ensemble semi-algébrique est calculée dans un deuxième temps. Dans [DJC07], un complexe simplicial homotopiquement équivalent à un ensemble donné est calculé par arithmétique d'intervalles. Dans [SH97], l'analyse par intervalles et la théorie de Morse sont combinées pour calculer la topologie d'une surface implicite dans \mathbb{R}^3 .

Quelles fonctionnalités devrait fournir une représentation idéale des ensembles géométriques ?

Premièrement, une représentation d'ensembles géométriques devrait permettre de représenter :

- les ensembles géométriques définis par des primitives, *i.e.* les feuilles des arbres CSG en CAD/CAM. Ces primitives doivent être épaisses. En général, ce n'est pas un problème dans le contexte CAD/CAM. Un exemple classique d'une primitive 3D (*i.e.* défini par des inégalités polynômiales $f(x, y, z) \leq 0$) qui n'est pas épais est le parapluie de Whitney.
- le complémentaire d'un ensemble. Remarquons que,

dans notre contexte, un ensemble et son complémentaire ont le même bord. Cette caractéristique se rapporte à la notion d'ensembles réguliers, et la régularisation des opérations booléennes, introduites par A. A. G. Requicha et R. B. Tilove [RT78] en CAD/CAM.

- l'intersection, la réunion et la différence de deux ensembles. Remarquons que l'intersection de deux ensembles épais peut ne pas être un ensemble épais, dans des cas dégénérés, par exemple lorsque les deux ensembles sont tangents l'un à l'autre de l'extérieur. Dans un tel cas, le solveur par intervalles échouera, *i.e.* il terminera mais il ne sera pas capable de décider si l'ensemble est vide ou pas : il retournera une réponse "je ne sais pas".
- la projection d'un ensemble, et la projection cylindrique d'un ensemble (voir Figure 2 ; par exemple, la projection de la sphère 3D $x^2 + y^2 + z^2 - 1 \leq 0$ orthogonalement au plan Oxy est un disque ; la projection cylindrique de cette sphère 3D est un cylindre, c'est le volume balayé par la sphère, ou le cercle, lorsqu'ils se déplacent le long de l'axe z ou y respectivement. Notons cependant, que pour étendre la méthode DJC (Delanoue, Jaulin, Cotteceau) aux projections, nous n'avons pas besoin de cette notion de volumes balayés. Les projections sont traitées dans Sec. 3.2.
- la somme de Minkowski de deux ensembles. La somme de Minkowski de A et B est l'ensemble des points $a + b$ où $a \in A$ et $b \in B$. Voir Figure 1.
- l'extrusion ou le balayage d'un ensemble, illustré en 2D dans la figure 2. En 3D, chaque transformation affine (translation, rotation, mise à l'échelle) est typiquement représentée par une matrice 4×4 , ainsi le mouvement est décrit par une fonction explicite $t \in [0, 1] \rightarrow M(t)$ où t est la variable temps et $M(t)$ la transformation affine au temps t . Parfois, l'extrusion peut être décrite comme une somme de Minkowski de la courbe de la trajectoire et de l'objet. Cependant, on n'utilise pas les sommes de Minkowski, premièrement parce-que la courbe de la trajectoire n'est pas un ensemble épais, et deuxièmement, les sommes de Minkowski ne permettent pas de tourner ou de mettre à l'échelle les objets. Remarquons que la projection cylindrique peut être représentée par une extrusion.

Deuxièmement, une représentation d'ensembles géométriques devrait permettre de décider si un ensemble est vide ou pas. Un solveur par intervalles sera utilisé pour en décider. Le solveur par intervalles terminera toujours. Il a trois sorties possible : (1) soit l'ensemble est vide ; ou (2) il n'est pas vide et un point (parfois appelé un témoin) est calculé ; en fait, le solveur peut même fournir une boîte non vide qui est entièrement incluse dans l'ensemble ; ou (3) le solveur ne peut pas décider : par exemple l'ensemble n'est pas épais (il a une mesure nulle), ou la précision de l'arithmétique flottante sous-jacente utilisée par le solveur par intervalles n'est pas suffisante, ou le solveur tombe en deçà d'espace mémoire parce-que l'ensemble contient des parties trop fines.

Troisièmement, une représentation d'ensembles géométriques devrait permettre d'introduire des contraintes sur les ensembles : il devrait permettre de spécifier que certains ensembles sont vide, et que d'autres ensembles ne le sont pas.

Motivation

L'idée principale de la méthode de Delanoue consiste à subdiviser récursivement une boîte englobante de l'ensemble E , décrit par l'arbre CSG, jusqu'à ce que la boîte étudiée B soit vide ou que l'on tombe sur une boîte B contenant un point $S \in E \cap B$ qui « voie » les autres points p de $E \cap B$. On dit que le point S est une étoile pour $E \cap B$. Delanoue, Jaulin et Cotteceau ont remarqué que si S est une étoile pour $E_1 \cap B$ et $E_2 \cap B$, alors c'est une étoile pour $(E_1 \cap E_2) \cap B$ et pour $(E_1 \cup E_2) \cap B$. On peut donc réduire le test de l'étoile aux primitives CSG uniquement. S est une étoile pour E si c'est une étoile pour toutes les primitives de E . De plus S est une étoile de la primitive $E = \{X \in \mathbb{R}^3 \mid f(X) \leq 0\}$ si et seulement si $f(S) < 0$ et qu'il n'y a pas de solution $X \in B$ au système :

$$f(X) = 0, \nabla f(X)(X - S) < 0$$

On résout ce système par analyse d'intervalles.

La méthode de Delanoue ne fonctionne qu'avec des ensemble « épais ». Une autre limitation concerne les opérations de projection.

Est-il possible d'étendre la méthode de Delanoue pour tenir compte aussi des objets définis par projection ?

On a étudié ce problème. Il s'avère que les équations du test de l'étoile pour la méthode DJC initiale sont très simples, mais elles ne sont pas plus évidentes lorsque les projections, projections d'intersections, intersections de projections, etc sont considérées. On a compris qu'une représentation d'ensembles géométriques a besoin de générer automatiquement des systèmes d'équations pour les test de l'étoile.

2. Représentation des ensembles géométriques

Cette section présente le principe de notre nouvelle représentation pour une première famille d'ensembles géométriques : les primitives et les opérations booléennes.

Chaque « ensemble épais » est représenté par un système d'équations (*i.e.*, sans inégalité) et une variable caractéristique.

Dans le paragraphe suivant, on considère le cas des primitives, d'union de deux ensembles, d'intersection de deux ensembles, de différence de deux ensembles. On n'aura pas besoin d'un traitement spécifique pour les ensembles définis par projection pour étendre la méthode DJC (Delanoue, Jaulin, Cotteceau).

On reporte la représentation des projections, sommes de Minkowski et extrusions à la section 3.

On utilise les variables x, y ou x, y, z pour les coordonnées dans l'espace visible V . Éventuellement, on utilisera les variables $x_1, y_1, z_1, x_2, y_2, z_2$ lorsqu'on aura besoin de désigner des points particuliers dans l'espace visible. D'autres identifiants (u, v, w, t , ou u_i, \dots, t_i) sont utilisés pour les variables dans l'espace non visible.

2.1. Représentations des primitives

Chaque primitive est représentée par une équation et une variable caractéristique.

Par exemple, la sphère en 3 dimensions $x^2 + y^2 + z^2 - 1 \leq 0$ est représentée par une variable caractéristique s et une équation : $x^2 + y^2 + z^2 - 1 - s = 0$. Les points de la sphère pleine : $s \leq 0$. Les points du bord plein : $s = 0$. Les points du complémentaire de la sphère pleine : $s \geq 0$ (une fois de plus, on remarque que la frontière appartient à la sphère et son complémentaire : c'est l'idée de régularisation introduite par Tilove et Requicha).

Soit $C(E)$ le nombre d'équations utilisées pour décrire une primitive géométrique E dans la représentation ci-dessus. Il est clair que $C(E) = 1$.

On suppose connue une boîte englobante de l'ensemble étudié dans l'espace visible. Le solveur par intervalles qu'on utilise a besoin d'un intervalle englobant pour chaque variable. Un intervalle pour la variable caractéristique s est calculé par l'évaluation par arithmétique d'intervalles de : $s = x^2 + y^2 + z^2 - 1$. Notons que cet intervalle n'a pas besoin d'être très fin.

Deuxième exemple. La primitive du demi-espace $x \leq 0$ est représentée par une variable h et l'équation : $h - x = 0$.

Les points dans le demi-espace plein : $h \leq 0$, les points dans le bord plein : $h = 0$, les points du complémentaire du demi-espace plein : $h \geq 0$.

Un intervalle englobant initial pour h peut être calculé à partir d'une boîte englobante de la variable visible x par arithmétique d'intervalles.

On donne ici un exemple faisant intervenir des projections et des intersections. Un arc épais, d'une épaisseur donnée 2ρ , d'une courbe paramétrée définie par : $x = f(t) + [-\rho, \rho], 0 \leq t \leq 1$, est représenté par l'intersection de quatre primitives : une primitive pour $0 \leq t$, une autre pour $t \leq 1$, une troisième pour $x \leq f(t) + [-\rho, \rho]$ et une dernière pour $x \geq f(t) + [-\rho, \rho]$. L'intersection est traitée dans la section 2.3.

2.2. Représentation des unions $E = E_1 \cup E_2$

Supposons à présent que nous ayons deux objets géométriques : E_1 avec une variable e_1 et E_2 avec une variable e_2 . E_1 (resp. E_2) est décrit par un système d'équations $E_1(e_1, x, y, z, \dots) = 0$ (resp. $E_2(e_2, x, y, z, \dots) = 0$). Les points de E_1 (resp. E_2) sont entièrement caractérisés par $e_1 \leq 0$

(resp. $e_2 \leq 0$). L'objet $E = E_1 \cup E_2$ est décrit par une nouvelle variable e , et un système d'équations qui garantiront que $e = \min(e_1, e_2)$. Clairement $e = \min(e_1, e_2)$ est équivalent à maximiser e sous les contraintes $e \leq e_1, e \leq e_2 \Rightarrow e + \alpha_1^2 = e_1, e + \alpha_2^2 = e_2$ pour certains $\alpha_1, \alpha_2 \in \mathbb{R}$.

On considère alors le Lagrangien :

$$L : e + \lambda_1(e + \alpha_1^2 - e_1) + \lambda_2(e + \alpha_2^2 - e_2)$$

La solution vérifie ce système :

$$\begin{aligned} L'_e : 1 + \lambda_1 + \lambda_2 &= 0 & L'_{\alpha_1} : 2\lambda_1\alpha_1 &= 0 \\ L'_{\lambda_1} : e + \alpha_1^2 - e_1 &= 0 & L'_{\alpha_2} : 2\lambda_2\alpha_2 &= 0 \\ L'_{\lambda_2} : e + \alpha_2^2 - e_2 &= 0 & & \end{aligned}$$

En effet, lorsque $e_1 < e_2$, le système précédent admet pour solution $e = e_1, \alpha_1^2 = 0, \alpha_2^2 = e_2 - e_1, \lambda_1 = -1, \lambda_2 = 0$.

Cette solution est unique, excepté pour $\alpha_2 = \pm\sqrt{e_2 - e_1}$.

Par symétrie, lorsque $e_2 < e_1$, le système admet pour solution $e = e_2, \alpha_2^2 = 0, \alpha_1^2 = e_1 - e_2, \lambda_1 = 0, \lambda_2 = -1$.

Cette solution est unique, excepté pour $\alpha_1 = \pm\sqrt{e_1 - e_2}$.

Le système a les deux solutions lorsque $e_1 = e_2$.

Ainsi, l'ensemble $E = E_1 \cup E_2$ est décrit par la variable e ; son système d'équations est la concaténation du système pour E_1 , du système pour E_2 et du système équivalent à $e = \min(e_1, e_2)$.

Les points de E sont entièrement caractérisés par $e \leq 0$, ceux du complémentaire de E par $e \geq 0$ et ceux du bord de E par $e = 0$.

Pour le solveur par intervalles, on a besoin d'initialiser avec un intervalle englobant pour la variable e ; c'est $\min(e_1, e_2)$ qui est calculé par analyse d'intervalles à partir d'intervalles englobants pour e_1 et e_2 .

On rappelle que $\min([a, b], [a', b']) = [\min(a, a'), \min(b, b')]$.

Si l'intervalle englobant pour α_1^2 est $(e_1 - e) \cap [0, +\infty)$, calculé encore par intervalles, alors l'intervalle englobant pour α_1 est la racine carrée de l'intervalle pour α_1^2 . Par exemple, si $e_1 - e = [-6, 4]$, alors l'intervalle pour α_1^2 est $[-6, 4] \cap [0, +\infty) = [0, 4]$ et l'intervalle pour α_1 est $[0, 2]$.

On procède similairement pour initialiser les intervalles englobants pour α_2^2 et α_2 . Enfin, pour λ_1 et λ_2 , ils appartiennent tous les deux à l'intervalle $[-1, 0]$.

Une optimisation possible consiste à remplacer λ_2 par sa valeur $-1 - \lambda_1$ et à supprimer la première équation $L'_e = 0$.

On laisse aussi au lecteur une autre optimisation possible lorsque, dans la boîte étudiée, l'intervalle englobant pour e_1 est strictement plus petit (pour l'inclusion) que celui pour e_2 , ou vice-versa.

Supposons que λ_2 a été éliminée, on a alors que :

$$C(E) = C(E_1) + C(E_2) + 4.$$

On présente maintenant une autre représentation pour l'union. On considère e_1, e_2 comme des racines d'un polynôme de degré 2. Si Δ est le discriminant de ce polynôme, alors il vérifie $\Delta = (e_2 - e_1)^2$ et si $e = \min(e_1, e_2)$, alors $2e = (e_1 + e_2) - \sqrt{\Delta}$ (alors que si $e = \max(e_1, e_2)$, on a $2e = (e_1 + e_2) + \sqrt{\Delta}$).

Cela nous conduit au système suivant :

$$\begin{aligned} \delta^2 - (e_2 - e_1)^2 &= 0 \\ 2e - (e_1 + e_2) + \delta &= 0 \end{aligned}$$

avec δ qui vaut $\sqrt{\Delta}$, donc $\delta \in [0, \sqrt{\Delta}]$. Dans cette approche algébrique on a $C(E) = C(E_1) + C(E_2) + 2$, qui est plus efficace en nombre d'équations et d'inconnues introduites, alors qu'effectuer l'élimination de variables dans l'approche Lagrangienne peut aussi conduire à un système similaire.

2.3. La représentation des intersections $E = E_1 \cap E_2$

On remplace seulement \min dans le paragraphe précédent par \max , car $E = E_1 \cap E_2$ sera présenté par la variable $e = \max(e_1, e_2)$. Clairement, $e = \max(e_1, e_2)$ est équivalent à minimiser e sous les contraintes :

$$e_1 \leq e, e_2 \leq e \Rightarrow e_1 + \alpha_1^2 = e, e_2 + \alpha_2^2 = e$$

On considère donc le Lagrangien :

$$L = e + \lambda_1(e_1 + \alpha_1^2 - e) + \lambda_2(e_2 + \alpha_2^2 - e)$$

On sait que la solution vérifie ce système :

$$\begin{aligned} L'_e : \quad 1 - \lambda_1 - \lambda_2 &= 0 & L'_{\alpha_1} : \quad 2\lambda_1\alpha_1 &= 0 \\ L_{\lambda_1} : \quad e_1 + \alpha_1^2 - e &= 0 & L'_{\alpha_2} : \quad 2\lambda_2\alpha_2 &= 0 \\ L_{\lambda_2} : \quad e_2 + \alpha_2^2 - e &= 0 \end{aligned}$$

Lorsque $e_1 < e_2$, le système précédent a pour solution $e = e_2, \alpha_1^2 = e_2 - e_1, \alpha_2 = 0, \lambda_1 = 0, \lambda_2 = 1$.

Symétriquement, lorsque $e_2 < e_1$, le système précédent a pour solution $e = e_1, \alpha_1^2 = 0, \alpha_2^2 = e_1 - e_2, \lambda_1 = 1, \lambda_2 = 0$.

Le système a les deux solutions lorsque $e_1 = e_2$.

Ainsi, l'ensemble $E = E_1 \cap E_2$ est décrit par la variable e ; son système d'équations est la concaténation du système pour E_1 , du système pour E_2 et du système équivalent à $e = \max(e_1, e_2)$. Les points de E sont entièrement caractérisés par $e \leq 0$, ceux du complémentaire de E par $e \geq 0$ et ceux du bord de E par $e = 0$.

L'initialisation des intervalles pour $e, \alpha_1, \alpha_2, \lambda_1, \lambda_2$ est très similaire au cas précédent de l'union. L'intervalle pour

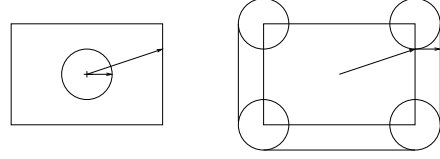


Figure 1: La somme de Minkowski d'un disque et d'un rectangle.

e est $\max(e_1, e_2)$ qui est calculé à partir des intervalles pour e_1 et e_2 . Rappelons que $\max([a, b], [a', b'])$ est l'intervalle $[\max(a, a'), \max(b, b')]$. L'intervalle pour α_1^2 est l'intervalle $(e - e_1) \cap [0, +\infty)$, etc. Cette fois, λ_1 et λ_2 appartiennent tous les deux à $[0, 1]$.

Optimisations : une première optimisation possible consiste à remplacer λ_2 par sa valeur $1 - \lambda_1$. Là encore, si les intervalles pour e_1 et e_2 sont disjoints, des optimisations et simplifications immédiates sont possibles et sont laissées au lecteur.

La deuxième formulation avec le discriminant diffère seulement d'un signe par rapport à l'union : on remplace uniquement la seconde équation par

$$2e - (e_1 + e_2) - \delta = 0.$$

2.4. La représentation du complémentaire

Soit E_1 un ensemble défini par le système d'équations $E_1(e_1, X) = 0$ et qui a pour variable caractéristique e_1 . Pour définir le complémentaire \bar{E}_1 de l'ensemble E_1 , on introduit une nouvelle variable \bar{e}_1 , qui sera la variable caractéristique de \bar{E}_1 , et le système d'équations définissant \bar{E}_1 est la concaténation du système $E_1(e_1, X) = 0$ et de la nouvelle équation : $e_1 + \bar{e}_1 = 0$.

Il vient que : $C(\bar{E}_1) = C(E_1) + 1$.

Variante : remplacer e_1 dans la définition de E_1 par $-\bar{e}_1$ pour obtenir la définition de \bar{E}_1 . Dans ce cas, on a : $C(\bar{E}_1) = C(E_1)$.

2.5. La représentation des différences $E = E_1 - E_2$

La représentation de $E = E_1 - E_2$ est laissée au lecteur : $E = E_1 - E_2$ est simplement $E_1 \cap (\bar{E}_2)$. Pour l'approche Lagrangienne, on a que :

$$C(E) = C(E_1) + C(\bar{E}_2) + 4 = C(E_1) + C(E_2) + 5$$

alors que pour l'approche algébrique, on a que :

$$C(E) = C(E_1) + C(E_2) + 3.$$

3. Représentation d'autres objets

3.1. La somme de Minkowski

La somme de Minkowski de deux ensembles en 3 dimensions E_1 et E_2 est l'ensemble des points :

$$E = \{(x, y, z) = (x_1 + x_2, y_1 + y_2, z_1 + z_2) \mid (x_1, y_1, z_1) \in E_1, (x_2, y_2, z_2) \in E_2\}$$

Lorsque E_1 est défini par la variable e_1 et le système d'équations E_1 , de même pour E_2 , on a que l'ensemble $E = E_1 \oplus E_2$ est défini par une nouvelle variable e , la concaténation des systèmes pour $X_1 \in E_1$, pour $X_2 \in E_2$ et $x_1 + x_2 - x = y_1 + y_2 - y = z_1 + z_2 - z = 0$. De plus : $e = \min(e_1 + e_2)$.

Avec cette définition, la frontière de E est caractérisée par $e = 0$; les points strictement à l'intérieur (resp. strictement à l'extérieur) de E sont caractérisés par $e < 0$ (resp. $e > 0$).

On présente maintenant le système correspondant au problème de minimisation. Soit E_0 la somme de Minkowski de deux ensembles géométriques E_1 et E_2 en dimension d , à savoir :

$$\mathbf{z} \in E_0 \iff \exists \mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{R}^d : \mathbf{z} = \mathbf{x} + \mathbf{y} \wedge \mathbf{x} \in E_1 \wedge \mathbf{y} \in E_2.$$

Soit \mathcal{S}_i l'ensemble ordonné des équations utilisées dans la description de l'ensemble géométrique E_i , $i = 0, 1, 2$. Soit $f_k(\mathcal{S}_i)$ la k -ième équation de \mathcal{S}_i .

Soit \mathcal{V}_i l'ensemble des variables dans \mathcal{S}_i , $i = 0, 1, 2$. En général, $|\mathcal{V}_i| = |\mathcal{S}_i| + d$. Soit $s_i \in \mathcal{V}_i$ la variable d'écart de l'ensemble E_i . On veut minimiser $s_1 + s_2$. Pour cela, on doit utiliser les conditions de Fritz John avec une condition de normalisation. Les conditions de Fritz John donnent des contraintes d'inégalités et traitent aussi le cas où les contraintes sont linéairement dépendantes au minimum. La condition de normalisation fournit des bornes simples pour les multiplicateurs. †

Les conditions de John peuvent être générées à l'aide de la quantité suivante J , similaire à un Lagrangien :

$$J := \mu_0(s_1 + s_2) + \mu_1 s_1 + \mu_2 s_2 + \sum_{i=1}^{|\mathcal{S}_1|} u_i f_i(\mathcal{S}_1) + \sum_{i=1}^{|\mathcal{S}_2|} v_i f_i(\mathcal{S}_2) + \mathbf{w} \cdot (\mathbf{z} - \mathbf{x} - \mathbf{y})$$

Maintenant que \mathcal{S}_0 contient $\frac{\partial J}{\partial a}$, $\forall a \in \mathcal{V}_i, i = 1, 2$

($|\mathcal{S}_1| + |\mathcal{S}_2| + 2d$ équations en général), aussi bien que les équations suivantes :

$$\begin{aligned} z_i - x_i - y_i &= 0, & i = 1, \dots, d \\ \mu_i s_i &= 0, & i = 1, 2 \\ f_i(\mathcal{S}_1) &= 0, & i = 1, \dots, |\mathcal{S}_1| \\ f_i(\mathcal{S}_2) &= 0, & i = 1, \dots, |\mathcal{S}_2| \end{aligned}$$

†. On doit aussi vérifier que les inégalités s'appliquent ! Dans notre cas, cela est géré par les bornes dans s_i .

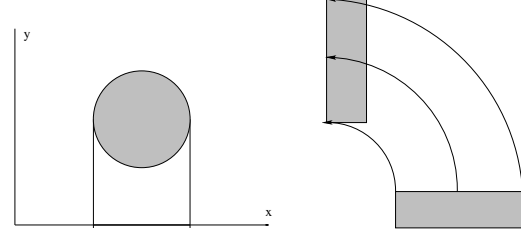


Figure 2: À gauche : une projection d'un disque. À droite : une extrusion d'un rectangle.

$$\begin{aligned} \mu_0 + \mu_1 + \mu_2 + \sum_{i=1}^d w_i^2 + \sum_{i=1}^{|\mathcal{S}_1|} u_i^2 + \sum_{i=1}^{|\mathcal{S}_2|} v_i^2 &= 1, & \mu_i \geq 0 \\ s_0 - s_1 - s_2 &= 0, \end{aligned}$$

avec la condition de normalisation impliquent que $\mu_i \in [0, 1]$ et $w_i, u_i, v_i \in [-1, 1]$. Il s'ensuit également que : $C(E_0) = 2C(E_1) + 2C(E_2) + 3d + 4$.

3.2. La projection d'un ensemble

On considère uniquement l'exemple suivant, on laisse la généralisation au lecteur. Soit E l'ensemble en 3 dimensions : $\{(x, y, z) \mid E(x, y, z, e) = 0, e \leq 0\}$. On considère l'ensemble E_p qui est la projection orthogonale de E sur la plan Oxy . Autrement dit : $E_p = \{(x, y) \mid \exists z, E(x, y, z, e) = 0, e \leq 0\}$. Cela peut être formulé comme un problème de minimisation : (x, y) appartient à E_p si pour le z qui minimise e pour ce couple (x, y) , on a que e est négatif ou nul.

Ainsi le système pour E_p est la concaténation du système qui définit E , et du système pour : $e_p = \min e$; la variable e_p est la variable qui caractérise E_p ; notons que x, y sont des paramètres "donnés" (ce sont des coordonnées dans Y), et la minimisation est faite sur z .

Cette projection "oublie" la coordonnée z ; pour les ensembles paramétrés par des paramètres u_1, u_2, \dots, u_p , on veut "oublier" tous les paramètres u_i ; la généralisation est laissée au lecteur.

3.3. L'extrusion

La figure 2 illustre une extrusion d'un rectangle en 2 dimensions.

Soit E un ensemble défini par le système $E(e, X) = 0$ et la variable e .

Soit $M(t), 0 \leq t \leq 1$ la matrice d'une transformation affine, au temps t . Par exemple, pour tourner autour de l'axe z :

$$M(t) = \begin{pmatrix} \cos(2\pi t) & \sin(2\pi t) & 0 & 0 \\ -\sin(2\pi t) & \cos(2\pi t) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

E est extrudé, ou balayé, dans $M(t)$. Cela donne l'ensemble E' . Les points x', y', z' de E' sont définis par $(x', y', z', 1) = (x, y, z, 1)M(t)$.

La variable e' caractérisant E' est min e telle que $(x', y', z', 1) = (x, y, z, 1)M(t)$.

Ici, E s'est déplacé essentiellement le long d'une courbe (le paramètre de temps étant t dans $[0, 1]$). Il est possible de faire déplacer E le long d'un carreau de surface, paramétré par deux paramètres u et v , ou dans un volume paramétré par trois paramètres u , v et w . Il suffit de concaténer la définition de cet espace paramétré pour obtenir la définition complète de E' .

4. Implémentation

En utilisant SAGE [S*12], nous avons implémenté la modélisation d'ensembles géométriques proposée. L'avantage de SAGE vient de l'environnement de programmation qu'il fournit à travers le langage *Python*, ainsi que des interfaces pour différentes bibliothèques pour le calcul scientifique. Nous avons programmé la génération des systèmes algébriques nécessaires qui sont ensuite résolus (plus précisément, on calcule une couverture de l'espace des solutions pour une résolution donnée) en utilisant :

- (a) un solveur naïf qui recherche les solutions dans \mathbb{R}^d
- (b) Quimper [CJ09]
- (c) un solveur utilisant l'algorithme de Newton étendu aux intervalles, programmé en C++ [Kub12]

Le solveur (a) est utilisé pour le prototypage et est programmé directement dans SAGE. Nous avons programmé quatre variantes : (a_0) recherche naïve dans \mathbb{R}^d , (a_1) recherche utilisant une contraction algébrique triviale, (a_2) subdivision de l'espace des solutions couplée à une contraction des intervalles, (a_3) exploitation des dépendances entre variables qui sont données *a priori*.

Les solveurs (b) et (c) sont utilisés via des programmes externes, appelé depuis le script. Nous pouvons également effectuer une analyse pour la décomposition DM [AaJM93] en utilisant un graphe bipartite (graphe des variables [WBL09]) que nous pouvons visualiser (nous souhaitons explorer l'utilisation de ce type de techniques dans de futures travaux).

Nous avons effectué quelques tests de performance avec les primitives géométriques suivantes :

ensemble	équation	ensemble	$x \times y$
E_1	$\frac{1}{5}x^2 + y^2 < 1$	ellipse	$[-2, 2] \times [-2, 2]$
E_2	$(x - \frac{3}{4})^2 + (y - \frac{3}{4})^2 < \frac{2}{5}$	disque	$[-2, 2] \times [-2, 2]$
E_3	$\frac{1}{5}x^2 + y^2 < \frac{2}{5}$	ellipse	$[-2, 2] \times [-3, 1]$
E_4	$x^2 + y^2 < \frac{1}{5}$	disque	$[-2, 2] \times [-1, 1]$
E_5	$\frac{1}{5}x^2 + (y + \frac{1}{5})^2 < \frac{2}{5}$	disque	$[-2, 2] \times [-2, 2]$

Ces primitives sont utilisés pour décrire les ensembles géométriques suivants :

	ensemble	description	taille du sys.
(i)	E_1	primitive simple	1×3
(ii)	$\overline{E_1}$	complément	2×4
(iii)	$E_1 \cap E_2$	intersection (Lagrangien)	6×8
(iv)	$E_1 \cap E_2$	intersection (nombre algébrique)	4×6
(v)	$(E_3 - E_4) \cap E_5$	ensemble plus complexe	8×10
(vi)	$E_3 \oplus E_4$	Somme de Minkowski	14×16

qui sont résolus en utilisant les solveurs (a), (b) et (c) comme l'indique les figures 3, 4 et 5. les deux lignes du dessus montre la couverture de l'ensemble avec une précision de 0.1, calculé par les solveurs a_0 , a_1 , a_2 , a_3 , b , c dans cet ordre. Par commodité, seul l'espace visible (x, y) est affiché. Cependant si la dimension est plus élevée, la somme de Minkowski de deux primitives fait intervenir 16 variables, comparé à seulement 3 pour une primitive simple. En rouge, il s'agit des boîtes non-retenues, en bleu les boîtes retenues et en vert les boîtes pour lesquelles il y a au moins une solution. Ceci est implémenté uniquement dans le solveur (a) en utilisant le test de Miranda [Vra89], mais nous souhaitons l'améliorer de façon à ce que toutes les boîtes intérieures soient vertes et seules les boîtes de la frontière soient bleues. La ligne suivante affiche le temps d'exécution avec différentes précisions. Les barres de chaque groupe correspondent aux six solveurs testés. Sur la ligne du bas, à gauche, nous affichons le graphe bipartite correspondant [AaJM93] : un sommet est créée pour chaque équation et chaque variables, une arête joint une équation à une variable si et seulement si l'équation dépend de cette variable. À droite on affiche le graphe des variables [WBL09] : chaque sommet représente une variable et deux sommets sont connectés si et seulement si les variables apparaissent toutes les deux dans une même équation.

Nous avons fixé le temps de calcul maximum à dix minutes. On observe les phénomènes suivant sur les tests.

Pour les ensembles (i) et (ii) (primitives simples et leurs compléments), les solveurs (b) et (c) sont très efficaces. Quand une grande précision est requise, (c) semble plus efficace que (b). Ils sont suivis de la variante (a_3) qui en fait fixe « en dur » les dépendance des variables d'écart sur l'espace visible (x, y) dans nos exemples). La différence pour les temps d'exécutions vient de l'implémentation en *Python* de (a_3) alors que les deux autres sont programmé en C++.

Pour (iii) et (iv) (intersections), le nombre d'inconnues augmente. Pour (iv), on remarque la performance du solveurs (b) comparé à ces compétiteurs. Pour (iii), qui utilise l'approche lagrangienne, le temps d'exécution de (b) augmente d'un ordre de grandeur, tandis que le nombres d'inconnues passent de 6 à 8. On observe également que la recherche naïve dans l'algorithme (a_0) échoue, tandis que (a_2)

et (a_3) se comporte plutôt bien et (a_3) semble être efficace. Cela montre qu'il vaut mieux se focaliser sur l'espace visible ou la dépendance entre variables. Cependant la version actuelle de (a_2) est encore expérimentale, et peut ne pas rejeter les boîtes vides dans le cas du calcul de la somme de Minkowski.

Pour l'ensemble complexe (v) , de nouveau (b) est le plus efficace. Le solveur (c) échoue pour la première fois avec 10 inconnues.

Enfin, pour la somme de Minkowski (vi) , seul (b) semble donner des résultats. (a_2) se comporte plutôt bien, mais garde trop de boîtes.

Le solveur (c) semble prometteur mais doit être étendu afin d'exploiter les dépendances entre variables pour rester efficace en haute dimension.

Enfin, nous souhaitons mentionner que nous n'avons pas traité un problème tout aussi important : tester si ensemble est vide ou non. Pour cela on peut :

- Modifier les solveurs existants dès que l'on rencontre une solution.
- Augmenter le système sous-contraint de façon à le rendre bien-contraint et résoudre le système obtenu. On peut par exemple calculer un point dans l'espace des solutions qui soit le plus proche de l'origine en résolvant un problème d'optimisation.

5. Conclusion

La méthode DJC initialement introduite s'applique uniquement pour les combinaisons d'opérations booléennes d'ensembles défini par des primitives. Cette nouvelle représentation des ensembles géométriques permet d'étendre facilement de façon significative la portée de la méthode DJC : elle s'applique maintenant pour chaque combinaison d'opérations booléennes, projections, somme de Minkowski, extrusions, déformations. La nouvelle représentation des ensembles géométriques permet probablement d'étendre la portée d'autres algorithmes géométriques. Sa principale caractéristique est de fournir un langage spécifiant les contraintes portant sur des ensembles géométriques, et d'obtenir des caractéristiques sur des ensembles géométriques. L'analyse par intervalles est utilisée pour résoudre les systèmes d'équations générés automatiquement. En effet, avec cette nouvelle représentation, il semble que de nombreux problèmes géométriques se ramènent à la résolution de systèmes d'équations ou de problèmes d'optimisation.

On conclut par des questions d'ouverture et des travaux futurs.

Les implémentations pour les solveurs par intervalles.

Tout d'abord, les exemples sont en accord avec la théorie et confirment que la modélisation par les systèmes algébriques proposés est correct. Les critères actuels suggèrent

qu'il pourrait être bénéfique d'exploiter les dépendances entre les variables.

Ainsi, les travaux futures incluent :

- L'utilisation des solveurs de Bernstein dans l'espoir qu'ils produiront des boîtes de taille plus efficace. Dans cette direction, on a déjà implémenté un analyseur pour les polynômes multivariés pour être utilisé avec [Dje12] (qui concerne un solveur LP (linear programming) efficace pour des problèmes creux) pour une implémentation efficace de [FMF10].
- Les techniques appliquées dans [AaJM93] et [WBL09] dans le but d'exploiter les dépendances entre les variables et le fait que les systèmes générés sont assez creux (chaque équation contient peu d'inconnues, comparé au nombre total d'inconnues). Dans [WBL09], une technique de partitionnement a été utilisée avec succès pour s'attaquer aux problèmes de cryptanalyse.
- L'implémentation d'un test pour savoir si l'ensemble est vide. Cela peut être fait comme indiqué par l'arrêt à la première solution rencontrée, ou en résolvant un système bien contraint approprié (cf. Sec. 4).
- L'implémentation de tests de dérivées secondes pour des problèmes d'optimisation génériques (ce n'est pas requis pour le moment).

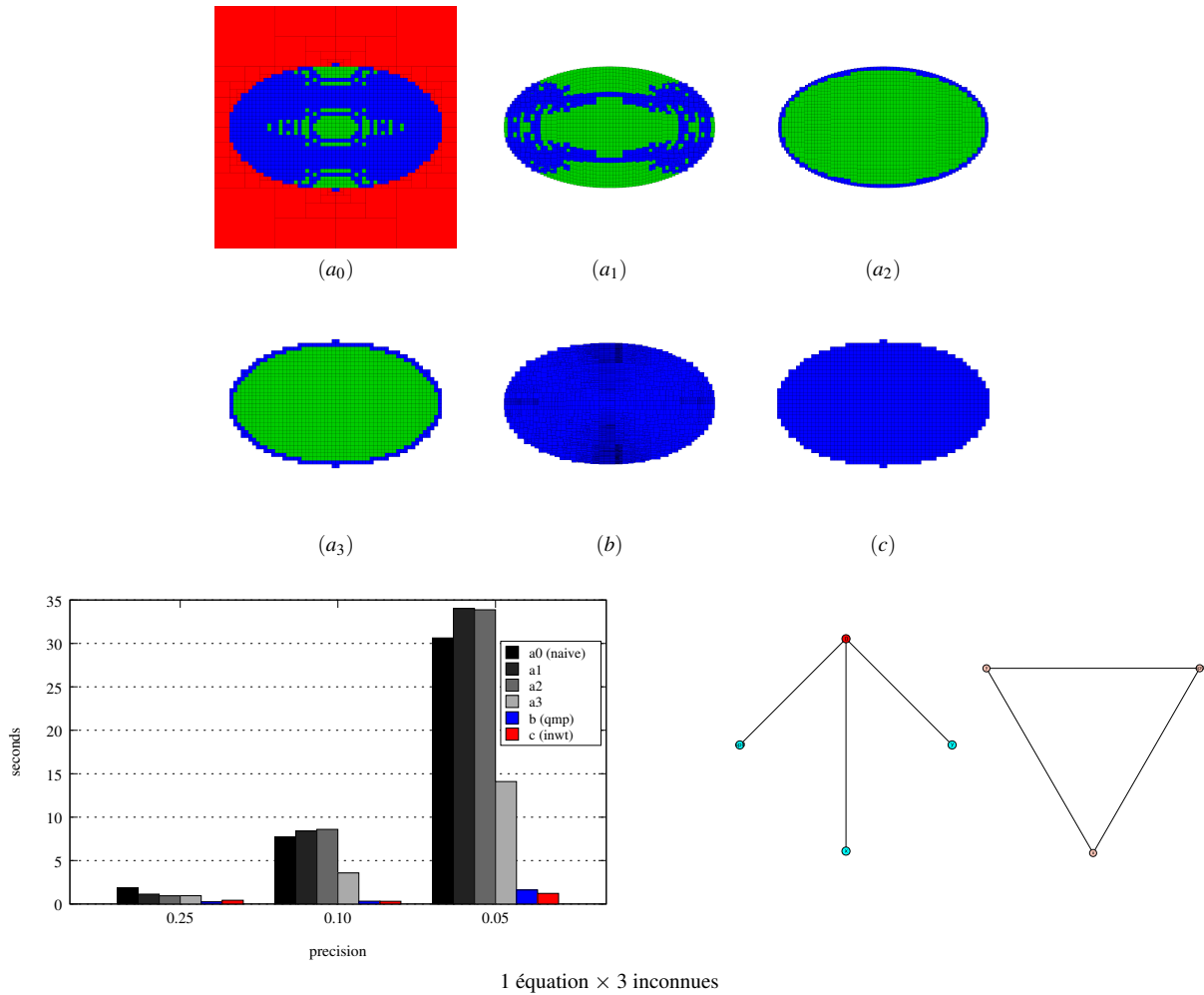
Améliorations de la méthode DJC.

La méthode DJC fournit un complexe simplicial homotope à l'ensemble étudié. La distance de Hausdorff entre l'ensemble étudié et le complexe simplicial peut être grande ; par exemple, un complexe simplicial constitué d'un seul sommet est homotope à toute boule (topologique). Il semble possible de modifier la méthode DJC, pour tenir compte de certains seuils sur la distance de Hausdorff. Une autre question possible est : est-il possible d'obtenir un complexe simplicial isotope (au lieu d'homotope seulement) à l'ensemble étudié ?

Est-il possible de relaxer les contraintes d'un « ensemble épais » ? Par exemple un carré dans le plan, moins un cercle (qui n'est pas un « ensemble épais ») dans le carré, est un « ensemble épais » avec deux composantes (l'intérieur du disque, et l'extérieur). En outre, est-il possible de tenir compte des morceaux de frontières : des arcs paramétrés de courbes 2D en 2D, ou des carreaux de surfaces paramétrés en 3D ? On aimerait étudier la topologie imposée par un ensemble de morceaux de frontières : par exemple, est-ce qu'ils délimitent un bord ou non ? Cela pourrait être utilisé pour prouver qu'une B-rep (représentation par les bords) donnée est valide.

Acknowledgements

The authors want to thank Vishal Donderia for useful discussions during his internship. This research work, especially Vishal

Figure 3: E_1

Donderia's internship, George Tzoumas' current postdoc in Dijon and Jean-Marc Cane's PhD have been funded by NPRP grant number NPRP 09-906-1-137 from the Qatar National Research Fund (a member of The Qatar Foundation).

Références

- [AaJM93] AIT-AOUDIA S., JEGOU R., MICHELUCCI D. : Reduction of constraint systems. In *Compugraphics* (Alvor, Portugal, 1993), pp. 83–92.
- [CJ09] CHABERT G., JAULIN L. : Contractor programming. *Artificial Intelligence. Vol. 173*, Num. 11 (2009), 1079 – 1100.
- [Col75] COLLINS G. E. : Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In *Proc. 2nd GI Conference on Automata Theory and Formal Languages* (1975), vol. 33 de *Lecture Notes Comput. Sci.*, Springer-Verlag, pp. 134–183.
- [DJC07] DELANOUE N., JAULIN L., COTTENCEAU B. : Guaranteeing the homotopy type of a set defined by non-linear inequalities. *Reliable computing. Vol. 13*, Num. 5 (2007), 381–398.

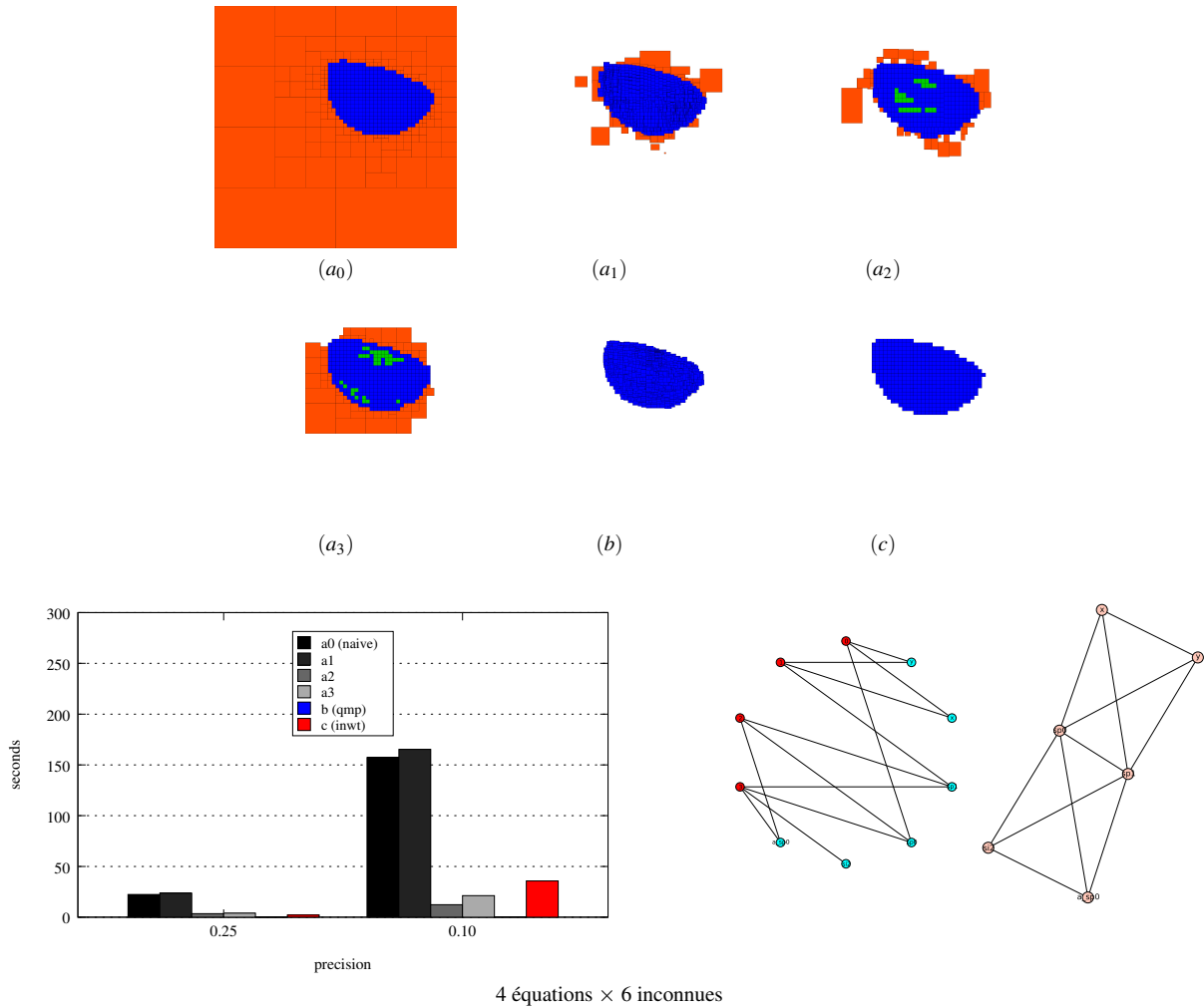
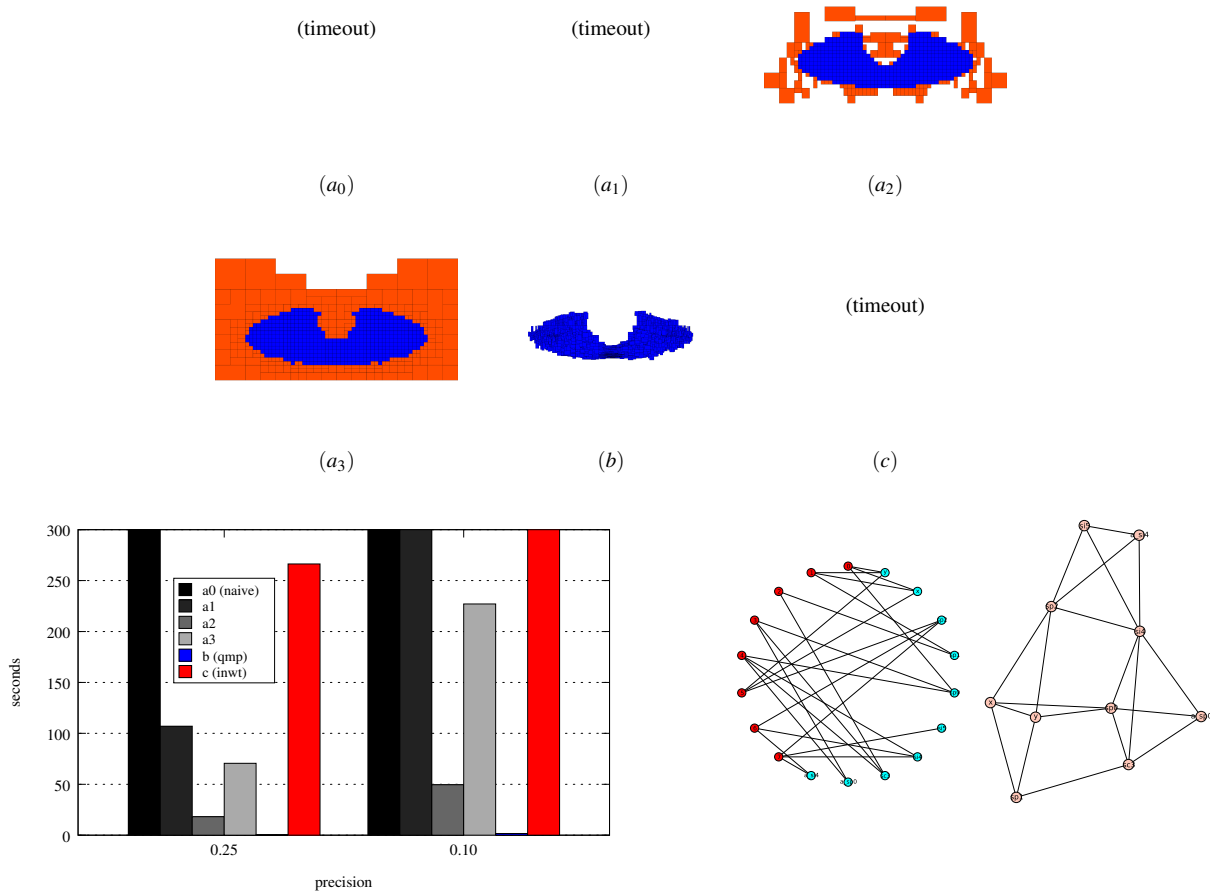


Figure 4: $E_1 \cap E_2$ (système algébrique)

- [Dje12] DJEDAINI M. : *Internship report*. Tech. rep., Qatar University, 2012.
- [FMF10] FÜNFIG C., MICHELUCCI D., FOUFOU S. : Optimizations for tensorial Bernstein-based solvers by using polyhedral bounds. *International Journal of Shape Modeling*. Vol. 16, Num. 01n02 (2010), 109–128.
- [Kub12] KUBICKY A. : *Interval Newton solver in C++*. University of Burgundy, 2012.
- [RT78] REQUICHA A. A. G., TILOVE R. B. : *Mathematical Foundations of Constructive Solid Geometry : General Topology of Closed Regular Sets*. Tech. rep., UR Research [http://dspace.lib.rochester.edu/oai/request] (United States), 1978.
- [S*12] STEIN W., ET AL. : *Sage Mathematics Software (Version 5.0.1)*. The Sage Development Team, 2012. http://www.sagemath.org.

- [SH97] STANDER B. T., HART J. C. : Guaranteeing the topology of an implicit surface polygonization for interactive modeling. pp. 279–286.
- [Vra89] VRAHATIS M. N. : A Short Proof and a Generalization of Miranda’s Existence Theorem. *Proceedings of the American Mathematical Society*. Vol. 107, Num. 3 (1989), pp. 701–703.
- [WBL09] WONG K. K.-H., BARD G. V., LEWIS R. H. : Partitioning multivariate polynomial equations via vertex separators for algebraic cryptanalysis and mathematical applications. Cryptology ePrint Archive, Report 2009/343, 2009. http://eprint.iacr.org/.



8 équations × 10 inconnues

Figure 5: $(E_3 - E_4) \cap E_5$