HAL
open science

Approcher des courbes par des hélices

Florence Bertails-Descoubes, Alexandre Derouet-Jourdan

To cite this version:

Florence Bertails-Descoubes, Alexandre Derouet-Jourdan. Approcher des courbes par des hélices. Quadrature, 2015, 96. hal-01224893

HAL Id: hal-01224893

https://hal.science/hal-01224893v1

Submitted on 25 Nov 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Approcher des courbes par des hélices

Florence Bertails-Descoubes* et Alexandre Derouet-Jourdan

INRIA et Laboratoire Jean Kuntzman, Grenoble, France

Paru dans Quadrature, avril 2015

Résumé

La forme élancée des structures fibreuses - telles que les cheveux, les brins d'ADN ou encore les tiges de plantes - peut se représenter à l'aide de courbes de l'espace. En particulier, les courbes en hélices par morceaux offrent une représentation intéressante pour modéliser numériquement de telles données, car elle s'avère très compacte : 2 nombres (une courbure et une torsion) suffisent à caractériser un morceau d'hélice circulaire. Dans cet article, nous présentons une méthode à la fois précise et rapide pour approcher une courbe quelconque de l'espace par une courbe lisse en hélices par morceaux.

1 Des hélices dans notre quotidien

Il est assez étonnant de constater à quel point les structures filiformes, c'est-à-dire les objets longs et très fins, font partie de notre environnement quotidien : les exemples biologiques naturels (branches d'arbre, brins d'herbe, cheveux, etc.) rivalisent en nombre avec les objets issus de l'industrie (fibres optiques, cordons de téléphone, câbles d'ordinateur, etc.). Et l'on a beau zoomer et dézoomer notre angle de vue, on retrouve encore ces mêmes structures longilignes ou fibreuses, aussi bien en descendant dans les petites échelles (fibres musculaires du cœur, nano-tubes de carbone, protéines ou encore ADN) qu'en remontant vers les grandes échelles (structure chevelue du champ

*florence.descoubes@inria.fr

magnétique d'une étoile, filaments galactiques). Par ailleurs, quelle que soit l'échelle considérée, on peut remarquer la tendance spontanée qu'ont la plupart de ces structures à s'enrouler, s'enchevêtrer ou se vriller, pour dessiner des formes tridimensionnelles des plus élémentaires, mais aussi des plus esthétiques : des *hélices*. La double hélice de l'ADN, bien sûr. Mais aussi les cheveux ondulés ou bouclés, les vrilles de vigne entortillées, les cordons téléphoniques spiralés, ou encore les tornades et courants hélicoïdaux dans les rivières. Quelques exemples de formes hélicoïdales réelles sont présentés en Figure 1.

FIGURE 1 – Exemples d'hélices réelles. De gauche à droite : vrille de vigne - ©J. Sullivan, pdphoto.org, mèche de cheveux bouclés - ©F. Bertails-Descoubes, et sculpture de Charles Jencks aux Kew Gardens de Londres, représentant la double hélice de l'ADN - ©J.-P. Truc.

2 Acquisition et traitement numérique

Si les artistes célèbrent par la peinture ou la sculpture la grâce des objets filiformes et hélicoïdaux depuis l'antiquité, les scientifiques - non moins sensibles à cette grâce - cherchent davantage à observer, analyser et modéliser mathématiquement ces structures pour en comprendre l'origine et en prédire la formation. Pour saisir la structure des objets étudiés, à deux ou trois dimensions, de nombreuses techniques d'acquisition ont été développées au cours des soixante dernières années, depuis la cristallographie révélant la

structure hélicoïdale de l'ADN jusqu'à l'IRM mettant en lumière la structure fibreuse de la matière blanche du cerveau, en passant par la capture multi-caméras sous lumière contrôlée permettant d'acquérir précisément la géométrie d'une chevelure humaine réelle. Ainsi, aujourd'hui, un grand nombre de données complexes sous forme d'images ou de géométrie brute (comme des nuages de points) sont disponibles dans tous les champs de la science, notamment en biologie, médecine, astrophysique, ou en informatique de l'image (vision par ordinateur, informatique graphique). Un traitement numérique est alors indispensable afin de trier, de simplifier, d'analyser, et de comprendre ces données.

La modélisation géométrique est l'une des étapes importantes de l'interprétation des données d'entrée : elle consiste à représenter les structures observées par des primitives géométriques élémentaires (comme des courbes ou des surfaces paramétrées) et permet ainsi de simplifier et de rationaliser les formes de la vie réelle. Dans notre cas, où nous nous intéressons à des structures filiformes plus ou moins bouclées, ce sont les courbes, et plus précisément les courbes *constituées d'hélices*, qui apparaissent particulièrement pertinentes pour représenter nos données d'entrée. Nous allons alors nous intéresser au problème suivant : comment approcher une courbe quelconque de l'espace par une courbe en hélices par morceaux ?

3 Approximation en hélices par morceaux

Voilà donc notre problème. En entrée, nous considérerons une courbe spatiale, aussi tordue que l'on veut. Cette courbe pourra provenir d'un travail de modélisation géométrique préalable (par exemple, sculpture virtuelle de cheveux par un artiste), ou au contraire être issue d'une capture automatique sur un sujet réel (par exemple, capture et reconstruction 3d d'une fibre à partir de photographies). Le format d'entrée n'aura que peu d'importance : il pourra s'agir d'une séquence de points 3D, ou d'une spline, ou autre. Par contre nous considérerons que l'information est *complète* (pas de trou dans les données) et *exacte* (pas de bruit dans les données). En sortie, notre objectif sera de générer une courbe en hélice par morceaux. Par hélice, nous entendrons en fait "hélice circulaire", c'est-à-dire une hélice à base circulaire et à pas constant (voir Figure 2). Nous nous imposerons une contrainte supplémentaire : les arcs d'hélice devront se recoller de manière lisse, c'est à dire de telle sorte que les tangentes coïncident aux points de jonction.

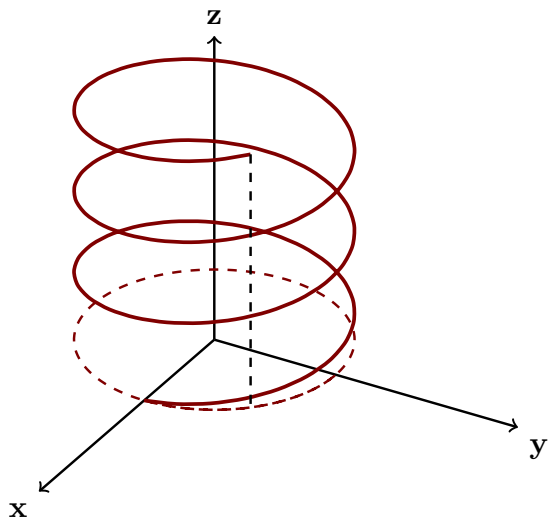

FIGURE 2 – Hélice circulaire à pas constant. Ici son axe est l'axe **z**. Projeté dans le plan (**xy**), l'hélice décrit un cercle (représenté par une ligne discontinue).

Ainsi, notre démarche est celle d'une approximation (ou conversion) géométrique, dans le sens où nous cherchons à faire coïncider "au mieux" des arcs d'hélices avec la courbe d'entrée.

4 Nombre minimal de morceaux ou erreur maximale ?

Dans un problème d'approximation géométrique, où l'on approche une donnée d'entrée (ici, une courbe de l'espace) par un nombre fini de primitives géométriques élémentaires (ici, des arcs d'hélice circulaire), il est légitime de se poser la question suivante : souhaite-t-on contrôler le nombre de morceaux de primitives que l'on utilise pour approcher au mieux la donnée d'entrée, ou bien l'erreur d'approximation commise ? Évidemment, ces deux quantités sont totalement inter-dépendantes, et le minimum que l'on puisse exiger d'une bonne méthode d'approximation est que l'erreur d'approximation soit une fonction strictement décroissante du nombre de primitives utilisées. Autrement dit, plus l'on utilise un nombre élevé de primitives, plus l'on se doit d'être précis. Dans le cas où l'on souhaite garantir une erreur d'ap-

proximation minimale entre la forme d'entrée et la forme approchée, il est souhaitable de pouvoir contrôler directement l'erreur causée par la méthode d'approximation, sans se préoccuper du nombre de primitives qu'il est alors nécessaire d'utiliser pour atteindre la précision requise (quitte à avoir besoin d'un nombre gigantesque de primitives). À l'inverse, si l'on considère que l'utilisation d'une primitive a un coût (par exemple, un coût en mémoire de stockage), il est sans doute préférable de contrôler directement le nombre de primitives employées, et en fonction de la quantité de ressources que l'on veut bien s'accorder, tâcher d'approcher au mieux la forme d'entrée (quitte à rester très loin de la forme d'entrée, faute d'un nombre suffisant de ressources). Ce choix du *paramètre* que l'on laisse à l'utilisateur est important car il conditionne la façon dont on va construire l'*algorithme* d'approximation, c'est-à-dire la méthode pour calculer automatiquement une approximation de la courbe d'entrée en une hélice par morceaux.

Dans cet article nous opterons pour la seconde stratégie. Plus précisément, notre algorithme d'approximation prendra comme paramètre d'entrée le nombre N d'arcs d'hélice souhaités, et produira en sortie une courbe $\mathcal{C}^1$ constituée de N arcs d'hélice de longueur égale, approchant la courbe d'entrée avec une erreur ϵ correspondant à une certaine notion de distance (que nous préciserons plus tard) entre la courbe d'origine et la courbe approchée.

Pour simplifier, nous commencerons par découper la courbe d'entrée en N morceaux de longueur égale. Nous appellerons *points de jonction* et noterons $\mathbf{p}_i$ les $N+1$ points de la courbe localisés à l'endroit de découpe. En chaque point $\mathbf{p}_i$, nous supposerons qu'il est possible de calculer ou d'estimer précisément la tangente à la courbe, notée $\mathbf{t}_i$. Une question naturelle qui se pose est alors la suivante : étant donnés deux points successifs $\mathbf{p}_i$ et $\mathbf{p}_{i+1}$ ainsi que leurs tangentes respectives $\mathbf{t}_i$ et $\mathbf{t}_{i+1}$, est-il possible de construire un arc d'hélice interpolant les deux points et les deux tangentes ? Y a-t-il toujours existence d'une telle hélice interpolante, ou uniquement dans certains cas ? Comment la construire en pratique ? Si une telle hélice existe dans une configuration donnée, est-elle unique ? Aussi élémentaires qu'elles y paraissent, ces questions semblent n'avoir été abordées en réalité que très récemment.

5 Interpolation et hélices : un problème peu étudié ?

Les questions d'interpolation pour les courbes et les surfaces se posent naturellement dans le domaine de la conception assistée par ordinateur – domaine qui a émergé dans les années 60 avec la conception informatique de modèles pour l'industrie automobile notamment – où l'on souhaite créer des formes respectant certaines contraintes imposées : par exemple, un câble doit relier un point A à un point B, ou bien la surface d'un modèle de carrosserie de voiture doit être suffisamment "lisse" pour refléter la lumière de manière esthétique, etc. Ces questions interviennent aussi dans d'autres champs comme la physique ou la biologie, où les formes géométriques élémentaires (cercles, ellipses, hélices) sont répandues et pour lesquelles on a besoin de comprendre la nature de leur formation ou de leur déformation dans le temps, en fonction des contraintes internes ou externes qui leur sont imposées. L'étude de l'existence et la caractérisation de ces formes élémentaires soumises à différents types de contraintes requièrent en général l'usage d'outils mathématiques classiques : ces questions n'intéressent donc que rarement les mathématiciens modernes, qui s'attèlent davantage à développer de nouvelles théories complexes. Pourtant, ces problèmes géométriques d'apparence simple sont loin d'être triviaux. Et encore une fois, ils trouvent bon nombre d'applications dans tout un champ de disciplines : conception par ordinateur, informatique graphique, modélisation en physique et en biologie. Il n'est ainsi pas rare de voir apparaître de temps en temps, en provenance d'informaticiens, de physiciens, de biologistes, ou de mathématiciens dits "appliqués", de nouveaux petits théorèmes qu'on aurait pu croire démontrés depuis l'antiquité, tant leur énoncé est simple (mais en mathématique, il faut toujours se méfier d'un énoncé trop simple).

6 À deux dimensions, le cas facile

Pour y voir plus clair, regardons tout d'abord le cas à deux dimensions. L'analogue d'un arc d'hélice circulaire (caractérisé par des courbure et torsion constantes) dans le plan est l'arc de cercle (caractérisé par une courbure constante). Notre problème d'interpolation initial revient ainsi à étudier les conditions d'existence et d'unicité d'un arc de cercle orienté interpolant deux points $\mathbf{p}_0$, $\mathbf{p}_1$ et deux vecteurs tangents $\mathbf{t}_0$, $\mathbf{t}_1$. Ces conditions seront par la

suite appelées conditions de *co-circularité* pour $\mathbf{p}_0$, $\mathbf{p}_1$, $\mathbf{t}_0$, et $\mathbf{t}_1$.

Plaçons-nous tout d'abord dans un cas encore plus simple et moins contraint, en omettant le second vecteur tangent $\mathbf{t}_1$. Comment construire un arc de cercle partant de $\mathbf{p}_0$ avec comme vecteur tangent $\mathbf{t}_0$, et passant par $\mathbf{p}_1$? Nous voyons rapidement qu'un tel arc orienté existe : il s'obtient en construisant le centre du cercle interpolant comme point d'intersection entre la médiatrice du segment $[\mathbf{p}_0,\mathbf{p}_1]$ et la droite perpendiculaire à $\mathbf{t}_0$, comme indiqué sur la Figure 3.

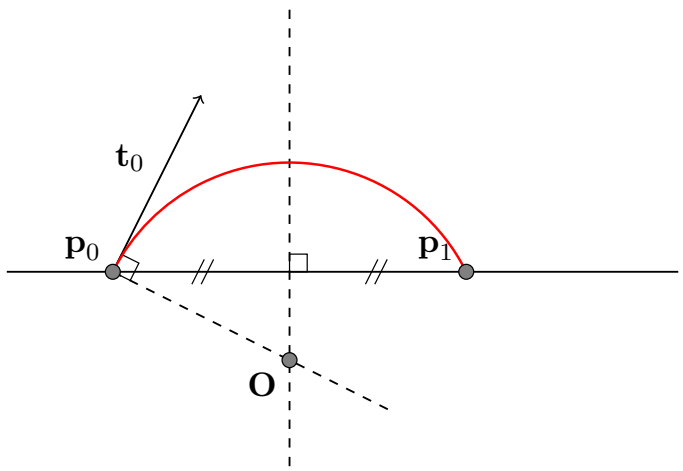

FIGURE 3 – Construction de l'arc de cercle orienté joignant $\mathbf{p}_0$ à $\mathbf{p}_1$ et ayant $\mathbf{t}_0$ pour vecteur tangent en $\mathbf{p}_0$.

D'après la contrainte d'interpolation du vecteur tangent $\mathbf{t}_0$ au point $\mathbf{p}_0$, l'arc que l'on recherche est en réalité *orienté*. De plus, si l'on se limite à un arc *court*, c'est-à-dire ne faisant pas un tour complet, l'arc interpolant $\mathbf{p}_0$, $\mathbf{t}_0$ et $\mathbf{p}_1$ est *unique*.

Considérons maintenant la contrainte que nous avions laissée de côté, à savoir l'interpolation du vecteur tangent $\mathbf{t}_1$ au point $\mathbf{p}_1$. Evidemment, vu ce que nous venons d'établir précédemment, cette nouvelle contrainte ne peut pas être satisfaite dans le cas général : le vecteur tangent $\mathbf{t}_1$ ne peut pas être fixé de manière arbitraire si l'on veut pouvoir l'interpoler.

Dans le cas général, on peut en fait formuler le petit théorème suivant qui donne un critère simple pour caractériser la co-circularité de deux points et deux vecteurs tangents :

Soient deux points $\mathbf{p}_0$ et $\mathbf{p}_1$ de $\mathbb{R}^2$ tels que $\mathbf{p}_0 \neq \mathbf{p}_1$, et deux vecteurs unitaires $\mathbf{t}_0$ et $\mathbf{t}_1$ de $\mathbb{R}^2$ tels que $\mathbf{t}_1 \neq \pm\mathbf{t}_0$. Il existe un unique arc de cercle (orienté) court reliant $\mathbf{p}_0$ à $\mathbf{p}_1$ et admettant $\mathbf{t}_0$ et $\mathbf{t}_1$ comme vecteurs tangents en $\mathbf{p}_0$ et $\mathbf{p}_1$, respectivement, si et seulement si

$$\langle \mathbf{p}_1 - \mathbf{p}_0 \,,\, \mathbf{t}_1 - \mathbf{t}_0 \rangle = 0 \tag{1}$$

ou, en d'autres termes, si et seulement si le vecteur $\mathbf{p}_1 - \mathbf{p}_0$ appartient à la droite orthogonale au vecteur $\mathbf{t}_1 - \mathbf{t}_0$.

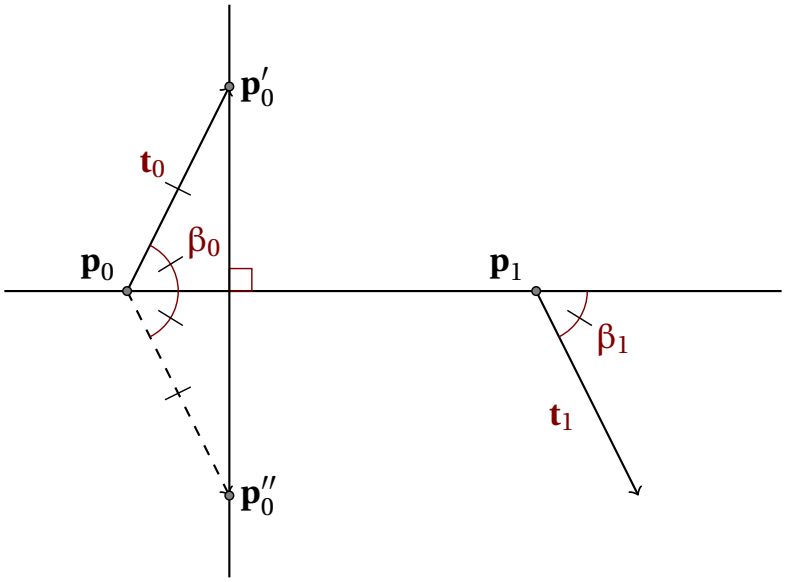

FIGURE 4 – Les points $\mathbf{p}_0$ et $\mathbf{p}_1$ munis des tangentes $\mathbf{t}_0$ et $\mathbf{t}_1$ satisfont la condition d'orthogonalité lorsque la droite reliant $\mathbf{p}_0$ et $\mathbf{p}_1$ est orthogonale à la droite $(\mathbf{p}'_0, \mathbf{p}''_0)$ de vecteur directeur $\mathbf{t}_1 - \mathbf{t}_0$.

Comment peut-on exprimer ce résultat de manière géométrique ? Considérons $\mathbf{p}'_0$ l'image de $\mathbf{p}_0$ par la translation de vecteur $\mathbf{t}_0$ et $\mathbf{p}''_0$ l'image de $\mathbf{p}_0$ par la translation de vecteur $\mathbf{t}_1$ (voir Figure 4). Alors la condition d'orthogonalité signifie que les deux points $\mathbf{p}_0$ et $\mathbf{p}_1$ forment une droite perpendiculaire au segment $[\mathbf{p}'_0\,\mathbf{p}''_0]$. Comme les deux tangentes sont unitaires, elles ont même

norme, et donc le triangle $(\mathbf{p}_0\,\mathbf{p}'_0\,\mathbf{p}''_0)$ est isocèle. Ainsi, la condition d'orthogonalité revient à dire que $\mathbf{p}_1$ est sur la bissectrice issue de $\mathbf{p}_0$ dans le triangle $(\mathbf{p}_0\,\mathbf{p}'_0\,\mathbf{p}''_0)$, ce qui équivaut à $\beta_0 = \beta_1$.

Nous pouvons maintenant démontrer le théorème. Supposons dans un premier temps qu'il existe un arc court (orienté) reliant $\mathbf{p}_0$ à $\mathbf{p}_1$ et admettant $\mathbf{t}_0$ et $\mathbf{t}_1$ comme vecteurs tangents en $\mathbf{p}_0$ et $\mathbf{p}_1$, respectivement. En reprenant les notations précédentes, montrons que $\mathbf{p}_1$ est sur la bissectrice issue de $\mathbf{p}_0$ dans le triangle $(\mathbf{p}_0\,\mathbf{p}'_0\,\mathbf{p}''_0)$ (voir Figure 5), ce qui est équivalent à montrer que la condition (1) est vérifiée.

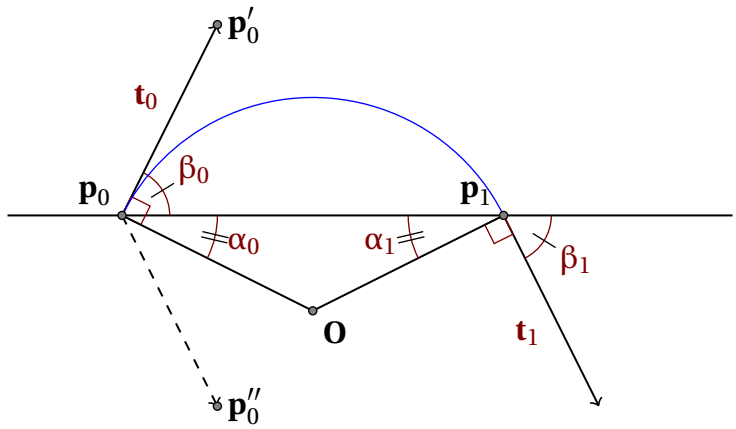

FIGURE 5 – On démontre que $\mathbf{p}_1$ est sur la bissectrice issue de $\mathbf{p}_0$ dans le triangle $(\mathbf{p}_0\,\mathbf{p}'_0\,\mathbf{p}''_0)$ en montrant l'égalité des angles β_0 et β_1.

Notons $\mathbf{O}$ le centre du cercle interpolant. Alors le triangle $(\mathbf{O}\,\mathbf{p}_1\,\mathbf{p}_0)$ est isocèle d'où $\alpha_0 = \alpha_1$. Or par construction, $\alpha_0 + \beta_0 = \frac{\pi}{2}$ et $\alpha_1 + \frac{\pi}{2} + \beta_1 = \pi$ d'où nous déduisons immédiatement $\beta_0 = \beta_1$ et $\mathbf{p}_1$ est sur la bissectrice issue de $\mathbf{p}_0$ dans le triangle $(\mathbf{p}_0\,\mathbf{p}'_0\,\mathbf{p}''_0)$.

Réciproquement, si la condition (1) est vérifiée, i.e., si $\beta_0 = \beta_1$, traçons la droite passant par $\mathbf{p}_0$ et perpendiculaire à $\mathbf{t}_0$ ainsi que celle passant par $\mathbf{p}_1$ et perpendiculaire à $\mathbf{t}_1$. Ces deux droites s'intersectent en un point $\mathbf{O}$. Montrons que le triangle $(\mathbf{O}\,\mathbf{p}_1\,\mathbf{p}_0)$ est isocèle. Comme précédemment, par construction, $\alpha_0 + \beta_0 = \frac{\pi}{2}$ et $\alpha_1 + \frac{\pi}{2} + \beta_1 = \pi$ et comme $\beta_0 = \beta_1$, on a $\alpha_0 = \alpha_1$ et le triangle est isocèle. L'arc de cercle de centre $\mathbf{O}$ qui part de $\mathbf{p}_0$ avec la tangente $\mathbf{t}_0$ atteint donc le point $\mathbf{p}_1$ avec la tangente $\mathbf{t}_1$. L'unicité de l'arc découle du fait que l'arc est nécessairement orienté (car il interpole des

vecteurs tangents, donc respecte un sens de rotation), et court (il ne fait pas de tour complet) [1].

Notons que la démonstration précédente a le grand intérêt d'être *constructive* : lorsque les conditions de co-circularité sont vérifiées, elle fournit une méthode pour construire l'unique arc de cercle interpolant (à partir du calcul du centre du cercle $\mathbf{O}$). Idéalement, nous souhaiterions avoir le même genre de caractérisation simple pour la *co-hélicité* en 3D, ainsi qu'une méthode systématique de construction en 3D de la ou des hélices interpolantes.

7 Et à trois dimensions ?

De manière remarquable, le théorème précédent ainsi que sa démonstration constructive se généralisent parfaitement à la 3D. On peut en effet formuler le théorème suivant pour caractériser la co-hélicité de deux points et deux vecteurs tangents :

Soient deux points $\mathbf{p}_0$ et $\mathbf{p}_1$ de $\mathbb{R}^3$ tels que $\mathbf{p}_0 \neq \mathbf{p}_1$, et deux vecteurs unitaires $\mathbf{t}_0$ et $\mathbf{t}_1$ de $\mathbb{R}^3$ tels que $\mathbf{t}_1 \neq \pm\mathbf{t}_0$. Il existe une unique hélice courte reliant $\mathbf{p}_0$ à $\mathbf{p}_1$ et admettant $\mathbf{t}_0$ et $\mathbf{t}_1$ comme vecteurs tangents en $\mathbf{p}_0$ et $\mathbf{p}_1$, respectivement, si et seulement si

$$\langle \mathbf{p}_1 - \mathbf{p}_0 \,,\, \mathbf{t}_1 - \mathbf{t}_0 \rangle = 0. \tag{2}$$

ou, en d'autres termes, si et seulement si le vecteur $\mathbf{p}_1 - \mathbf{p}_0$ appartient au plan orthogonal au vecteur $\mathbf{t}_1 - \mathbf{t}_0$.

Ce joli théorème, surprenant de simplicité, semble avoir été formulé pour la première fois par Ghosh en 2010 dans sa thèse de doctorat [4]. Cependant, contrairement à la 2D, en 3D la démontration de ce théorème n'est pas si simple. Dans sa thèse, Ghosh fournit une démonstration analytique du théorème, peu difficile d'accès mais pénible car calculatoire, et surtout incomplète : la condition suffisante du théorème n'est que partiellement démontrée. En allant jusqu'au bout du raisonnement et des calculs, on peut cependant démontrer le théorème dans son intégralité [2]. La démonstration complète est longue (une vingtaine de pages), et il existe certainement une démonstration plus courte s'appuyant sur des outils de géométrie différentielle. Néanmoins, l'intérêt de cette démonstration analytique (et c'était aussi le but poursuivi par Ghosh), est de fournir une méthode systématique pour

1. Exercice : Que se passe-t-il dans le cas où $\mathbf{t}_0$ et $\mathbf{t}_1$ sont colinéaires ?

construire l'unique hélice courte interpolante, une fois les points et tangentes placés selon la configuration favorable. Comme en 2D, la démonstration du théorème est dite *constructive*, ce qui présente un intérêt pratique évident.

8 Relaxation des points

Revenons à notre problème d'approximation initial. Nous avons vu que dans le cas général il est impossible d'interpoler deux points et deux tangentes par un arc de cercle ou une hélice. Il faut pour cela qu'ils satisfassent la condition de co-circularité ou de co-hélicité énoncée précédemment. Dans le cas où nous souhaitons interpoler des points et des tangentes pris sur une courbe donnée en entrée, il est donc nécessaire de relâcher une contrainte pour rendre l'interpolation possible. Notre intuition est que si l'interpolation des vecteurs tangents est importante afin de préserver l'allure générale de la courbe d'entrée, une interpolation stricte des points n'est pas indispensable : nous choisissons donc de relâcher la position des points. Nous interpolerons ainsi des tangentes "flottantes", c'est-à-dire non fixées à des points prédéterminés. Bien entendu, nous ne souhaitons pas que les points s'écartent trop de leur position d'origine. C'est la raison pour laquelle nous imposons de minimiser le déplacement de ces points. Considérons le cas de deux points $\mathbf{p}_0$ et $\mathbf{p}_1$ ainsi que deux tangentes $\mathbf{t}_0$ et $\mathbf{t}_1$ que nous prendrons unitaires et non colinéaires. Notons $\mathbf{p}'_0$ et $\mathbf{p}'_1$ les nouveaux points obtenus après déplacement. Nous pouvons alors formuler notre problème d'approximation de la façon suivante,

$$\begin{array}{ll} \min & \|\mathbf{p}'_0 - \mathbf{p}_0\|^2 + \|\mathbf{p}'_1 - \mathbf{p}_1\|^2 \\ (\mathbf{p}'_0, \mathbf{p}'_1) & \text{t.q. } (\mathbf{p}'_0, \mathbf{t}_0) \text{ et } (\mathbf{p}'_1, \mathbf{t}_1) \text{ sont co-hélicaux.} \end{array}$$

En d'autres termes, nous recherchons les points $\mathbf{p}'_0$ et $\mathbf{p}'_1$ tels qu'il existe une hélice passant par $\mathbf{p}'_0$ avec la tangente $\mathbf{t}_0$ et $\mathbf{p}'_1$ avec la tangente $\mathbf{t}_1$ (condition $(\mathbf{p}'_0, \mathbf{t}_0)$ et $(\mathbf{p}'_1, \mathbf{t}_1)$ co-hélicaux) et qui minimisent le déplacement par rapport à $\mathbf{p}_0$ et $\mathbf{p}_1$ (min $\|\mathbf{p}'_0 - \mathbf{p}_0\|^2 + \|\mathbf{p}'_1 - \mathbf{p}_1\|^2$). Résoudre ce problème de minimisation ne semble pas chose aisée au premier abord. Mais en remplaçant la condition de co-hélicité par son équivalent (2), nous obtenons un problème plus simple,

$$\begin{array}{ll} \min & \|\mathbf{p}'_0 - \mathbf{p}_0\|^2 + \|\mathbf{p}'_1 - \mathbf{p}_1\|^2 \\ (\mathbf{p}'_0, \mathbf{p}'_1) & \text{t.q. } \langle \mathbf{p}_1 - \mathbf{p}_0 \,,\, \mathbf{t}_1 - \mathbf{t}_0 \rangle = 0. \end{array}$$

Dans le cas planaire, la condition de co-hélicité signifie que les deux points sont sur la droite vectorielle orthogonale à la différence des deux tangentes et dans le cas spatial, elle signifie que les deux points sont dans le plan vectoriel orthogonal à la différence des deux tangentes. Considérons le cas le plus général de l'espace. Notons $(\mathbf{u}, \mathbf{v})$ une base du plan vectoriel orthogonal à la différence des deux tangentes [2]. Dès lors, nous pouvons écrire que la condition de co-hélicité est équivalente à l'existence de deux scalaires α et β tels que

$$\mathbf{p}_1' = \mathbf{p}_0' + \alpha\mathbf{u} + \beta\mathbf{v}.$$

Nous pouvons alors transformer notre problème en un problème plus simple encore,

$$\begin{array}{ll} \min & \|\mathbf{p}_0' - \mathbf{p}_0\|^2 + \|\mathbf{p}_1' - \mathbf{p}_1\|^2 \\ (\mathbf{p}_0', \mathbf{p}_1') & \text{t.q. } \mathbf{p}_1' = \mathbf{p}_0' + \alpha\mathbf{u} + \beta\mathbf{v}. \end{array}$$

Puis nous pouvons injecter la contrainte de co-hélicité directement dans la fonction objectif du problème de minimisation et nous obtenons enfin le problème suivant,

$$\min_{(\mathbf{p}_0', \alpha, \beta)} \quad \|\mathbf{p}_0' - \mathbf{p}_0\|^2 + \|\mathbf{p}_0' + \alpha\mathbf{u} + \beta\mathbf{v} - \mathbf{p}_1\|^2.$$

On identifie très facilement ici un problème classique de moindres carrés linéaires dont la solution unique est donnée par la simple résolution d'un systéme linéaire !

9 Récapitulation : algorithme des tangentes flottantes

Notre algorithme d'approximation, appelé "algorithme des tangentes flottantes", fonctionne en 3 étapes illustrées sur la Figure 6 ci-dessous.

En entrée, considérons une courbe régulière, à gauche sur la figure (sur l'illustration la courbe est planaire mais l'algorithme reste le même dans le cas de courbes spatiales).

1) Prenons des points avec leurs tangentes respectives régulièrement espacés sur la courbe à approcher. Puisque nous voulons N morceaux, considérons $N + 1$ points, comme représenté deuxième colonne de gauche sur la figure.

2. Exercice : comment construire facilement une telle base, en considérant les deux tangentes comme unitaires et non-colinéaires ?

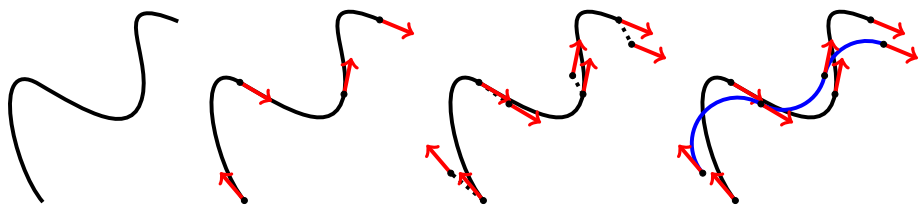

FIGURE 6 – Algorithme des tangentes flottantes.

2) Puisque dans le cas général il est impossible d'interpoler deux points successifs avec leurs tangentes par un unique arc de cercle ou d'hélice, relâchons la position de ces points de sorte que l'on puisse les interpoler, troisième colonne sur la figure. Cette relaxation revient à déplacer les points pour satisfaire la condition de co-circularité ou de co-hélicité, tout en minimisant ce déplacement, c'est-à-dire la distance entre la position sur la courbe en entrée et la position relâchée.

3) Maintenant que l'on peut les interpoler par un arc de cercle ou un arc d'hélice, nous obtenons la courbe en arcs de cercle ou en hélices par morceaux désirée comme représenté à droite sur la figure.

10 Validation

Il y a deux manières de montrer qu'un algorithme fonctionne correctement. La première, qui est la plus précise et la plus définitive, est de démontrer logiquement que l'algorithme produit bien les résultats souhaités. Ici, cela prendrait la forme d'un théorème assurant, pour une courbe donnée en entrée, un résultat compris dans un voisinage de cette courbe (voisinage qu'il faudrait définir précisément). Une telle manière de procéder est bien sûr satisfaisante intellectuellement, mais parfois difficile à mettre en place. Dans notre cas, il s'agit encore d'un problème ouvert.

La seconde approche, celle que nous allons employer ici, est plus empirique. Si nous ne pouvons démontrer précisément que l'algorithme fonctionne dans tous les cas, montrons de manière expérimentale qu'il fonctionne dans de nombreux cas. Ainsi, cette seconde approche consiste à faire tourner l'algorithme sur de nombreuses courbes données en entrée, si possible de nature variée, et à vérifier que les résultats obtenus sont satisfaisants.

Dans ce qui suit, nous nous proposons ainsi d'évaluer expérimentalement la qualité de notre algorithme. Deux aspects sont à évaluer : le temps de calcul d'une part, la qualité de l'approximation d'autre part. Pour évaluer le temps de calcul, il suffit de mesurer le temps que met l'algorithme (ou plus précisément, son implémentation sur un ordinateur) pour produire un résultat. Pour en évaluer la qualité, nous devons mesurer l'erreur commise par notre algorithme sur une courbe donnée. Mais comment mesurer cette erreur ? Ce que nous souhaitons mesurer, c'est la distance entre les deux courbes, celle donnée en entrée et la courbe calculée par notre algorithme. Pour cela, nous choisissons d'utiliser la distance de Fréchet qui est adaptée à la mesure de l'écart entre deux courbes orientées. Intuitivement, cette distance peut être décrite comme la longueur minimale nécessaire à une laisse pour qu'un chien et son maître se promènent, chacun sur une des deux courbes, à des vitesses différentes et quelconques, sans jamais rebrousser chemin. Plus formellement, la distance de Fréchet entre deux courbes Γ_1 et Γ_2 peut être définie comme la limite inférieure du maximum de $\|u_1(t) - u_2(t)\|$ pour deux paramétrisations quelconques u_1 et u_2 des courbes Γ_1 et $\Gamma 2$. En pratique, nous utiliserons la distance de Fréchet discrète [3], rapide à calculer, sur des échantillonnages de chacune des courbes.

Il est à noter que les résultats que nous présentons ici ont été réalisés avec une implémentation récente de l'algorithme des tangentes flottantes, disponible librement à l'adresse suivante : `http://www.inrialpes.fr/bipop/people/bertails/Papiers/floatingTangents3d.html`. Par ailleurs, cette implémentation étant plus récente que celle utilisée pour produire les résultats de l'article [2], les résultats présentés ici ont été calculés plus rapidement.

Courbes synthétiques Dans un premier temps, nous allons tester l'algorithme des tangentes flottantes sur des jeux de données synthétiques (voir Figure 7), représentant des chevelures crées par un artiste. Chaque coiffure contient environ 240 cheveux, chacun d'une longueur d'environ 30 cm.

Une caractéristique intéressante de ces trois jeux de données est le degré de complexité visuelle qui les compose. Intuitivement, une courbe est d'autant plus complexe qu'elle comporte de nombreuses boucles ou ondulations. Nous constatons visuellement que ces trois jeux de données représentent des courbes de complexité croissante. En effet, si les cheveux qui composent la première coiffure sont presque droits, ceux de la seconde sont un peu plus ondulés alors que ceux qui constituent la troisième sont nettement plus on-

FIGURE 7 – Trois coiffures synthétiques.

dulés que sur les deux premières. Plus quantitativement, nous proposons de mesurer la complexité des courbes par la quantité suivante, introduite par Levien dans son travail de thèse [6],

$$C(\Gamma) = \int_\Gamma \kappa^2 + \tau^2,$$

où κ représente la courbure géométrique de la courbe Γ, et τ sa torsion géométrique.

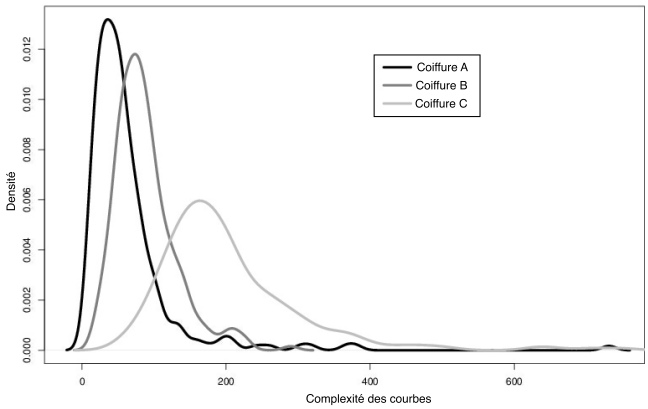

FIGURE 8 – Répartition des complexités pour les trois coiffures synthétiques.

Sur la Figure 8, nous avons représenté les distributions de complexité des courbes 3d pour chaque jeu de données. Nous constatons que la mesure de Levien correspond bien à l'intuition visuelle de la complexité d'une courbe.

En Figure 9, nous avons représenté la distribution des erreurs commises lors de l'approximation de chaque cheveu pour les trois coiffures, pour un nombre d'arcs d'hélice par courbe égal à 10 et à 15. Figure 10, nous avons représenté la distribution des temps de calcul nécessaire à l'approximation. Nous constatons une grande indépendance du temps de calcul de l'algorithme par rapport à la complexité de la courbe initiale. En effet, notre algorithme fonctionne en interpolant des tangentes prélevées sur la courbe d'origine. Par conséquent, la complexité de la courbe d'entrée n'est pas prise en compte lors de l'approximation. Nous voyons en revanche que l'erreur commise est plus importante lorsque la complexité s'accroît, à nombre d'arcs d'hélice égal, ce qui paraît parfaitement raisonnable. En effet, plus une courbe est complexe, plus elle requiert d'arcs d'hélice pour être approchée convenablement.

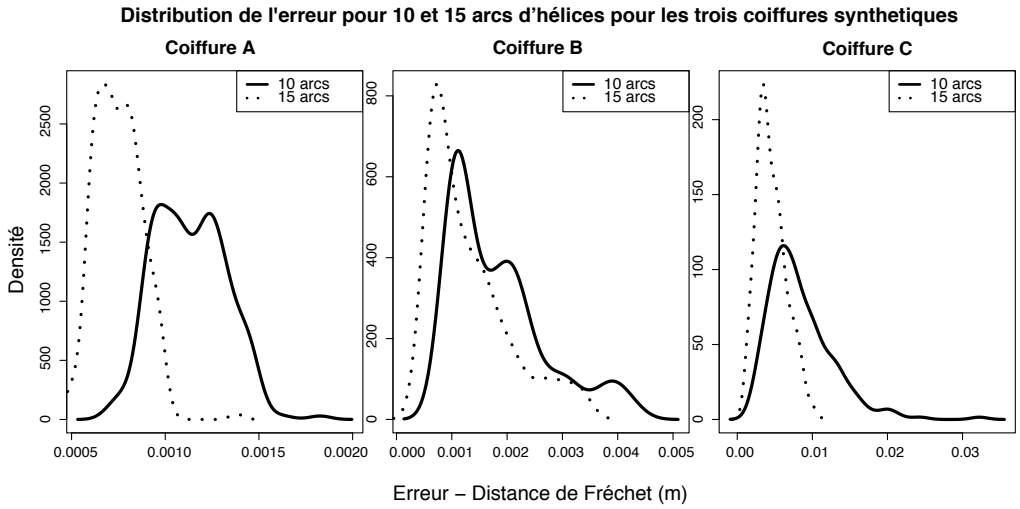

FIGURE 9 – Répartition des erreurs d'approximation pour l'algorithme des tangentes flottantes appliqué aux trois coiffures synthétiques.

La Figure 11 indique l'erreur moyenne commise sur chaque coiffure lorsque le nombre d'arcs d'hélice augmente. Nous constatons que l'algorithme présente un comportement souhaitable dans tous ces cas : plus le nombre d'arcs d'hélice passé en entrée augmente (autrement dit plus l'algorithme des tangentes flottantes possède de degrés de liberté), plus l'approximation est précise.

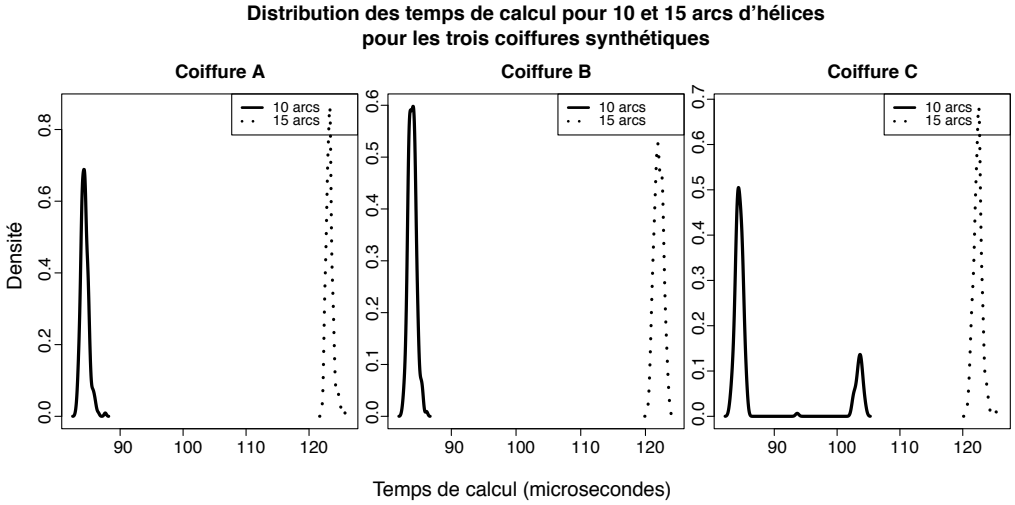

FIGURE 10 – Répartition des temps de calcul pour l'algorithme des tangentes flottantes appliqué aux trois coiffures synthétiques.

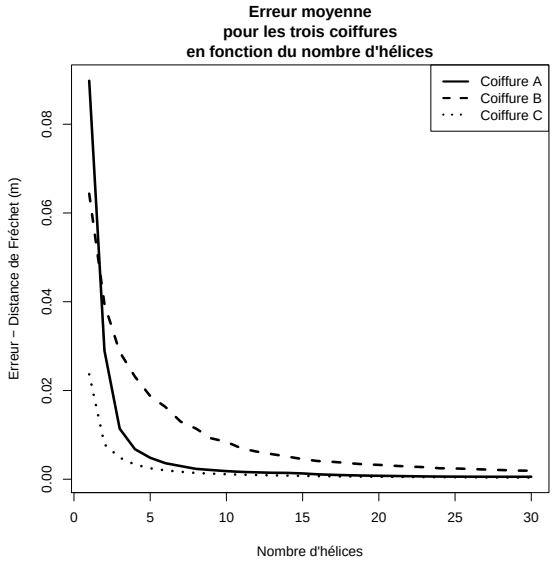

FIGURE 11 – Effet du nombre d'arcs d'hélices sur l'erreur d'approximation pour l'algorithme des tangentes flottantes appliqué aux trois coiffures synthétiques.

Courbes réelles Nous allons appliquer notre algorithme des tangentes flottantes à l'approximation de 3 types de jeux de données, représentant des courbes variées, issues de la capture de portions de cheveux [5], Figures 12, 13 et 14 ou de fibres musculaires de cœur [7], Figure 15 ou encore issues de la simulation des lignes du champ magnétique d'une étoile [1], Figure 16.

Sur chaque figure, nous représentons à gauche les données originales et à droite le résultat de notre algorithme. Toutefois, à cause de l'hétérogénéité des longueurs des courbes, nous devons adapter le nombre d'éléments utilisés pour l'approximation. Ainsi, pour les données issues de la capture de cheveux, nous avons utilisé de 5 à 20 éléments par courbe, de 2 à 100 éléments par courbe pour les fibres musculaires et de 2 à 200 éléments par courbe pour les lignes du champ magnétique. Nous reportons les erreurs commises et les temps de calcul pour chaque jeu de données dans le tableau 17.

Les erreurs sont exprimées en pourcentage de la taille du médian des longueurs des courbes du jeu de données et les temps de calcul en millisecondes. Nous constatons que sur tous les jeux de données, l'algorithme des tangentes flottantes permet d'obtenir en un temps raisonnable (au plus quelques minutes pour un millier de courbes) une approximation précise des courbes passées en entrée. Ces résultats illustrent bien que l'algorithme des tangentes flottantes permet d'approcher de manière précise et rapide de gros jeux de données issus d'applications réelles.

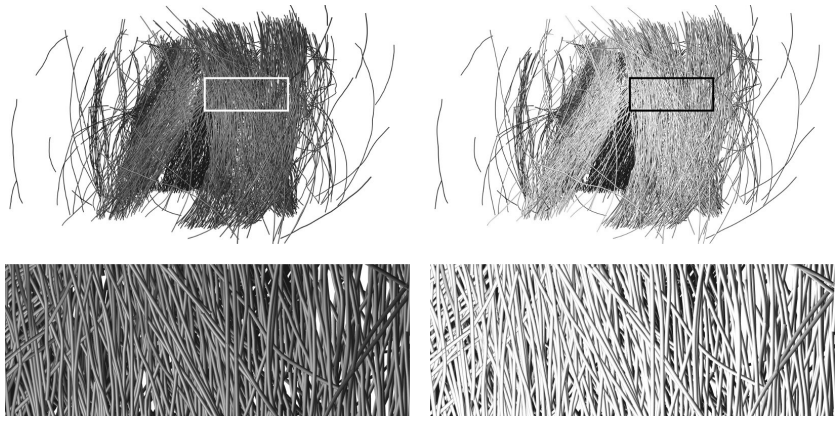

FIGURE 12 – Jeu de données (curls) représentant une portion de chevelure réelle capturée par un procédé automatique. À gauche, les données originales, à droite le résultat de l'algorithme des tangentes flottantes.

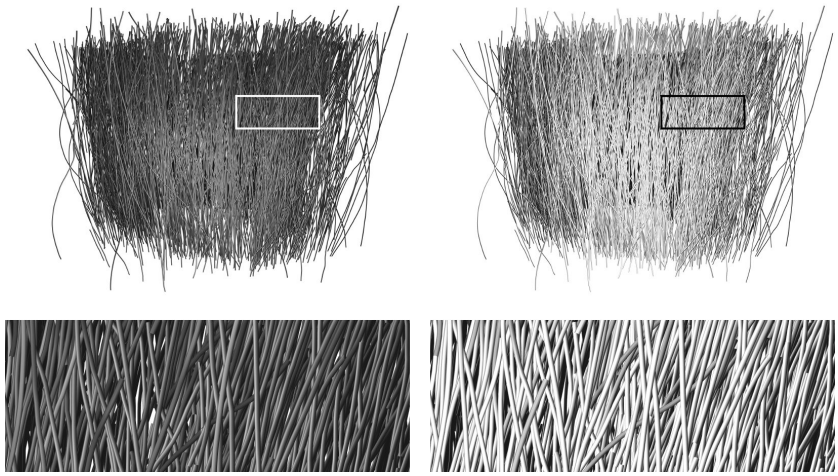

FIGURE 13 – Jeu de données (straight) représentant une portion de chevelure réelle capturée par un procédé automatique. À gauche, les données originales, à droite le résultat de l'algorithme des tangentes flottantes.

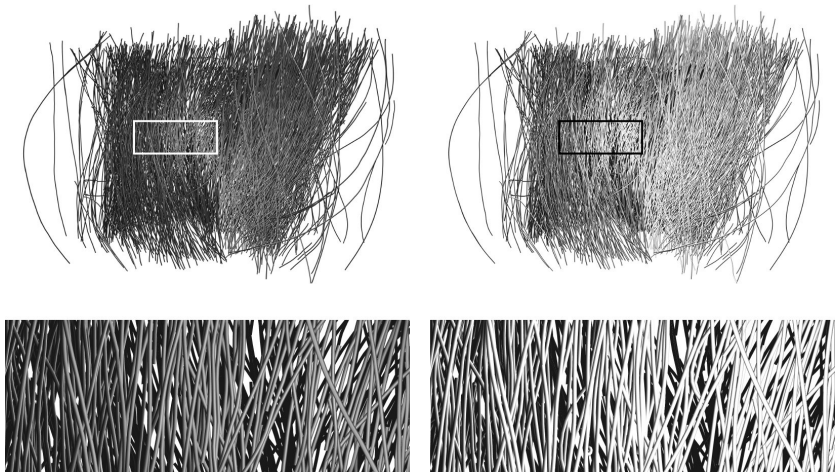

FIGURE 14 – Jeu de données (wavy) représentant une portion de chevelure réelle capturée par un procédé automatique. À gauche, les données originales, à droite le résultat de l'algorithme des tangentes flottantes.

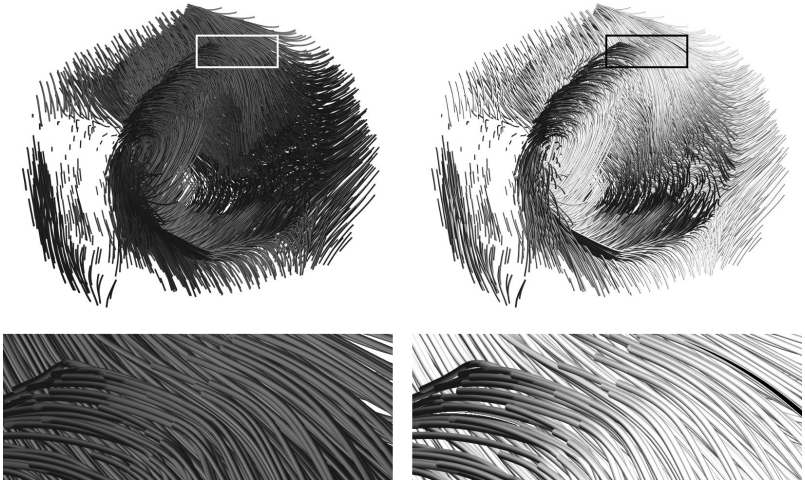

FIGURE 15 – Jeu de données représentant les fibres musculaires d'un cœur. À gauche, les données originales, à droite le résultat de l'algorithme des tangentes flottantes.

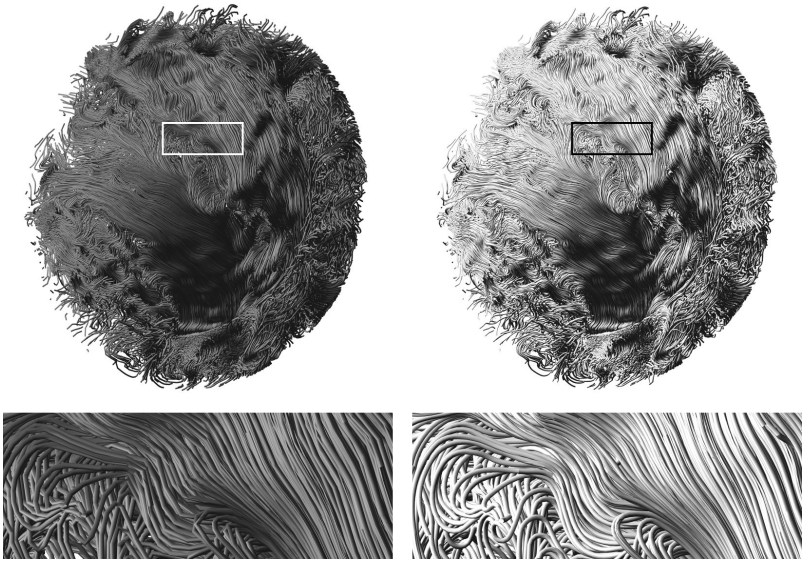

FIGURE 16 – Jeu de données représentant les lignes d'un champ magnétique d'une étoile, obtenues par simulation. À gauche, les données originales, à droite le résultat de l'algorithme des tangentes flottantes.

	Cheveux 1 (curls)	Cheveux 2 (straight)	Cheveux 3 (wavy)	Coeur	Étoile
erreur moyenne	0.7%	0.7%	0.6%	0.2%	0.7%
erreur maximale	5%	14%	2%	1%	6%
Tps moyen (ms)	30	32	38	14	101
Tps maximal (ms)	206	128	143	257	3320
Tps total (s)	60	79	58	42	890

FIGURE 17 – Temps de calcul et erreur d'approximation de l'algorithme des tangentes flottantes appliqué aux jeux de données réelles.

11 Limitations de l'algorithme

Phénomène de boucle Nous avons vu que dans de nombreux cas, synthétiques comme réels, l'algorithme des tangentes flottantes répond très bien aux exigences de l'utilisateur : qualité et rapidité. Toutefois, comme nous pouvons le voir pour l'erreur représentée Figure 18, il arrive dans certains cas dégénérés que des erreurs importantes apparaissent, même avec un nombre d'arcs d'hélice important. Ce phénomène, que nous désignons par "phénomène de boucle" se produit lorsque deux tangentes proches l'une de l'autre sur la courbe sont inversées par la relaxation des points, ce qui transforme de petits arcs en arcs de grande longueur. Nous illustrons ce phénomène dans le cas 2d en Figure 19.

Découpage régulier Par ailleurs, lors du découpage de la spline, nous avons choisi d'espacer les points de manière régulière sur la spline, pour produire une courbe en morceaux d'hélices ayant à peu près la même longueur. Si ce choix a peu de répercussions lorsque le nombre d'arcs est grand, il peut devenir problématique lorsque le nombre d'arcs est faible. En effet, si nous envisageons une courbe qui aurait deux parties, l'une longue et de courbure faible et l'autre courte mais de courbure importante, nous pourrions vouloir l'approcher avec seulement deux arcs d'hélice. Hélas, avec l'algorithme tel qu'il est présenté ici, nous ne pourrions pas obtenir de résultat satisfaisant, puisque le point de raccord des deux arcs d'hélice se trouverait au milieu de la courbe d'entrée et non pas à la jonction des deux parties la composant. Afin d'éviter ce type de problème, il serait judicieux de trouver les meilleurs points où découper la courbe en entrée, par exemple en étudiant les variations de la courbe pour trouver les parties les plus susceptibles d'être approchées par un arc d'hélice.

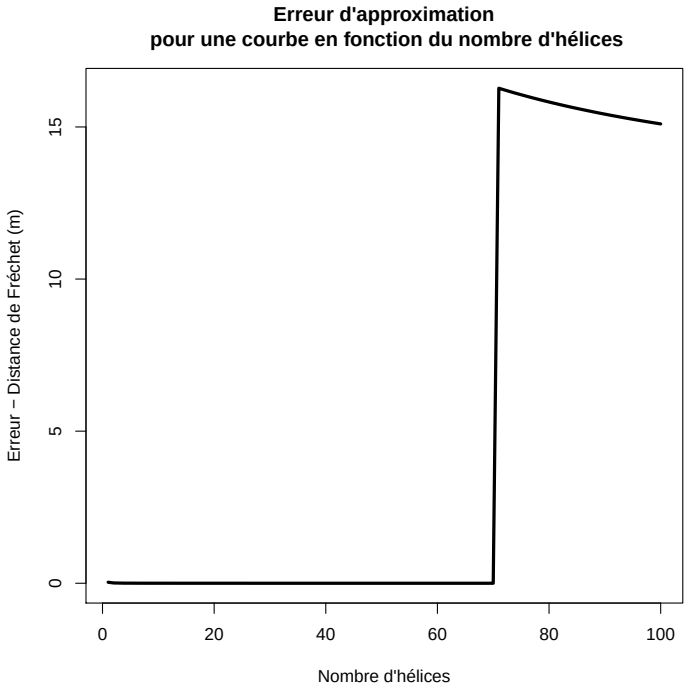

FIGURE 18 – Erreur d'approximation de l'algorithme des tangentes flottantes pour une courbe donnée. Le brusque saut qui survient autour de 70 arcs d'hélice est la manifestation du phénomène de boucle qui peut apparaitre lorsque des points de rupture sont proches.

Faible contrôle sur l'erreur Enfin, nous avons fait le choix de laisser le nombre d'arcs en paramètre d'entrée de l'algorithme, permettant à l'utilisateur de le contrôler précisément. Ce choix s'accompagne d'une limitation évidente sur l'erreur d'approximation : il est difficile de la contrôler finement. En effet, nous avons vu que lorsque le nombre d'arcs d'hélice était trop faible, l'erreur pouvait être grande. De même, lorsque le nombre d'arcs d'hélice est élevé, le phénomène de boucle peut survenir et produire une approximation de piètre qualité. Nous pourrions donc envisager comme extension de l'algorithme une méthode pour prédire l'erreur d'approximation commise, ce qui permettrait de guider l'utilisateur de l'algorithme dans son choix du nombre d'arcs d'hélice. Plus simplement, nous pourrions envisager de détecter les boucles et d'y remédier lors de la relaxation des points.

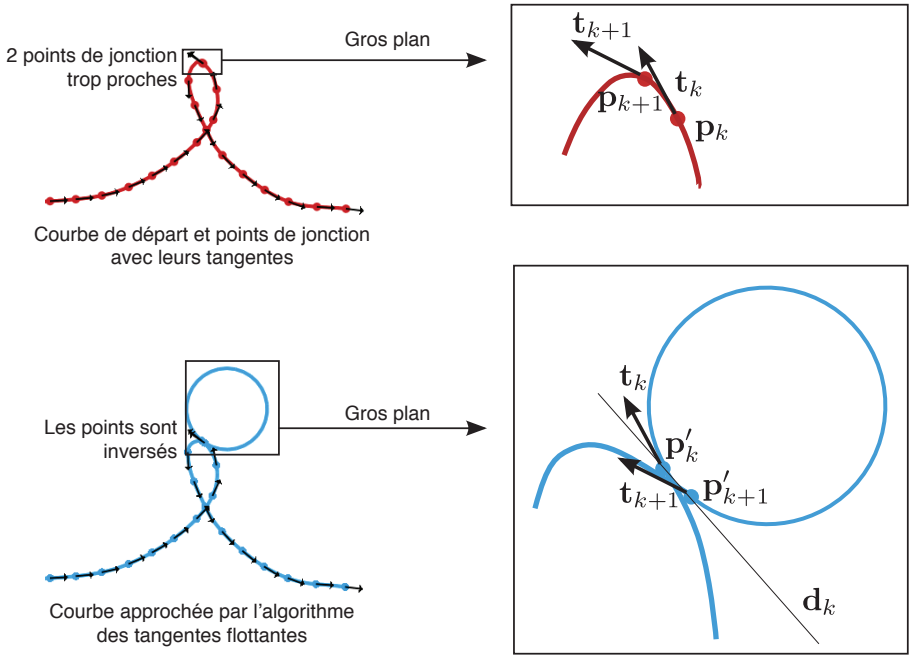

FIGURE 19 – Le phénomène de boucle est dû à l'inversion de la position des tangentes lors de la relaxation.

Références

[1] Benjamin P. Brown, Mark S. Miesch, Matthew K. Browning, Allan Sacha Brun, and Juri Toomre. Magnetic cycles in a convective dynamo simulation of a young solar-type star. *The Astrophysical Journal*, 731(1) :69, 2011.

[2] Alexandre Derouet-Jourdan, Florence Bertails-Descoubes, and Joëlle Thollot. Floating tangents for approximating spatial curves with piecewise helices. *Computer Aided Geometric Design*, 30(5) :490–520, 2013.

[3] Thomas Eiter and Heikki Mannila. Computing discrete Fréchet distance. Technical report, Technische Universitat Wien, 1994.

[4] Sunayana Ghosh. *Geometric approximation of curves and singularities of secant maps. A differential geometric approach.* PhD thesis, University of Groningen, Johann Bernoulli Institute for Mathematics and Computer Science., December 2010.

[5] Wenzel Jakob, Jonathan T. Moon, and Steve Marschner. Capturing hair assemblies fiber by fiber. *ACM Trans. Graph.*, 28(5) :164 :1–164 :9, December 2009.

[6] Raphael Linus Levien. *From Spiral to Spline : Optimal Techniques in Interactive Curve Design.* PhD thesis, EECS Department, University of California, Berkeley, Dec 2009.

[7] Damien Rohmer, Arkadiusz Sitek, and Grant T Gullberg. Reconstruction and visualization of fiber and laminar structure in the normal human heart from ex vivo diffusion tensor magnetic resonance imaging (dtmri) data. *Investigative Radiology*, 42(11) :777–789, Nov 2007.