THÈSE

présentée à

L'UNIVERSITÉ DE PARIS VII

pour obtenir le titre de

DOCTEUR ES SCIENCES

Spécialité :

INFORMATIQUE

par

Francis LAZARUS

Sujet de la thèse :

COURBES, CYLINDRES ET MÉTAMORPHOSES pour l'image de synthèse

Soutenue le 21 Décembre 1995, devant le jury composé de :

M.	**Christian**	**CHOFFRUT**	Président
M.	**Claude**	**PUECH**	Rapporteur
M.	**Francis**	**SCHMITT**	Rapporteur
M.	**Pascal**	**LIENHARDT**	Examinateur
M.	**André**	**GAGALOWICZ**	Directeur de thèse
Mm.	**Anne**	**VERROUST**	Responsable de thèse

A Pascale et Mehdi

La forme belle, qu'elle soit un arbre ou un fleuve, les seins d'une femme ou ses flancs, les épaules ou les bras d'un homme ou le crâne d'un dieu, la forme belle c'est la forme qui s'adapte à sa fonction. L'idée n'a pas d'autre rôle que de nous la définir. L'idée, c'est l'aspect supérieur et l'extension infinie dans le monde et l'avenir du plus impérieux de nos instincts qu'elle résume et dénonce comme la fleur et le fruit résument la plante la prolongent et la perpétuent.

ELIE FAURE.

Remerciements

Ce travail a été réalisé dans d'excellentes conditions et je tiens à exprimer mes remerciements envers toutes celles et tous ceux qui ont contribué à son accomplissement :

Je remercie Anne Verroust qui par son encadrement et son soutien quotidien m'a guidé tout au long de cette thèse. Nos nombreuses discussions évoquent pour moi un idéal du monde de la recherche où l'écoute de l'autre et l'ouverture d'esprit s'accordent au son d'un travail harmonieux. Je la remercie également pour son aide précieuse dans la rédaction de ce manuscrit, je n'aurai pu espérer meilleur conseiller.

Je remercie André Gagalowicz qui m'a accueilli dans son laboratoire et a mis à ma disposition des moyens informatiques et audiovisuels à la mesure de ce qu'exige la synthèse d'images.

Je remercie Claude Puech et Francis Schmitt qui ont accepté d'être rapporteurs de cette thèse et de siéger à son jury. Qu'ils trouvent ici l'expression de ma gratitude pour l'intérêt qu'ils ont porté à mon travail.

Je remercie Christian Choffrut et Pascal Lienhardt pour l'intérêt qu'ils ont porté à mon travail et pour m'avoir fait l'honneur de bien vouloir faire partie de mon jury.

Je remercie l'équipe audiovisuelle de l'INRIA et tout particulièrement Arghyro Paouri, Christian Blonz et Alexandre Eidelman pour leur précieuse collaboration, pour leurs conseils et leur gentillesse.

Je remercie Jean-Paul Chieze qui m'a guidé dans les méandres de l'univers informatique toujours avec patience et gentillesse. Son importance est capitale dans la réalisation de ce travail.

Je remercie Philippe Decaudin, Fabrice Neyret et Frederic Taillefer, mes compagnons de bureau, pour l'aide qu'ils m'ont apportée et leur gentillesse à mon égard. Ils ont grandement contribué à rendre ce travail agréable.

Un grand merci à Laurence Bourcier, notre secrétaire pour sa grande disponibilité et son extraordinaire patience, ainsi qu'à tous les membres du projet SYNTIM Bruno, Jean-philippe, Xavier, Christophe, Marie-Luce, Pierre, Sabine et tous les autres...

Je remercie Pascale pour sa lecture attentive de ce manuscrit, ainsi bien sûr que pour son soutien, sa confiance et l'aide inestimable qu'elle m'a apportée durant ces trois années de thèse.

Je remercie Mehdi dont la naissance a coïncidé avec le début de la rédaction de ce manuscrit et qui m'a fourni toute l'énergie nécessaire pour cette tâche difficile.

Enfin, je remercie tous mes proches pour leurs encouragements.

Table des matières

Introduction **1**

I Interpolation de courbes **5**

1 Un procédé canonique d'interpolation de courbes sur les groupes de Lie **9**
1.1 Interpolation de points 9
1.2 Définition d'une courbe 10
1.3 Position du problème 11
1.3.1 Forme et position 12
1.4 Interpolation de courbes 13
1.4.1 Homotopie dans $\mathcal{C}^p(I, X)$ 14
1.4.2 Homotopie régulière dans $\mathcal{C}^p(I, X)$ 17
1.5 Conclusion 22

2 Un algorithme d'interpolation de courbes polygonales **25**
2.1 Du modèle continu au modèle discret 26
2.1.1 La notion de modèle en modélisation géométrique 26
2.1.2 Modèle et classes d'équivalence 27
2.1.3 Un modèle de courbe 30
2.2 Algorithme d'interpolation 32
2.2.1 Schéma principal 32
2.2.2 Construction d'un repère adapté à une courbe échantillonnée 33
2.2.3 Interpolation des repères (première version) 34
2.2.4 Interpolation des repères (seconde version) 37
2.3 Extension 39
2.3.1 Torsion des repères 39
2.3.2 Contrôle de la vitesse de transformation 40
2.3.3 Redressement automatique 40

2.4 Implémentation et résultats 41
2.5 Conclusion 44

II Aide à la création d'axe 47

1 Squelettisation et axe médian 51
1.1 Axe médian et morphologie mathématique 52
1.1.1 Algorithmes de construction de l'axe médian 53
1.2 Conclusion 54

2 Vers un diagramme de Reeb 57

3 Un algorithme de découpage en bandelettes 63
3.1 Introduction 63
3.2 Géodésiques sur les polyèdres 64
3.3 Fonctions numériques sur les polyèdres 66
3.3.1 Coloriage et caractéristique d'Euler-Poincaré 68
3.3.2 Courbes de niveau des coloriages 71
3.3.3 Courbes de niveau et coloriage sur les triangulations 73
3.3.4 Décomposition en bandelettes 77
3.3.5 Cas des polyèdres non triangulés 78
3.3.6 Cas des triangulations non coloriées 79
3.4 Application au coloriage par la distance géodésique 79
3.4.1 Algorithme 82
3.4.2 Le cas pratique 87
3.5 Création d'axe 89
3.5.1 Quelques résultats 90
3.5.2 À propos du centrage 92
3.6 Conclusion 96

III Métamorphoses 99

1 État de l'art 103
1.1 L'approche volumique 103
1.1.1 Métamorphoses bidimensionnelles 103
1.1.2 Métamorphoses tridimensionnelles 105
1.2 L'approche surfacique 108

1.2.1 Métamorphoses de contours 109
1.2.2 Métamorphoses de surfaces 112
1.2.3 Métamorphoses et évolutions de surfaces 114
1.3 Conclusion 115

2 Un algorithme de métamorphose 117
2.1 Schéma principal 117
2.2 Système de coordonnées cylindriques 119
2.2.1 Système de coordonnées associé à un modèle de courbe 121
2.2.2 Attachement d'un objet à un modèle de courbe 122
2.2.3 Extensions aux extrémités d'une courbe 124
2.3 Paramétrage et échantillonnage 126
2.3.1 Paramétrage à partir d'un axe 126
2.3.2 Échantillonnage à partir d'un axe 128
2.4 Recalage 132
2.5 Mise en correspondance et interpolation 137
2.5.1 Mise en correspondance 137
2.5.2 Interpolation 138
2.6 Exemples de métamorphoses 141
2.7 Conclusion 146

Conclusion 147

Appendices 151

A Linéarisation d'un champ invariant 151

B Γ_i^n converge uniformément vers Γ 153

C Solution numérique de $R' = R\ \boldsymbol{\Omega}$ 155

D Interpolation des repères (seconde version) 159

E La topologie en tout ou rien 163

F Construction d'un champ sur un triangle 167
F.1 Construction d'un champ particulier 167
F.2 Renormalisation 170

Lexique 173

Bibliographie **179**

Introduction

Motivation

Nous abordons ici le calcul automatique de métamorphoses entre objets surfaciques. Pour nous, une métamorphose désigne le passage continu d'une forme à une autre. En particulier, il faut distinguer ce sens de celui, également commun, d'évolutions biologique. Or, si la simulation informatique d'évolution trouve des applications dans l'étude des sciences botaniques, par exemple, celle de métamorphose a par contre bien du mal à se justifier "scientifiquement". Car les métamorphoses, comme nous l'entendons, n'existent que dans notre imaginaire, et c'est bien là leur intérêt. Les cinéastes l'ont bien vite perçu qui dès l'apparition du cinématographe ont tentés de donner forme à cet imaginaire. Les moyens techniques aidant, l'informatique en tête, l'emploi des métamorphoses à l'écran s'est multiplié, offrant au spectateur des images aux contours de plus en plus précis, au risque d'y perdre un peu de poésie.

N'est-il pas étrange de commencer par dénoncer un sujet apparemment non-scientifique et dont les implications pourraient au mieux réduire encore l'espace imaginaire bien étroit laissé par notre société moderne et technologique ? À la vérité, nous espérons ces critiques infondées et ceci pour deux raisons. D'abord parce que nous croyons que ce qui permet de qualifier un texte de scientifique ce ne sont pas tant ces implications ou le fait qu'il s'inscrive dans telle grande théorie, que son caractère rigoureux et avant tout honnête et novateur. Ensuite, parce que nous n'avons nullement la prétention de rivaliser avec la bonne vieille "technique" du flou artistique probablement bien plus efficace et chargée de mystères que les transformations stériles d'un Michael Jackson.

Notre travail s'inscrit dans une discipline nouvelle, née de l'informatique, l'*image de synthèse*. Comme son nom l'indique, cette discipline s'intéresse à la production d'images "synthétiques" qui ne pourraient être réalisées sans le support de l'ordinateur. Notre objectif est de définir puis de proposer des outils informatiques qui permettent à des personnes douées d'un certain sens esthétique ou artistique de créer des images fixes ou animées. L'intérêt même de ces images, c'est-à-dire leur potentiel d'expression, reste en grande partie à démontrer, mais là n'est pas notre rôle. Notre motivation est venue d'une part de l'intuition que des rapprochements entre diverses disciplines telles que l'informatique, l'algorithmique ou les mathématiques et les arts graphiques (au sens large) pouvaient être la source de problèmes riches et passionnants, d'autre part d'un sentiment après tout bien naturel : la curiosité.

Principes et Organisation

Nous nous sommes fixé pour tâche de concevoir un outil informatique permettant d'effectuer des métamorphoses d'objets surfaciques (nous ne nous interesserons ici qu'au cas des polyèdres ou des surfaces implicites). Pour qu'un outil informatique mérite cette appellation, il doit comporter un certain degré d'automatisation évitant à l'utilisateur potentiel des spécifications souvent fastidieuses voire irréalisables par leur nombre. Paradoxalement, de par ce caractère automatique, un outil définit implicitement une réalisation "canonique" de la tâche à effectuer et prive l'utilisateur d'une quelconque liberté d'expression. Nous pouvons cependant assouplir cette automatisation à l'aide de paramètres intuitifs qui permettent de réaliser une même tâche de manières différentes selon des critères déterminés par l'utilisateur. De ce point de vue, la conception d'un outil de métamorphose est particulièrement ardue car il n'existe pas de moyen a priori pour paramétrer des transformations entre objets.

Comme souvent dans ce type de situations nous pouvons procéder par simplification en nous plaçant dans un cadre beaucoup plus restreint, empruntant aux mathématiques leurs capacités d'abstraction et de simplification.

Nous supposerons ainsi que nos objets informatiques représentent des variétés compactes de dimension 2, plongées ou immergées dans $\mathbb{R}^3$. Une métamorphose peut alors s'identifier à une homotopie de plongement ou d'immersion. Ceci suppose des topologies identiques pour deux objets à métamorphoser. Comme les topologies de nos objets sont dénombrables (toute variété compacte de dimension 2 est homéomorphe soit à un tore à n trous soit au recollement de n espaces projectifs) nous pouvons imaginer, dans un premier temps, que les changements topologiques surviennent à des instants isolés où l'on effectue certaines opérations topologiques. En dehors de ces instants nous devons effectivement réaliser des homotopies.

Plus précisément, nous nous sommes intéressés à des objets dont la forme tubulaire permet de les identifier "naturellement" à des cylindres fermés par deux petits disques.

Le cylindre est en effet un objet simple qui s'identifie, sur le plan topologique, au produit $S^1 \times D^1$ du cercle S^1 par le disque D^1 (i.e. un segment) ; nous pouvons "dé-symétriser" ce produit en considérant le cylindre comme un espace fibré de base D^1 dont les fibres sont isomorphes à S^1. D^1 peut ainsi être considéré comme l'**axe** du cylindre auquel sont attachées ses fibres. L'identification naturelle de nos objets signifie que cet axe et ces fibres peuvent être obtenus naturellement par une construction géométrique.

En paramétrant les axes associés à deux objets cylindriques, nous pouvons mettre en correspondance les fibres des deux objets qui se projettent sur leur axe respectif en un point de même paramètre. Une métamorphose entre deux objets peut alors être conçue comme une transformation entre les axes de ce cylindre suivie d'une transformation entre les paires de fibres associées.

Dans les trois parties qui composent ce mémoire nous verrons ainsi comment transformer un axe en un autre, comment construire un axe à partir d'une forme tubulaire et, finalement, comment réaliser une métamorphose entre objets tubulaires. Plus précisément :

- Dans la première partie, pour définir une transformation entre deux axes, nous construisons une homotopie régulière entre deux segments immergés et nous en déduisons un algorithme d'interpolation entre deux courbes polygonales. Dans cette partie nous insistons particulièrement sur la suite de raisonnements qui mènent de l'intuition mathématique à l'implémentation de l'algorithme.

- Dans la seconde partie nous étudions comment associer une courbe squelette à un objet polyédrique. Dans ce cadre nous proposons une décomposition des polyèdres à l'aide de cycles d'arêtes comparables aux fibres évoquées plus haut.

- Dans la troisième partie nous décrivons un algorithme de métamorphose entre deux objets de forme tubulaire. Des problèmes liés à la définition d'un système de coordonnées cylindriques y sont notamment détaillés dans une optique algorithmique.

Dans la première partie, tout particulièrement, nous emprunterons à la géométrie différentielle un peu de son vocabulaire. Comme ce vocabulaire n'est pas forcément celui de la communauté de l'image de synthèse, nous avons jugé utile d'ajouter en fin de ce mémoire un lexique reprenant les principaux termes employés.

Première partie

Interpolation de courbes

Dans cette première partie, nous nous intéressons au problème de l'interpolation dans $\mathbb{R}^3$ de deux courbes gauches non-fermées, modélisées par des courbes polygonales.

Les déformations et transformations de courbes ont largement été étudiées en analyse d'image. Ainsi, la recherche des lignes de contours dans une image composée de pixels a conduit à développer un modèle de *contours actifs* [KWT87]. Dans ce cadre le problème est de trouver une courbe suffisamment régulière vérifiant au mieux certaines contraintes liées aux résultats cherchés dans l'image. Comme la définition de cette courbe n'est en général pas explicite, la solution proposée est de faire évoluer une courbe initialement donnée et supposée proche de la solution. Cette évolution, guidée par les contraintes, permet de trouver la courbe optimale. La résolution à l'aide de l'ordinateur fait appel à des techniques d'analyse fonctionnelle et de calcul numérique. Dans ce cadre elle est particulièrement adaptée à des calculs dans des espaces discrets et tout particulièrement dans le plan $\mathbb{Z}^2$.

Dans un cadre plus général on étudie dans [KTZ92, ST94, Fau93] l'évolution de courbes fermées en fonction d'une équation d'évolution faisant intervenir des quantités propres à un *plongement* de la courbe dans le plan (euclidien, affine ou projectif), c'est à dire essentiellement la courbure de la courbe et ses dérivées. Dans [Fau93], on introduit la méthode du repère mobile pour analyser les déformations d'une courbe. Nous reprendrons par la suite cette idée.

Ces techniques ne sont cependant pas applicables à notre problématique car :

- les objets étudiés sont différents : il s'agit de courbes de $\mathbb{R}^3$ et non du plan,
- la problématique est différente : il s'agit pour nous de faire évoluer une courbe entre des positions initiale et finale *données*,
- les objets ne sont pas représentés par les points d'une grille (le plan $\mathbb{Z}^2$) mais par des n-uplets de réels (c.a.d des courbes polygonales).

Quelques approches ont étés proposées en infographie pour transformer une ligne polygonale en une autre [SG92, SGWM93, GH94, SR95, GG95] afin de réaliser une métamorphose ou de calculer des images intermédiaires entre deux images clés d'une animation (key-framing). Dans ces méthodes les polygones sont supposés planaires et fermés, ce qui n'est pas notre cas. Nous aurons l'occasion cependant de revenir sur les deux dernières références.

Il nous faut ainsi définir entièrement un procédé d'interpolation de courbes de $\mathbb{R}^3$. Dans un premier temps nous considèrerons ce problème dans le cadre classique de la géométrie différentielle et nous proposerons une solution compatible avec nos besoins. Nous verrons ensuite comment adapter cette solution dans un cadre algorithmique où les objets étudiés ne sont plus des variétés différentielles mais des objets décrits par une combinatoire.

L'idée directrice sera de définir des procédés les plus simples possibles ne faisant appel, si possible, qu'à des notions intrinsèques aux objets étudiés. Dans les sections suivantes nous progresserons vers une solution en écartant au fur et à mesure certains procédés intuitifs d'interpolation et en montrant pourquoi ils ne peuvent fonctionner.

Chapitre 1

Un procédé canonique d'interpolation de courbes sur les groupes de Lie

1.1 Interpolation de points

Avant de se poser le problème de l'interpolation de courbes nous pouvons déjà regarder comment interpoler deux points P_0 et P_1 : nous cherchons à relier P_0 à P_1 par un chemin. Pour donner un sens à cette quête, nous supposerons que ces points appartiennent à un espace topologique donné X.

- Si nous n'avons aucune information supplémentaire sur l'espace X, ce problème peut ne pas avoir de solution. C'est le cas par exemple si P_0 et P_1 sont dans deux composantes connexes distinctes de X.
- Si nous supposons X connexe par arc alors nous savons que P_0 et P_1 peuvent être reliés mais nous n'avons aucun moyen de définir un chemin particulier.
- Si X est de plus muni d'une métrique nous pouvons tenter de chercher notre solution parmi les chemins γ les plus courts. Rappelons que la longueur d'un chemin [LFA77] est donnée par

  $$\sup_{\sigma\in\mathcal{S}} \sum_{0\leq i<n} d(\gamma(t_i),\gamma(t_{i+1}))$$

  où $\mathcal{S}$ est l'ensemble des subdivisions $\sigma = (t_0, t_1, ..., t_n)$ de $[0,1]$ $(t_0 = 0, t_n = 1)$ et d la distance sur X. Lorsque cette quantité est finie on dit que γ est *rectifiable*.

  Malheureusement, dans le cas général, nous ne savons pas s'il existe un chemin rectifiable joignant deux points et quand il en existe un, s'il existe un unique chemin rectifiable de longueur minimale.

Aux vues des problèmes soulevés par ces quelques exemples nous proposons une définition du mot interpolation.

Définition 1.1 *Un procédé d'interpolation sur un espace topologique X est un moyen (canonique) pour construire une application γ continue de [0,1] dans X, d'extrémités $\gamma(0) = P_0$ et $\gamma(1) = P_1$, où P_0 et P_1 sont deux points quelconques de X.*

L'exemple fondamental est, à ce titre, celui de l'interpolation linéaire sur les espaces vectoriels (topologiques).

Si l'on se restreint à une interpolation *locale*, les points P_0 et P_1 étant supposés proches, il est également possible de définir des procédés d'interpolation sur deux autres catégories d'espaces topologiques : les groupes de Lie et les variétés riemaniennes.

Dans ces deux cas, on peut en effet définir une paramétrisation locale canonique (l'application exponentielle). L'inverse de cette paramétrisation envoie localement la variété sur des ouverts (convexes) d'un espace vectoriel où l'on peut utiliser l'interpolation linéaire.

Sur les variétés riemaniennes ce procédé consiste à relier P_0 et P_1 par une géodésique de longueur minimale. Nous savons en effet qu'autour de tout point il existe un voisinage dans lequel tout couple de points est relié par une unique géodésique de longueur minimale. Ce voisinage est appelé voisinage totalement normal dans [Car93]. La proximité des points est nécessaire comme le montre l'exemple de deux points diamétralement opposés sur une sphère munie de sa métrique usuelle, puisqu'il existe une infinité de géodésiques de longueur minimale (des demi-arcs de grand cercles) reliant ces points. Notons tout de même que si X est connexe et complet (comme espace topologique) le théorème de Hopf-Rinow [Car93] nous assure l'existence d'au moins un plus court chemin pour tout couple de points.

Ces considérations peuvent paraître éloignées de notre but, interpoler des courbes pour l'infographie, mais nous verrons par la suite qu'elles permettent effectivement dans des cas bien précis de définir une interpolation de courbe canonique simplifiant au maximum l'interaction avec l'utilisateur.

Après avoir défini un mode d'interpolation pour les points nous pouvons voir si cette interpolation s'étend simplement aux courbes. Rappelons tout d'abord ce que nous entendons par le mot courbe.

1.2 Définition d'une courbe

Le mot "courbe" désigne pour nous un objet bien précis qui est distinct du sens usuel. En effet, dans les livres de géométrie différentielle classique [BG87] une courbe est une variété différentielle de dimension 1, connexe, de classe $\mathcal{C}^1$ au moins. On sait d'ailleurs ([BG87], p

127) qu'une telle courbe est difféomorphe soit à la droite réelle soit au cercle et que dans ce cas elle est compacte.

Mais nous utiliserons le mot courbe dans le sens d'arc paramétré :

Un arc paramétré de classe $\mathcal{C}^p$ $(p \geq 1)$ *d'une variété* X *est la donnée d'une immersion* $\gamma \in \mathcal{C}^p(]0,1[, X)$.

La condition d'immersion assure qu'on peut définir l'espace tangent à la courbe et sa vitesse pour tout *paramètre* $t \in]0,1[$ mais n'interdit pas les autointersections.

Pour être plus précis nous utiliserons une définition restreinte plus adaptée à notre problème, à savoir un *arc compact* [LFA77], l'arc paramétré étant défini sur le segment [0,1] et non]0,1[(une courbe est donc l'immersion d'une variété de dimension 1 à bord, le bord étant constitué de ses deux extrémités, c'est à dire de la sphère S^0).

Remarquons que si X est un espace vectoriel, les courbes *ne forment pas* un sous-espace vectoriel de l'espace $\mathcal{C}^p([0,1], X)$. Ainsi la somme de deux courbes n'est a priori pas une courbe.

Ayant défini les objets de travail nous pouvons poser le problème de l'interpolation de courbes.

1.3 Position du problème

Le problème (P_c) de l'interpolation de courbes se pose ainsi :

Soient γ_0 *et* γ_1 *deux courbes de classe* $\mathcal{C}^p$*, comment définir une interpolation* $\gamma_t, t \in [0,1]$ *telle que :*

- $\gamma_{t=0} = \gamma_0$, $\gamma_{t=1} = \gamma_1$,
- *pour tout* t, γ_t *est une courbe de classe* $\mathcal{C}^p$,
- *l'application* $(t,s) \mapsto \gamma_t(s)$ *est continue.*

En particulier une interpolation suppose les deux courbes γ_0 et γ_1 homotopes (si nous voulions interpoler des courbes fermées sur un tore, cette dernière remarque a le mérite de montrer les limites de cette interpolation...).

Comme les courbes sont données par des paramétrisations, nous avons une correspondance naturelle $\gamma_0(s) \mapsto \gamma_1(s)$. On pourrait donc penser à définir $\gamma_t(s)$ par l'interpolation des points $\gamma_0(s)$ et $\gamma_1(s)$ comme proposé dans la section 1.1. Les exemples de la figure 1.1

montrent cependant que cette construction n'est pas acceptable : dans la figure de gauche la paramétrisation n'est pas injective, tandis que dans celle de droite elle n'est pas régulière. Ainsi l'extension du problème de l'interpolation des points à celui de l'interpolation

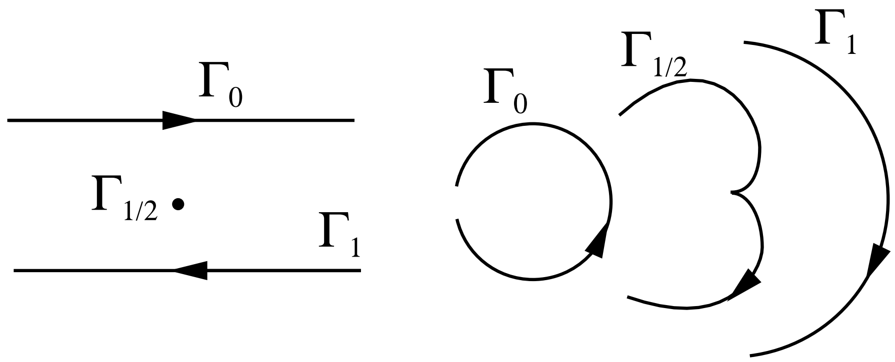


FIG. 1.1 – À gauche : interpolation au temps $1/2$ entre deux segments Γ_0 et Γ_1 du plan orientés dans le sens contraire. À droite : interpolation au temps $1/2$ entre deux arcs de cercle. La courbe interpolée est un arc de cardioïde.

des courbes n'est pas si immédiate qu'elle y paraît.

Avant de proposer une solution plus satisfaisante nous présentons rapidement dans la section 1.3.1 une autre approche. Nous supposons temporairement que l'espace ambiant X est l'espace euclidien $\mathbb{R}^3$. Le fait de nous restreindre à $\mathbb{R}^3$ nous permet de définir la forme d'une courbe par l'intermédiaire d'un groupe agissant sur l'espace. Ce groupe (géométrique) est ici le groupe des déplacements.
Nous caractériserons donc une courbe par sa forme et sa position.

1.3.1 Forme et position

Si on fait agir le groupe des déplacements de $\mathbb{R}^3$ sur une courbe γ on obtient un ensemble de courbes (l'orbite de γ par l'action des déplacements).
Cette classe de courbes constitue la **forme** de γ.
Une **position** est un élément particulier de cette classe ou encore un déplacement.

Pour interpoler une courbe nous pouvons ainsi tenter d'interpoler séparément sa forme et sa position. Or la forme d'une courbe de $\mathbb{R}^3$ est déterminée par deux fonctions appelées courbure et torsion. Plus précisément nous avons le théorème de représentation suivant ([BG87], p 339) :

Théorème 1.1 *Soit I un intervalle de $\mathbb{R}$, $\kappa \in C^1(I\,;\mathbb{R}_+^*)$ et $\tau \in C^0(I\,;\mathbb{R})$, et* $\mathbf{u}$, $\mathbf{v}$, $\mathbf{w}$, *trois* *points de E espace vectoriel euclidien de dimension 3. Il existe alors un arc géométrique C orienté de classe C^3 (nécessairement birégulier[1]) de E, et un seul dont une paramétrisation normale (I, γ) soit telle que*

$$\gamma(0) = \mathbf{u},\ \gamma^{'}(0) = \mathbf{v},\ \frac{\gamma^{''}(0)}{\|\gamma''(0)\|} = \mathbf{w}$$

et que κ et τ soient respectivement la courbure et la torsion de C en $\gamma(t)$ pour tout $t \in I$.

Ce théorème se généralise d'ailleurs en dimension quelconque ([Die71b], p 323).

Cet énoncé s'applique à des courbes paramétrées par l'abscisse curviligne (i.e données par une paramétrisation normale). Or dans notre cas, c'est l'intervalle de paramétrisation qui est fixé et non la vitesse. Il faut donc ajouter la vitesse (qui détermine la paramétrisation) à la courbure et à la torsion pour définir la forme de nos courbes.

Pour interpoler la forme d'une courbe nous pouvons interpoler séparément chacune de ces quantités. Cependant l'ensemble des triplets (v, κ, τ) de fonctions qui représentent une forme ne constitue pas un espace vectoriel. Par exemple, il n'existe pas de courbes à courbure nulle et torsion non nulle. Le théorème de représentation précédent ne s'applique d'ailleurs qu'aux courbes birégulières. Les triplets de fonctions correspondantes constituent certes un ensemble convexe, mais il serait dommage de ne pouvoir intégrer les segments de droite dans nos interpolations...

La notion de forme précédemment décrite n'est finalement pas très bien adaptée à notre problème. Intuitivement cela peut s'interpréter par le fait que les déplacements n'opèrent pas simplement (transitivement) sur chacune des orbites de courbes comme le montre le cas du segment de droite, puisque toutes les rotations admettant un axe colinéaire au segment et un centre sur la droite support de ce segment le laisse invariant.

Dans la section suivante nous développons un procédé d'interpolation de courbes qui remédie à ce problème. Pour cela nous nous plaçons dans un cadre plus général où nos courbes sont des arcs compacts à valeurs dans un groupe de Lie. Ceci nous permet de représenter les courbes comme des lignes intégrales de certains champs non singuliers. L'interpolation de ces champs fournit alors l'interpolation des courbes.

1.4 Interpolation de courbes

Dans toute cette partie I désigne l'intervalle compact $[0, 1]$, X une variété connexe de dimension finie n et de classe $\mathcal{C}^p$, $p > 2$ et $\mathcal{C}^p(I, X)$ l'ensemble des applications de I dans X de classe $\mathcal{C}^p$.

1. un arc est dit *birégulier* s'il admet une paramétrisation γ telle que γ' et γ'' sont indépendants

Cette section est structurée de la façon suivante :

- En premier lieu, dans la section 1.4.1, nous décrirons comment interpoler deux fonctions γ_0 et γ_1 de $\mathcal{C}^p(I,X)$,
- puis, dans la section 1.4.2, nous montrerons comment étendre ce procédé d'interpolation à des courbes, c'est-à-dire à des *immersions* du segment I (i.e. $\gamma_0' \neq 0$ et $\gamma_1' \neq 0$). Cette extension requiert la donnée d'un repère mobile adapté à chaque immersion. Nous décrirons donc comment choisir un tel repère et nous commenterons l'influence de ce choix lorsque X est un espace vectoriel euclidien.

1.4.1 Homotopie dans $\mathcal{C}^p(I,X)$

Nous cherchons à définir une interpolation entre deux fonctions, γ_0 et γ_1, de $\mathcal{C}^p(I,X)$, c'est-à-dire que nous cherchons une famille γ_t de fonctions telles que :

1. $\gamma_t \in \mathcal{C}^p(I,X), \forall t \in I$,
2. $\gamma_{t=0} = \gamma_0$ et $\gamma_{t=1} = \gamma_1$,
3. l'application $(s,t) \mapsto \gamma_t(s)$ est continue de $I \times I$ dans X.

Pour pouvoir se ramener à des interpolations linéaires dans des espaces vectoriels, nous allons considérer γ_0 et γ_1 comme des courbes intégrales de certains champs de vecteurs sur X et interpoler ces champs plutôt que les fonctions elles-mêmes. Pour cela nous associons à chaque fonction γ son **hodographe** c'est-à-dire la fonction de classe $\mathcal{C}^{p-1}$ à valeur dans le fibré tangent TX :

$$s \mapsto \gamma'(s) = T_s\gamma.1_{\mathbb{R}}.$$

Cet hodographe est donc le *relèvement canonique de γ dans TX.*

Puis nous interpolons les hodographes de γ_0 et γ_1 et nous en déduisons un champ de vecteurs $\mathbf{V}_t$ sur X dépendant du temps.

Par ailleurs, comme X est connexe, nous supposons donné un chemin $t \mapsto \gamma_t(0)$ joignant $\gamma_0(0)$ à $\gamma_1(0)$.

γ_t est alors obtenu en intégrant $\mathbf{V}_t$ sur I avec la condition initiale $\gamma_t(0)$.

Ce principe d'interpolation soulève plusieurs questions :

1. Comment comparer des vecteurs de TX qui sont dans des espaces tangents (i.e des fibres) différents ?
2. Comment interpoler des hodographes ?

3. Comment construire à partir d'un hodographe un champ de vecteurs dépendant du temps sur X ?

4. Comment être sûr que ce champ est intégrable ?

En répondant point par point à ces questions, nous verrons qu'il est nécessaire de faire des hypothèses supplémentaires sur X et que la notion de groupe de Lie permet de s'assurer de ces hypothèses.

1. Comparaison entre vecteurs tangents : afin de comparer des vecteurs qui n'appartiennent pas à un même espace tangent, ce qui revient à construire un isomorphisme entre espaces tangents, on peut munir la variété X d'un opérateur appelé *connexion*. Généralement cet opérateur ne permet de définir cet isomorphisme que de manière locale (on peut donc obtenir canoniquement une *trivialisation locale* en tout point de TX). Pour être capable de comparer *globalement* tous les vecteurs de TX nous devons supposer que TX est globalement trivialisable, c'est à dire qu'il existe un difféomorphisme qui envoie TX sur le produit $X \times \mathbb{R}^n$ et qui respecte la projection canonique de TX sur X.

En supposant que X est un groupe de Lie (connexe de dimension n) nous savons que cette trivialisation est possible et qu'elle peut être définie canoniquement.

Dans ce cas, en effet, chaque élément x du groupe X induit un automorphisme de X, $y \mapsto x^{-1}y$, que nous notons τ_x. Par différentiation, cet automorphisme induit lui-même un isomorphisme entre l'espace T_xX tangent en x et l'espace T_eX tangent en l'unité e du groupe[2]. Nous notons θ_x cet isomorphisme :

$$\theta_x = T_x\tau_x.$$

L'hodographe γ' d'une fonction γ peut ainsi être transporté par θ_γ en T_eX (voir figure 1.2) :

$$\gamma' \mapsto \mathbf{v} = \theta_\gamma.\gamma'. \tag{1.1}$$

2. Interpolation d'hodographes : en nous ramenant à un problème dans T_eX, nous pouvons définir l'interpolation des hodographes par l'interpolation des hodographes transportés.

Les deux fonctions γ_0 et γ_1 fournissent en effet deux hodographes transportés $\mathbf{v}_0$ et $\mathbf{v}_1$ en utilisant le transport défini par l'équation 1.1. $\mathbf{v}_0$ et $\mathbf{v}_1$ étant à valeurs dans l'espace vectoriel T_eX, nous pouvons alors les interpoler canoniquement à l'aide de l'interpolation linéaire :

$$\mathbf{v}_t(s) = (1-t)\,\mathbf{v}_0(s) + t\,\mathbf{v}_1(s).$$

2. On note souvent $\mathcal{G}_e$ cet espace par référence à la structure d'algèbre dont il peut être muni. Comme nous n'utilisons pas cette structure nous préférons conserver la notation plus classique T_eX.

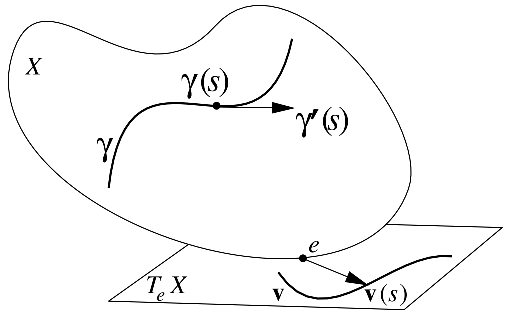


FIG. 1.2 – Transport du vecteur $\gamma'(s)$ en T_eX.

3. Construction d'un champ de vecteurs à partir d'un hodographe : un vecteur $\mathbf{v}$ de T_eX fournit un champ invariant pour le groupe, de la forme :

$$\mathbf{V}(x) = \theta_x^{-1}.\mathbf{v}.$$

À partir de la fonction interpolée $s \mapsto \mathbf{v}_t(s)$ on obtient donc par "translation" en tout point un champ de vecteurs dépendant du paramètre[3] s (voir figure 1.3) :

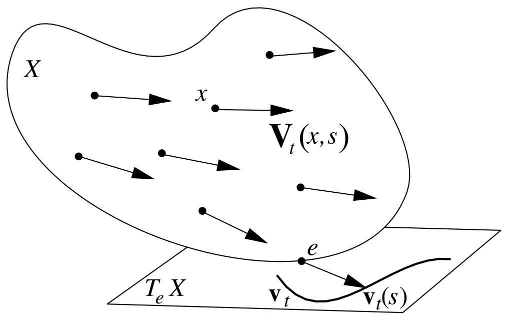


FIG. 1.3 – Le vecteur $\mathbf{v}_t(s)$ est transporté en tout point de X.

$$\mathbf{V}_t(x,s) = \theta_x^{-1}.\mathbf{v}_t(s).$$

4. Intégration de $\mathbf{V}_t$: pour une condition initiale $\gamma_t(0)$ donnée nous devons intégrer le système (de Cauchy) (E_t) suivant :

$$\begin{cases} \gamma'(s) = \mathbf{V}_t(\gamma(s), s) \\ \gamma(0) = \gamma_t(0) \end{cases}$$

3. Notons que la notion de champ dépendant d'un paramètre peut être ignorée si on considère le champ $(\mathbf{V}_t(x,s), 1_I)$ sur la variété produit $X \times I$.

En général, les courbes intégrales d'un système défini par un champ de vecteurs et une condition initiale ne sont pas définies sur tout intervalle :
prenons sur $\mathbb{R}$ l'exemple classique du champ $\mathbf{V}(x) = -x^2$. Le système correspondant de condition initiale x_0 a pour solution $s \mapsto \frac{x_0}{1+sx_0}$ qui est définie sur $]\frac{-1}{x_0}, +\infty[$ si x_0 est positif et $]-\infty, \frac{-1}{x_0}[$ sinon.

Afin de s'assurer que le système (E_t) a une solution, nous pouvons supposer que X est compact, ou bien que X admet une *représentation linéaire fidèle*, c'est à dire qu'il existe un morphisme injectif de groupe de Lie envoyant X dans $GL(n, \mathbb{R})$, le groupe linéaire des matrices inversibles de taille $n \times n$.

En effet,

- si X est compact, le problème de Cauchy (E_t) est intégrable sur I (c.f. [Mal72], §I.2.2),
- si X admet une représentation linéaire fidèle, alors nous montrons en annexe A que le système (E_t) se "transforme" en un système linéaire et la théorie des équations différentielles linéaires nous dit que ce système est intégrable (voir par exemple [BG87], §1.6).

Nous avons ainsi défini sur tout groupe de Lie connexe de dimension finie n qui est soit compact, soit représentable fidèlement dans $GL(n, \mathbb{R})$, un procédé canonique d'interpolation de deux fonctions γ_0 et γ_1 de $\mathcal{C}^p(I, X)$. Ce procédé nécessite la donnée d'un chemin reliant $\gamma_0(0)$ à $\gamma_1(0)$. Lorsque ces points sont suffisamment proches, on peut les relier, comme proposé dans la section 1.1, à l'aide de l'application exponentielle en $\gamma_0(0)$ qui est un difféomorphisme local (c.f. [Die71b], §19.8.6) de $T_{\gamma_0(0)}X$ sur X.

1.4.2 Homotopie régulière dans $\mathcal{C}^p(I, X)$

Nous cherchons désormais à définir une interpolation entre deux immersions de I de classe $\mathcal{C}^p$, c'est-à-dire que nous cherchons à résoudre le problème (P_c) de la section 1.3 que nous rappelons ci-dessous :
Comment trouver une famille γ_t de fonctions telles que :

1. γ_t est une immersion de I de classe $\mathcal{C}^p$ pour tout t de I,
2. $\gamma_{t=0} = \gamma_0$ et $\gamma_{t=1} = \gamma_1$,
3. l'application $(s, t) \mapsto \gamma_t(s)$ est continue de $I \times I$ dans X.

Le principe d'interpolation est une extension du cas précédent. Cependant dans le cas d'immersions, les hodographes transportés $\mathbf{v}_0$ et $\mathbf{v}_1$ précédemment définis, ne "passent" pas par le vecteur nul 0_e de T_eX et $\mathbf{v}_t$ ne doit pas non plus s'annuler. Nous devons donc effectuer une interpolation d'hodographes dans T_eX privé de 0_e. Comme cet ensemble n'est pas convexe nous ne pouvons plus utiliser l'interpolation linéaire pour $\mathbf{v}_t$. Pour y remédier nous allons remplacer les hodographes transportés par des fonctions à valeurs dans un groupe de Lie, et non plus dans T_eX privé de 0_e. Nous pourrons ainsi appliquer la méthode d'interpolation précédente à ces fonctions, puis nous en déduirons un hodographe $\mathbf{v}_t$. À partir de $\mathbf{v}_t$ nous pourrons procéder comme dans la section 1.4.1.

Afin de remplacer les hodographes par d'autres fonctions, nous enrichissons la structure de notre groupe de Lie en le munissant d'une métrique invariante, ce qui revient à munir T_eX d'un produit scalaire. $T_eX \setminus \{0_e\}$ s'identifie alors, comme sous-variété de T_eX, au produit $\mathbb{R}_+^* \times S_e^{n-1}$ par l'application :

$$\mathbf{v} \mapsto (\|\mathbf{v}\|, \frac{\mathbf{v}}{\|\mathbf{v}\|})$$

Nous pourrions alors effectuer l'interpolation séparément sur $\mathbb{R}_+^*$, qui est convexe, et sur la sphère unité S_e^{n-1} de T_e. Mais, sauf pour des valeurs particulières de n (2 et 4), cette sphère ne peut être un groupe de Lie et nous ne pouvons appliquer la méthode précédente.

Pour cela, nous remplaçons le couple $(\|\mathbf{v}\|, \frac{\mathbf{v}}{\|\mathbf{v}\|})$ associé à une courbe γ (c.f équation 1.1), par un couple $(\|\mathbf{v}\|, r)$, où r correspond au transport dans T_eX d'un repère mobile R *adapté*[4] à γ et orthonormé. En effet, après avoir défini un repère adapté R pour une courbe γ nous pouvons transporter ce repère dans T_eX comme pour l'équation de transport 1.1 :

$$R(s) = (\mathbf{e}_1(s), \mathbf{e}_2(s), \mathbf{e}_3(s)) \mapsto r(s) = \theta_{\gamma(s)}.R(s) = (\theta_{\gamma(s)}.\mathbf{e}_1(s), \theta_{\gamma(s)}.\mathbf{e}_2(s), \theta_{\gamma(s)}.\mathbf{e}_3(s)).$$

Le choix d'une métrique invariante sur X nous assure que $\theta_{\gamma(s)}$ est une isométrie et donc que $r(s)$ est orthonormé.

En fixant une base orthonormée de T_eX, nous pouvons identifier $r(s)$ à un élément de $SO(n, \mathbb{R})$. Nous pouvons donc interpoler les repères r_0 et r_1, correspondant aux courbes γ_0 et γ_1, sur le groupe de Lie connexe $SO(n, \mathbb{R})$ à l'aide de la première méthode d'interpolation. Remarquons qu'un changement de base dans T_eX se traduit par une multiplication par une matrice orthogonale à coefficients constants et ne modifie pas le résultat de l'interpolation, ce qui montre le caractère intrinsèque de l'interpolation des repères. Par contre, comme précisé dans la première méthode d'interpolation, il est nécessaire de spécifier une courbe $r_t(0)$ reliant $r_0(0)$ à $r_1(0)$.

On en déduit ensuite un hodographe interpolé (dans T_eX) en multipliant le premier vecteur du repère interpolé r_t par la vitesse interpolée v_t. Notons que **l'interpolation linéaire de la vitesse conduit à l'interpolation linéaire de la longueur**.

4. c'est à dire que le premier vecteur de R est $\frac{\gamma'}{\|\gamma'\|}$.

Pour compléter le procédé d'interpolation, il ne nous reste donc plus qu'à décrire comment associer un repère mobile adapté à une courbe tracée sur X.

Détermination d'un repère mobile adapté

Pour caractériser un repère adapté à une courbe, nous allons examiner sa matrice de Cartan.

Si $R = (\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ est un repère mobile orthonormé et adapté à la courbe γ (i.e. $\mathbf{e}_1 = \frac{\gamma'}{\|\gamma'\|}$), on peut lui associer sa matrice de Cartan $(\omega_{i,j})_{1 \leq i,j \leq n}$ où :

$$\omega_{i,j} = \langle \nabla_{\mathbf{e}_1} \mathbf{e}_i, \mathbf{e}_j \rangle$$

et où ∇ désigne la dérivation covariante associée à la métrique $\langle ., . \rangle$ sur X ([Mal72], §8.4). En particulier, cela signifie que pour tout champ de vecteur $\mathbf{A}, \mathbf{B}, \mathbf{C}$, on a :

$$\mathbf{A}\langle \mathbf{B}, \mathbf{C} \rangle = \langle \nabla_{\mathbf{A}} \mathbf{B}, \mathbf{C} \rangle + \langle \mathbf{B}, \nabla_{\mathbf{A}} \mathbf{C} \rangle \tag{1.2}$$

où $\mathbf{A}f$ désigne la dérivation de la fonction réelle f relativement au champ $\mathbf{A}$. Comme R est orthonormé, on en déduit que sa matrice de Cartan est antisymétrique.

Pour choisir un repère adapté nous pouvons ainsi imposer une forme particulière de matrice antisymétrique et voir si cette matrice peut être la matrice de Cartan d'un repère de γ. Nous donnons et commentons deux exemples classiques :

Le repère de Frenet : nous choisissons une matrice antisymétrique de la forme :

$$\begin{aligned} \omega_{i,j} &= 0 \quad \text{si } |i - j| \neq 1 \\ \omega_{i,i+1} &> 0 \end{aligned}$$

Dans [Die71b], p 323, on montre qu'il existe "en général" un unique repère adapté à γ qui admette une telle matrice de Cartan. Dans ce cas, ce repère est le repère de Frenet de γ et $\omega_{i,i+1}$ est appelé la $i^{\text{ème}}$ courbure de γ. Dans $\mathbb{R}^3$, par exemple, "en général" signifie que γ est birégulière.

Le repère du transport relativement parallèle : un autre repère très employé en infographie est le repère du transport relativement parallèle, ou RTRP, introduit dans [Bis75] dans le cas des espaces euclidiens. Par rapport au repère de Frenet il a l'avantage d'être défini pour toute courbe immergée. Par contre il est défini à un paramètre près : la donnée d'un repère adapté "initial" en un point de la courbe (par exemple en une extrémité).

Nous définissons le RTRP par sa matrice de Cartan, puis nous montrons l'existence de tels repères pour toute courbe γ.

Définition 1.2 *Un repère orthonormé et adapté à une courbe γ est un RTRP si sa matrice de Cartan est de la forme :*

$$\omega_{i,j} = 0 \qquad \text{si } i \neq 1 \text{ et } j \neq 1.$$

Ceci signifie que, pour tout $i \neq 1$, $\nabla_{\mathbf{e}_1}\mathbf{e}_i$ est parallèle à $\mathbf{e}_1$, d'où la dénomination de ce repère. En utilisant l'équation 1.2 de compatibilité de ∇ avec la métrique, on voit que l'on a nécessairement :

$$\nabla_{\mathbf{e}_1}\mathbf{e}_i = -\langle \nabla_{\mathbf{e}_1}\mathbf{e}_1, \mathbf{e}_i \rangle \mathbf{e}_1. \tag{1.3}$$

Remarquons que cette équation est une extension de l'équation

$$f' = -\langle \gamma'', f \rangle \gamma'$$

proposée dans [Klo86]. Elle montre en particulier que le transport relativement parallèle, ou TRP, coïncide avec le transport parallèle (i.e $\nabla_{\mathbf{e}_1}\mathbf{e}_i = 0$) si et seulement si γ est une géodésique.

Remarque 1.1 *Le TRP conserve les angles (entre deux champs relativement parallèles ou bien entre un tel champ et $\mathbf{e}_1$) et la norme. C'est une conséquence directe de l'équation de compatibilité 1.2 et de l'équation 1.3.*

De plus l'équation 1.3 permet effectivement de définir un repère pour une condition initiale donnée.

Proposition 1.1 *Pour toute courbe γ et tout repère R_0 adapté à $\gamma(0)$, il existe un unique RTRP R tel que $R(0) = R_0$.*

Démonstration : l'existence et l'unicité d'un champ relativement parallèle, pour une condition initiale donnée, se montre de la même manière que pour un champ parallèle (c.f. [Fer88], p 140) :

En tout point de γ, on peut par une carte locale de X, retranscrire l'équation 1.3 en une équation différentielle linéaire du premier ordre de la forme

$$\frac{\mathrm{d}\mathbf{E}_i}{\mathrm{d}s} = A.\mathbf{E}_i$$

où $\mathbf{E}_i$ représente $\mathbf{e}_i$ dans la carte, et A est une matrice carrée dont les coefficients font intervenir les composantes de $\frac{\mathrm{d}\mathbf{E}_1}{\mathrm{d}s}$ et ce qu'on appelle classiquement les symboles de Christoffel associés à ∇ dans la carte choisie. L'existence du TRP est donc assurée localement.

Nous pouvons ensuite procéder comme dans [Bis75] : comme l'image d'un arc compact est compacte, nous pouvons la recouvrir par un nombre fini de cartes et raccorder les solutions trouvées dans chaque carte. L'unicité pour une condition initiale donnée découle alors de la propriété de conservation des angles de la remarque 1.1$\square$

Remarque 1.2 *Le RTRP est caractérisé par une matrice de Cartan de norme minimale.*

Démonstration : si $R = (\mathbf{e}_1, \ldots, \mathbf{e}_n)$ est un repère mobile orthonormé et adapté à la courbe γ, et $(\omega_{i,j})_{1 \leq i,j \leq n}$ est sa matrice de Cartan, nous avons :

$$\sum_{i,j} \omega_{i,j}^2 = \sum_{i=1 \,\mathrm{ou}\, j=1} \omega_{i,j}^2 + \sum_{i \neq 1 \,\mathrm{et}\, j \neq 1} \omega_{i,j}^2$$

Comme $\omega_{1,i} = -\omega_{i,1} = \langle \nabla_{\mathbf{e}_1} \mathbf{e}_1, \mathbf{e}_i \rangle$ et que R est orthonormé, on a

$$\sum_{i,j} \omega_{i,j}^2 = 2 \|\nabla_{\mathbf{e}_1} \mathbf{e}_1\|^2 + \sum_{i \neq 1 \,\mathrm{et}\, j \neq 1} \omega_{i,j}^2$$

La définition 1.2 du RTRP permet de conclure □

Lien entre l'interpolation et les matrices de Cartan

On peut se demander si cette interpolation revient à interpoler les matrices de Cartan des repères, c'est à dire si

$$r' = r.(\omega_{i,j}).$$

La réponse est négative dans le cas général.
En effet, en supposant pour simplifier que γ est paramétrée par l'abscisse curviligne, les coefficients de $r^{-1}.r'$ sont de la forme :

$$\Omega_{i,j} = \langle (\theta_\gamma.\mathbf{e}_i)', \theta_\gamma.\mathbf{e}_j \rangle$$

tandis que

$$\omega_{i,j} = \langle \nabla_{\mathbf{e}_1} \mathbf{e}_i, \mathbf{e}_j \rangle = \langle \theta_\gamma.\nabla_{\mathbf{e}_1} \mathbf{e}_i, \theta_\gamma.\mathbf{e}_j \rangle$$

Or dans le cas général $\Omega_{i,j} \neq \omega_{i,j}$ même si R est un RTRP. On peut le vérifier par exemple dans $SO(3, \mathbb{R})$ sur des trajectoires non géodésiques.

Le cas euclidien : par contre, lorsque X est un espace euclidien on a toujours $\Omega_{i,j} = \omega_{i,j}$. Ceci entraîne des propriétés intéressantes pour notre interpolation.

Propriété 1.1 *Si les deux courbes γ_0 et γ_1 sont munies du RTRP (resp. du repère de Frenet) alors la courbe interpolée est automatiquement munie du RTRP (resp. du repère de Frenet).*

Propriété 1.2 *De plus, la courbure interpolée est majorée par une fonction monotone joignant la courbure κ_0 de γ_0 à la courbure κ_1 de γ_1.*

Démonstration :

1. En effet, les matrices de Cartan des repères de Frenet ou bien des RTRPs forment des ensembles convexes dans l'ensemble des matrices de Cartan (et même un espace vectoriel dans le cas des RTRPs), l'interpolation linéaire de ces matrices fournit donc une matrice de même type.

2. De plus la courbure d'une courbe γ vaut par définition :

$$\kappa = \|\nabla_{\mathbf{e}_1}\mathbf{e}_1\|.$$

Si γ_0 et γ_1 sont paramétrées par l'abscisse curviligne, le premier vecteur de la matrice de Cartan est justement le vecteur $\nabla_{\mathbf{e}_1}\mathbf{e}_1$ exprimé dans le repère type correspondant, on en déduit à l'aide de l'inégalité triangulaire que la courbure κ_t de la courbe interpolée vérifie :

$$\kappa_t \leq (1-t)\,\kappa_0 \,+\, t\,\kappa_1.$$

Cette inégalité devient d'ailleurs une égalité vérifiée par *toutes* les courbures dans le cas du repère de Frenet.

Si γ_0 et γ_1 ne sont pas paramétrées par l'abscisse curviligne nous devons tenir compte de la vitesse. Nous avons en effet :

$$\mathbf{e}_i'(s) = \nabla_{\gamma'(s)}\mathbf{e}_i(s) = \nabla_{\|\gamma'(s)\|\mathbf{e}_1(s)}\mathbf{e}_i(s) = \|\gamma'(s)\|\nabla_{\mathbf{e}_1(s)}\mathbf{e}_i(s)$$

ce qui montre que

$$\Omega_{i,j} = v\,\omega_{i,j}$$

où l'on a posé $v = \|\gamma'\|$. L'interpolation se traduit donc par une interpolation du produit de la vitesse par la(les) courbure(s). Comme la vitesse est interpolée linéairement on a

$$\kappa_t \leq \frac{(1-t)\,v_0\kappa_0 + t\,v_1\kappa_1}{v_t} = \kappa_0 \,+\, \frac{1}{(1-\frac{v_0}{v_1}) + \frac{1}{t}\frac{v_0}{v_1}}\,(\kappa_1-\kappa_0), \quad t \neq 0$$

ce qui montre que l'interpolation de la courbure est majorée par une fonction monotone valant κ_0 en 0 (resp. κ_1 en 1) (en particulier $\kappa_t \leq \max(\kappa_0,\kappa_1)$).

□

1.5 Conclusion

Nous avons ainsi défini sur tout groupe de Lie connexe, compact ou représentable fidèlement dans un groupe linéaire, et muni d'une métrique invariante, un procédé d'interpolation entre deux immersions γ_0 et γ_1 du segment $[0,1]$. Nous avons pour cela muni chaque courbe γ_0

et γ_1 d'un repère mobile adapté R_0 et R_1. En considérant les matrices de Cartan des repères, nous avons proposé l'utilisation du repère de Frenet ou du repère du transport relativement parallèle (RTRP).

Ce procédé nécessite la spécification d'un chemin reliant $\gamma_0(0)$ à $\gamma_1(0)$ et d'un chemin reliant $r_0(0)$ à $r_1(0)$ où r_0 et r_1 sont les repères R_0 et R_1 transportés en l'identité du groupe. De plus $R_0(0)$ et $R_1(0)$ doivent également être spécifiés si l'interpolation utilise le RTRP.

Lorsque X est un espace euclidien, $\gamma_0(0)$ et $\gamma_1(0)$ peuvent canoniquement être reliés par le segment qui les joint. De plus,

- la courbe interpolée est automatiquement munie d'un repère mobile du même type que celui de γ_0 et de γ_1,

- sa longueur totale est une fonction linéaire des longueurs de γ_0 et de γ_1,

- et sa courbure est majorée par une fonction monotone joignant les courbures de γ_0 et γ_1.

Cette dernière propriété s'avèrera utile lorsque nous décrirons dans la troisième et dernière partie notre méthode de métamorphose.

Nous avons restreint notre étude à des groupes de Lie car ils sont munis d'un parallélisme canonique qui permet de comparer les hodographes de chaque courbe γ_0 et γ_1. Nous les avons choisis connexes car il est nécessaire de relier $\gamma_0(0)$ à $\gamma_1(0)$. Enfin, afin de pouvoir intégrer l'hodographe interpolé nous avons imposé à ce groupe de Lie d'être compact ou de posséder une représentation linéaire fidèle. Une étude plus poussée des groupes de Lie sort largement du cadre de cette thèse, mais pour justifier le langage mathématique que nous avons employé, il faut s'assurer que ces restrictions sont malgré tout vérifiées par un grand nombre d'objets. Or les travaux de classification des groupes de Lie [Die75] montrent que tous les groupes de Lie connexes et commutatifs admettent une représentation fidèle (c'est également le cas des groupes compacts et simplement connexes mais la compacité suffit à nos besoins). Comme exemples classiques de groupes de Lie répondant à nos besoins citons les espaces vectoriels de dimension finie, les tores, les sphères S^1 et S^3, tous les sous-groupes connexes de $GL(n, \mathbb{R})$ et tous les produits de ces groupes...

L'extension de ce procédé d'interpolation à des variétés riemaniennes connexes et compactes peut se faire en utilisant l'application exponentielle et le parallélisme canonique le long d'une courbe :

1. on construit un repère adapté R_0 le long de γ_0 (resp. R_1 le long de γ_1),

2. on transporte ce repère en $T_{\gamma_0(0)}X$ (resp. $T_{\gamma_1(0)}X$) en utilisant le parallélisme induit par la courbe γ_0 (resp. γ_1).

3. Puis les deux repères sont à nouveau transportés en $\gamma_t(0)$, en utilisant le parallélisme le long de l'arc joignant $\gamma_0(0)$ à $\gamma_1(0)$, pour être interpolés.

4. On en déduit un hodographe $\mathbf{v}_t$ que l'on transporte localement en tout point en utilisant l'application exponentielle en $\gamma_t(0)$.

5. on intègre ce champ et on transporte $\mathbf{v}_t$ en l'extrémité $\gamma_t(s_1)$ de la courbe intégrale trouvée. On transporte à nouveau ce champ en utilisant l'application exponentielle en $\gamma_t(s_1)$ et on recommence...

Ce processus a une fin : ceci peut être montré en utilisant le fait que la vitesse interpolée est majorée par les vitesses des courbes γ_0 et γ_1 et que, sur une variété compacte, le domaine de l'application exponentielle contient au moins un disque de rayon fixé (appelé le rayon d'injectivité) et non nul en tout point. En notant v_{max} un majorant de la vitesse et ϵ le rayon d'injectivité, on doit répéter au plus $\frac{v_{max}}{\epsilon}$ fois l'opération 5.

Chapitre 2

Un algorithme d'interpolation de courbes polygonales

Dans le premier chapitre nous avons défini un procédé d'interpolation de courbes dans un cadre mathématique où les objets sont de nature continue (du cardinal de $\mathbb{R}$). Or, l'outil principal des infographistes, l'ordinateur, ne permet pas de manipuler des objets de taille infinie, de sorte qu'au lieu de travailler sur des objets de type sous-variétés de $\mathbb{R}^n$, nous devons définir des *modèles* discrets pour ces objets et voir de quelle manière les propriétés connues dans $\mathbb{R}^n$ se "transposent" sur ces modèles.

Par exemple dans [KKM91], on étudie la topologie discrète sur $\mathbb{Z}^2$ (et dans une moindre mesure sur $\mathbb{Z}^3$), on définit ce que peut être une courbe dans cet espace et on obtient ensuite sur ce modèle de courbe un équivalent du théorème de Jordan.

Dans le domaine de la topologie et de la géométrie discrète on s'intéresse d'ailleurs souvent à l'obtention, dans $\mathbb{Z}^n$ ou dans des espaces finis, de propriétés *analogues* à celles que l'on connaît sur $\mathbb{R}^n$ [Uni93]. Ainsi J.P. Réveillès de l'université Louis Pasteur à Strasbourg travaille sur la notion de droites discrètes dans $\mathbb{Z}^2$. Ces dernières sont définies par des inéquations à coefficients entiers appelées équations diophantiennes [Rév93]. Ces droites ne possèdent pas les mêmes propriétés que les droites usuelles du plan euclidien ; en particulier elles possèdent une épaisseur. Cependant, il est possible de définir certaines notions telles que l'intersection de deux droites discrètes et d'établir un lien avec les propriétés classiques de la géométrie euclidienne. Dans ce cas précis le lien est obtenu à partir d'une construction mathématique : l'analyse non-standard [CNR92], qui permet de créer un "pont" entre l'arithmétique sur $\mathbb{Z}^2$ et la géométrie euclidienne sur $\mathbb{R}^2$.

Nous ne nous placerons pas dans l'optique de la géométrie discrète, néanmoins nous dégagerons une notion de modèle pour faire la liaison entre les raisonnements "en continu" et les algorithmes utilisés dans l'ordinateur. Le chapitre s'organise de la manière suivante :

- Nous présentons une notion générale de modèle après avoir rappelé les approches existantes,
- nous proposons un modèle discret de courbes,
- puis nous présentons un algorithme d'interpolation de modèles de courbe par analogie avec la méthode décrite dans le premier chapitre.

Les résultats des sections 2.2 à 2.4 ont été publiés sous forme d'un rapport INRIA [Laz95].

2.1 Du modèle continu au modèle discret

2.1.1 La notion de modèle en modélisation géométrique

Dans le domaine de l'infographie, et tout particulièrement en modélisation géométrique, les objets manipulés sont de nature géométrique et topologique et correspondent à la notion mathématique de variété différentielle. Ces objets sont couramment représentés par des polygones, des polyèdres ou des assemblages de carreaux de spline [Ris91]. Pour justifier ces choix, une réflexion sur la notion de modèle apparaît souvent en introduction de la Géométrie Solide Constructive ou CSG ([Hof89], §4.4 : Representation and Model).

Dans [GV95], qui reprend [Req80], on introduit les notions *d'univers physique, d'univers mathématique et d'univers de représentation.* Appliqué à la modélisation, l'univers physique est l'ensemble des formes de $\mathbb{R}^3$. Les formes sont décrites de manière implicite, par des fonctions d'appartenance. L'univers mathématique, noté M, est alors le sous ensemble des formes qui peuvent être décrites par une fonction calculable. Ceci mène les auteurs à définir M comme l'ensemble des ensembles semi-algébriques (ensembles décrits par des inéquations polynomiales). Le théorème de Stone-Weirstrass est invoqué pour justifier la puissance de description de l'univers physique par M.

Les différents processus de codage des fonctions d'appartenance des objets de M constituent autant d'univers de représentation. Un bon univers de représentation R est un univers en bijection r avec M et tel que, pour toute opération L dans M, il existe une opération L_1 dans R qui rende commutatif[1] le diagramme suivant :

$$\begin{array}{ccc} M & \xrightarrow{r} & R \\ L \downarrow & & \downarrow L_1 \\ M & \xrightarrow{r} & R \end{array}$$

1. on dit qu'un diagramme est commutatif lorsque la composition des flèches qui vont d'un objet à un autre ne dépend pas du chemin orienté suivi. Dans le diagramme ci-dessus cela signifie que $L_1 \circ r = r \circ L$.

Ce diagramme traduit une bonne adéquation entre le modèle mathématique et sa représentation, et il exprime, en particulier, qu'on peut associer à tout procédé de construction mathématique dans M un *algorithme* dans R.

Notons que dans [GV95], on se place dans le cadre d'un ordinateur capable de travailler sur les nombres réels.

En choisissant de ne s'intéresser, au départ, qu'à un certain type d'objets mathématiques représentables dans l'ordinateur (les ensembles semi-algébriques), la présentation de [GV95] a le défaut, nous semble-t-il, d'ignorer les problèmes numériques.

Nous présentons une approche originale qui, à notre avis, a le mérite de mieux mettre en valeur les questions de convergence et d'approximation.

2.1.2 Modèle et classes d'équivalence

Si nous cherchons à formuler de façon grossière le problème principal de l'infographie, nous pouvons dire qu'il s'agit de reproduire à l'aide de l'ordinateur des phénomènes physiques ou des opérations mathématiques que nous imaginons de manière continue. C'est-à-dire, en nous inspirant de la définition du mot *simulation* dans le petit Larousse, que nous cherchons à remplacer un système à étudier par un modèle (informatique) plus simple ayant un comportement analogue.

Une des difficultés réside dans la définition d'un *modèle* et d'un *comportement analogue.* Nous proposons de donner un sens plus précis à ces notions. Notons qu'il existe une théorie des modèles fondées sur la notion de *catégorie* (c.f. [Hus94]). L'emploi de cette théorie nécessite un langage et un formalisme qui nous semblent disproportionnés en comparaison de l'apport possible de ce cadre théorique. Notre objectif est en effet de définir des notions suffisamment simples pour être opérantes et suffisamment générales pour être applicables à l'ensemble des problèmes que nous avons à traiter.

Dans le domaine de la modélisation d'objets, auquel nous nous rattachons, il est souvent plus approprié d'exprimer les problèmes à résoudre en termes d'opérations mathématiques sur certaines classes d'objets, plutôt qu'en termes de simulation physique. Nous nous intéresserons donc plus spécifiquement à un univers mathématique.

Comme il est rappelé dans [GV95], cet univers ne peut être représenté de façon biunivoque dans l'ordinateur. Seules certaines classes d'objets admettent une représentation non ambiguë, c'est-à-dire peuvent être codées dans l'ordinateur de sorte que chaque objet possède un code permettant de retrouver *toute* l'information qui sert à le définir.

Nous notons $\mathcal{M}$ l'univers des objets mathématiques et $\mathcal{R}$ l'ensemble des objets représentables[2]. Le travail du concepteur de logiciel peut alors s'exprimer comme suit : *Comment construire un algorithme A dans $\mathcal{R}$ qui soit analogue à une opération L dans $\mathcal{M}$.*

Exemple 2.1 *On peut chercher à simuler l'opération L de dérivation de l'ensemble $\mathcal{C}^p([0,1])$, des fonctions réelles de classe $\mathcal{C}^p$, dans l'ensemble $\mathcal{C}^{p-1}([0,1])$.*

Pour résoudre ce problème il faut définir des modèles informatiques pour $\mathcal{C}^p([0,1])$ et $\mathcal{C}^{p-1}([0,1])$, et construire un algorithme A sur ces modèles qui corresponde à L.

Comme il y a a priori moins d'objets dans $\mathcal{R}$ que dans $\mathcal{M}$, il paraît naturel d'associer à un élément de $\mathcal{R}$ un sous-ensemble, ou une classe, d'éléments de $\mathcal{M}$. Nous allons ainsi considérer les éléments de $\mathcal{R}$ comme des représentants de certaines classes d'équivalences sur $\mathcal{M}$. Nous proposons ainsi la notion de modèle suivante :

Définition 2.1 *Soit $\mathcal{C}$ une classe d'objets de $\mathcal{M}$, nous appelons* **modèle** *pour $\mathcal{C}$ une suite $(\sim_n, d_n)_{n\in\mathbb{N}}$ de relations d'équivalences $\sim_n$ sur $\mathcal{C}$ et de distances d_n sur le quotient $\mathcal{C}/_{\sim_n}$ telles que*

1. *toute classe de $\sim_n$ admet au moins un représentant dans $\mathcal{R}$ (i.e. un représentant représentable !),*

2. *pour tous objets $\mathcal{O}$ et $\mathcal{O}'$ de $\mathcal{M}$ et toute suite de représentants $(\mathcal{R}_n \,|\, \mathcal{R}_n \in \mathcal{C}/_{\sim_n})_{n\in\mathbb{N}}$ vérifiant*
$$\lim_{n\to\infty} d_n(\mathcal{O}_n, \mathcal{R}_n) = \lim_{n\to\infty} d_n(\mathcal{O}'_n, \mathcal{R}_n) = 0$$
on a $\mathcal{O} = \mathcal{O}'$ (on a respectivement noté $\mathcal{O}_n$ et $\mathcal{O}'_n$ les représentants de $\mathcal{O}$ et $\mathcal{O}'$ modulo $\sim_n$).

La première condition traduit la compatibilité du modèle avec le codage informatique, tandis que la seconde permet de considérer les éléments de $\mathcal{C}$ comme des limites de suites de représentants (représentables).
Par la suite nous notons $\mathcal{O} \xrightarrow{\sim_n} \mathcal{O}_n$ la projection canonique de $\mathcal{O}$ dans $\mathcal{C}/_{\sim_n}$ et $(\mathcal{O}_n)_{n\in\mathbb{N}} \xrightarrow{lim_\sim} \mathcal{O}$ le passage à la limite (lorsqu'elle existe).
Nous pouvons alors définir une notion d'algorithme relatif à une opération et des modèles.

Définition 2.2 *Soient $\mathcal{C}$ et $\mathcal{C}'$ deux classes de $\mathcal{M}$ pour lesquelles nous possédons des modèles respectifs $(\sim_n, d_n)_{n\in\mathbb{N}}$ et $(\sim'_n, d'_n)_{n\in\mathbb{N}}$. Soit L une opération $\mathcal{O} \mapsto \mathcal{O}'$ de $\mathcal{C}$ dans $\mathcal{C}'$, nous dirons qu'un algorithme sur $\mathcal{R}$ est* **analogue** *à l'opération L relativement aux modèles $(\sim_n$*

2. Dans [GV95] ce que nous notons $\mathcal{M}$ est appelé l'univers physique, tandis que ce que nous notons $\mathcal{R}$ correspond à l'univers mathématique...

$, d_n)_{n\in\mathbb{N}}$ *et* $(\sim'_n, d'_n)_{n\in\mathbb{N}}$ *si le diagramme suivant est commutatif:*

$$\begin{array}{ccc} \mathcal{O} & \xrightarrow{\sim_n} & (\mathcal{O}_n)_{n\in\mathbb{N}} \\ L\downarrow & & \downarrow A \\ \mathcal{O}' & \xleftarrow{lim_{\sim'}} & (A.\mathcal{O}_n)_{n\in\mathbb{N}} \end{array}$$

Ce qui exprime que $\mathcal{O}'$ peut être obtenu soit directement par l'opération L, soit en appliquant l'algorithme A à des "modèles" $\mathcal{O}_n$ de plus en plus précis.

Grâce à cette définition il devient possible de comparer l'efficacité de deux algorithmes A_1 et A_2 en fonction de la vitesse de convergence des suites $(A_1.\mathcal{O}_n)_{n\in\mathbb{N}}$ et $(A_2.\mathcal{O}_n)_{n\in\mathbb{N}}$.

Une classe d'objets mathématiques n'admet pas nécessairement un modèle. D'un autre côté il semble assez vain de parler d'algorithme analogue à une opération sur des ensembles qui n'admettent pas de modèles. Afin d'illustrer l'importance du modèle par rapport à l'algorithme nous détaillons ci-dessous un exemple.

Exemple 2.2 *Replaçons nous dans le cadre de l'exemple 2.1 où* $\mathcal{C} = \mathcal{C}^p([0,1])$, $\mathcal{C}' = \mathcal{C}^{p-1}([0,1])$ *et* L *est la dérivation d'ordre 1. Nous choisissons pour* $\mathcal{C}$ *le modèle suivant:*

- $f \sim_n g$ *si et seulement si*

  $$f^{(k)}(\frac{i}{n}) = g^{(k)}(\frac{i}{n}),\ \forall\, i = 0,\ldots,n\ \ et\ \forall\, k = 0,\ldots,m$$

  où m *est un entier fixé inférieur ou égal à* p.

  Un représentant $\mathcal{R}_n$ *d'une classe peut donc s'identifier à un (m+1)(n+1)-uplet de réels* $(r_i^k)_{0\le i\le n, 0\le k\le m}$

- *nous posons alors*

  $$d_n(f,\mathcal{R}_n) = \sqrt{\frac{1}{n}\sum_{i,k}(f^{(k)}(\frac{i}{n}) - r_i^k)^2}$$

Ici, la validité du modèle tient au fait que:

$$\lim_{n\to\infty} d_n(f,g) = \sqrt{\sum_k \int_0^1 (f^{(k)} - g^{(k)})^2}$$

Nous choisissons un modèle du même type pour $\mathcal{C}'$ *avec un entier* m' *strictement inférieur à* p.

- *Si* $m \ge 1$ *et* $m' < m$ *nous pouvons définir simplement l'algorithme* A *par:*

  $$A.(r_i^k)_{0\le i\le n, 0\le k\le m} = (r_i^l)_{0\le i\le n, 1\le l\le m'+1}$$

  Dans ce cas l'algorithme est clairement optimal.

- *Si $m = m' = 0$ nous pouvons définir A par :*

$$A.(r_i^0)_{0\leq i\leq n} = ({r'}_i^0)_{0\leq i\leq n}$$

avec

$$\begin{cases} {r'}_i^0 = n(r_{i+1}^0 - r_i^0) \ si \ i < n \\ {r'}_n^0 = {r'}_{n-1}^0 \end{cases}$$

Nous pouvons également remplacer l'expression de ${r'}_i^0$ par une formule du même type, telle que $\frac{n}{2}(r_{i+1}^0 - r_{i-1}^0)$. Aucune de ces méthodes n'est optimale et la vitesse de convergence dépend de l'ordre d'approximation de la dérivée.

Sur cet exemple trivial nous voyons que la richesse du modèle influe sur le choix de l'algorithme et sur son efficacité.

Notons par ailleurs que nous avons implicitement supposé l'ordinateur capable de manipuler des nombres réels. Nous pouvons cependant nous passer de cette hypothèse en remplaçant la définition de $\sim_n$ par une relation du type[3] :
$f \sim_n g$ si $f^{(k)}(\frac{i}{n})$ et $g^{(k)}(\frac{i}{n})$ ont un même développement décimal à l'ordre n.

2.1.3 Un modèle de courbe

Nous présentons un modèle discret pour les courbes définies en 1.2, c'est-à-dire les arcs compacts définis sur [0,1]. Ce modèle est une variante du modèle présenté dans l'exemple 2.2 précédent. Nous en donnons tout d'abord une justification plus intuitive.

Nous choisissons de représenter la topologie, l'immersion, la paramétrisation et la structure différentielle au premier ordre. Nous explicitons dans ce qui suit ce que nous entendons par la représentation de ces propriétés.

- **la topologie :** nos courbes sont définies par un ensemble de sommets, définir une topologie revient à se donner un graphe sur ces sommets. Comme nous voulons modéliser des courbes non fermées, un tel graphe correspond à la donnée d'un ordre sur les sommets.

- **l'immersion :** elle consiste ici à associer trois réels à chaque sommet (ses coordonnées dans $\mathbb{R}^3$), de sorte que deux sommets adjacents (ou successifs) n'aient pas les mêmes coordonnées.

- **la paramétrisation :** les n + 1 sommets $(\Gamma_i)_{0\leq i\leq n}$ étant ordonnés, nous choisissons de donner le paramètre i/n au point Γ_i. En d'autres termes nous choisissons un échantillonnage uniforme de [0,1].

3. Il n'est pas possible de définir une relation d'équivalence à partir d'une distance (i.e. $f \sim_n g$ si $d(f,g) < \frac{1}{n}$) car une telle relation n'est en général pas transitive.

- **La structure différentielle au premier ordre** : elle s'obtient par la donnée en chaque sommet d'un espace tangent orienté et donc par la donnée d'un vecteur unitaire de $\mathbb{R}^3$.

Finalement notre modèle n'est autre qu'une courbe polygonale de $\mathbb{R}^3$, dont chaque sommet est pourvu d'un vecteur unitaire.

Notons que la vitesse v_i en un point Γ_i se déduit de l'immersion en choisissant[4] $v_i = \|\Gamma_i\Gamma_{i+1}\|$. Nous aurions pu tout aussi bien enrichir le modèle en adjoignant à chaque sommet un réel positif représentant la vitesse. À l'inverse nous avons spécifié en chaque sommet un espace tangent que l'on aurait pu déduire de l'immersion en choisissant le vecteur $\Gamma_i\Gamma_{i+1}$ normalisé pour vecteur tangent.

Nous avons fait ce choix car, en pratique, les courbes que nous échantillonnerons seront des splines et le calcul du vecteur tangent pour un paramètre donné est simple. Ceci est également vrai pour la vitesse, mais nous avons préféré limiter le nombre d'attributs du modèle pour simplifier les algorithmes. Il est clair cependant que la fidélité dans la reproduction du modèle continu est fonction de la richesse du modèle discret.

En reprenant le formalisme de la section précédente, notre modèle de courbe est ainsi déterminé par les relations et distances suivantes :

- les relations $\sim_n$ sont déterminées par :

  $$\gamma_1 \sim_n \gamma_2 \Leftrightarrow \gamma_1(\tfrac{i}{n}) = \gamma_2(\tfrac{i}{n}) \quad \text{et} \quad \mathbf{t}_1(\tfrac{i}{n}) = \mathbf{t}_2(\tfrac{i}{n})\, \forall\, i = 0, \ldots, n$$

  où $\mathbf{t} = \gamma'/\|\gamma'\|$.
  Un représentant d'une classe peut donc s'identifier à un $(2n+2)$-uplet

  $$\mathcal{R}_n = (\Gamma_0, \ldots, \Gamma_n, \mathbf{t}_0, \ldots, \mathbf{t}_n)$$

- et on définit les distances d_n par :

  $$d_n(\gamma_1, \gamma_2) = \max_i \|\gamma_1(\tfrac{i}{n}) - \gamma_2(\tfrac{i}{n})\| + \max_i \|\mathbf{t}_1(\tfrac{i}{n}) - \mathbf{t}_2(\tfrac{i}{n})\|$$

  De sorte que

  $$\lim_{n\to\infty} d_n(\gamma_1, \gamma_2) = \|\gamma_1 - \gamma_2\|_\infty + \|\mathbf{t}_1 - \mathbf{t}_2\|_\infty$$

  où

  $$\|f\|_\infty = \sup_{s\in[0,1]} \|f(s)\|.$$

4. Pour le dernier sommet on peut prendre la vitesse du sommet précédent.

Ainsi une suite $(\Gamma_i^n, \mathbf{t}_i^n)_{0\leq i\leq n}$ de représentants "converge" vers une courbe γ si et seulement si la suite de fonctions constante par morceau[5] $\Gamma^n : s \mapsto \Gamma^n_{\lfloor sn \rfloor}$ (resp. $\mathbf{t}^n : s \mapsto \mathbf{t}^n_{\lfloor sn \rfloor}$) converge uniformément sur [0,1] vers la fonction γ (resp. $\gamma'/\|\gamma'\|$).

Définition 2.3 *Par la suite nous dirons qu'une suite de séquences $(f_i^n)_{0\leq i\leq n}$* **converge** *vers une fonction f si la suite de fonctions constante par morceau $f^n : s \mapsto f^n_{\lfloor sn \rfloor}$ converge uniformément vers f.*

Nous pouvons désormais définir un algorithme d'interpolation de courbes discrètes qui soit analogue à la méthode L d'interpolation proposée dans le premier chapitre. C'est un algorithme A qui rend le diagramme suivant commutatif :

$$\begin{array}{ccc} \gamma_0, \gamma_1 & \xrightarrow{\sim_n} & (\Gamma^n_{0,i}, \mathbf{t}^n_{0,i})_{n\in\mathbb{N}}, (\Gamma^n_{1,i}, \mathbf{t}^n_{1,i})_{n\in\mathbb{N}} \\ L\downarrow & & \downarrow A \\ \gamma_t & \xleftarrow{lim_\sim} & (\Gamma^n_{t,i}, \mathbf{t}^n_{t,i})_{n\in\mathbb{N}} \end{array}$$

2.2 Algorithme d'interpolation

Nous supposons désormais données deux courbes échantillonnées $(\Gamma^n_{0,i}, \mathbf{t}^n_{0,i})_{0\leq i<n}$ et $(\Gamma^n_{1,i}, \mathbf{t}^n_{1,i})_{0\leq i<n}$ que nous désirons interpoler. Par la suite nous omettrons les indices relatifs au temps (0 , 1 ou t) lorsqu'ils ne seront pas nécessaires.

2.2.1 Schéma principal

En se référant au procédé d'interpolation du premier chapitre, nous devons munir chaque courbe échantillonnée $(\Gamma_i^n, \mathbf{t}_i^n)$ d'un repère adapté (et orthonormé) $\mathcal{R}_i^n$. Ici, *adapté* signifie que $\mathbf{t}_i^n$ est le premier vecteur de R_i^n.

Nous devons ensuite interpoler ce repère, la vitesse $v_i^n = \|\Gamma_i^n\Gamma_{i+1}^n\|$, et l'extrémité Γ_0^n. Cette dernière quantité peut d'ailleurs être indépendante de n, ou plus exactement, $\Gamma^n_{t,0}$ peut ne dépendre que de t.

Nous devons ensuite reconstruire une courbe avec les quantités interpolées. Cependant notre définition des courbes échantillonnées ne permet pas d'écrire :

$$\Gamma_{i+1}^n = \Gamma_i^n + \frac{1}{n} v_i^n \mathbf{t}_i^n$$

puisque dans notre modèle Γ_i^n et $\mathbf{t}_i^n$ sont indépendants.
Afin d'être cohérent avec les données initiales, nous remplaçons cette équation par l'équation de progression suivante :

$$\Gamma_{i+1}^n = \Gamma_i^n + \frac{1}{n} R_i^n \Delta_i^n \tag{2.1}$$

5. $\lfloor x \rfloor$ désigne le plus grand entier inférieur ou égal à x

Δ_i^n est donc n fois la différence $\Gamma_{i+1}^n - \Gamma_i^n$ exprimée dans le repère R_i^n. Si $(\Gamma_i^n, \mathbf{t}_i^n)$ converge vers une courbe γ, il est facile de voir que Δ_i^n converge vers le vecteur

$$\|\gamma'\| \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = v \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}.$$

Par conséquent, en substituant l'interpolation linéaire de Δ_i^n à celle de v_i^n, nous conservons une interpolation linéaire de la vitesse et l'équation 2.1 nous permet de reconstruire la courbe interpolée, à tout instant, de façon compatible avec les données initiales.

Nous montrons en effet, en annexe B, que si $\mathcal{R}_i^n$ et Δ_i^n convergent respectivement vers un repère mobile R et un vecteur Δ, alors la courbe (Γ_i^n) reconstruite à l'aide de l'équation 2.1 converge vers la courbe γ définie par :

$$\gamma(s) = \Gamma_0 + \int_0^s R(u)\Delta(u)du$$

où $\Gamma_0 = \Gamma_0^n$ est indépendant de n. De plus, si Δ est de la forme $v \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$, γ a pour vitesse v et R est un repère mobile adapté pour γ. L'interpolation linéaire de Δ_i^n permet effectivement de remplir cette condition (i.e. $\Delta_{t,i}^n$ converge vers $v_t \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$ avec $v_t = (1-t)v_0 + tv_1$).

Par la suite il nous reste à définir comment construire un repère adapté R_i^n à une courbe échantillonnée $(\Gamma_i^n, \mathbf{t}_i^n)$ et comment interpoler deux repères $R_{0,i}^n$ et $R_{1,i}^n$.

2.2.2 Construction d'un repère adapté à une courbe échantillonnée

Nous devons munir une courbe échantillonnée $(\Gamma_i^n, \mathbf{t}_i^n)$ d'un repère adapté R_i^n (i.e. dont le premier vecteur est $\mathbf{t}_i^n$). Par la suite nous identifierons les repères et les rotations de $\mathbb{R}^3$, utilisant indifféremment un vocabulaire propre aux repères ou aux rotations. Par exemple le repère canonique et l'identité désignerons la même chose.

Par analogie avec le premier chapitre, nous définissons un repère de Frenet discret et un repère du transport relativement parallèle (RTRP) discret.

Le repère de Frenet discret

Le repère de Frenet discret se définit très simplement par le repère[6]

$$R_i^n = (\mathbf{t}_i^n, \mathbf{n}_i^n, \mathbf{b}_i^n)$$

avec

$$\mathbf{b}_i^n = \frac{\mathbf{t}_i^n \wedge \mathbf{t}_{i+1}^n}{\|\mathbf{t}_i^n \wedge \mathbf{t}_{i+1}^n\|} \text{ et } \mathbf{n}_i^n = \mathbf{b}_i^n \wedge \mathbf{t}_{i+1}^n.$$

De même que le repère de Frenet continu ne peut être défini que sur des courbes birégulières, sa version discrète ne peut être définie que si

$$\mathbf{t}_i^n \wedge \mathbf{t}_{i+1}^n \neq 0.$$

Dans ces conditions, il est facile de voir que si $(\Gamma_i^n, \mathbf{t}_i^n)$ converge vers une courbe γ alors R_i^n converge vers le repère de Frenet de γ.

Le RTRP discret

Nous construisons un RTRP discret à l'aide de la progression suivante :

1. nous choisissons un repère R_0^n adapté,
2. puis, par analogie avec la remarque 1.2 du premier chapitre, nous propageons ce repère de sorte que si R_i^n est le repère en Γ_i^n, R_{i+1}^n est adapté et $\|R_{i+1}^n - R_i^n\|$ est minimal pour la norme euclidienne sur les matrices.

Il est facile de voir que R_{i+1}^n est obtenu à partir de R_i^n par la rotation d'angle $\alpha_i^n = (\widehat{\mathbf{t}_i^n, \mathbf{t}_{i+1}^n})$ autour du vecteur $\mathbf{t}_i^n \wedge \mathbf{t}_{i+1}^n$ normalisé (si $\mathbf{t}_i^n \wedge \mathbf{t}_{i+1}^n$ est nul nous prenons l'identité).

Dans la section suivante nous verrons que cette construction, également introduite dans [Klo86], converge effectivement vers le repère du transport parallèle relatif par passage à la limite sur les courbes échantillonnées.

2.2.3 Interpolation des repères (première version)

Nous avons vu dans le premier chapitre que l'interpolation des repères est obtenue par l'interpolation linéaire de leur matrice de Cartan. Or dans $\mathbb{R}^3$ une telle matrice (antisymétrique) s'identifie à un vecteur, et réciproquement, par l'application qui associe à un vecteur $\mathbf{v}$ la matrice $\tilde{\mathbf{v}}$ (relativement à la base canonique) de l'endomorphisme :

$$\mathbf{u} \mapsto \mathbf{v} \wedge \mathbf{u}$$

6. On pose $R_n^n = R_{n-1}^n$

La matrice de Cartan d'un repère mobile (orthonormé) est donc de la forme $\tilde{\mathbf{\Omega}}$ où $\mathbf{\Omega}$ est un vecteur usuellement appelé le vecteur vitesse de rotation instantanée ou VVRI. Par la suite nous exprimerons le problème de l'interpolation en terme de VVRI et non plus de matrice de Cartan. Le choix du repère de Frenet ou du RTRP correspond par exemple à l'annulation de l'une des composantes du VVRI[7].

Dans le cas continu, le repère interpolé R_t est solution du système

$$\begin{cases} R_t'(s) = R_t(s)\,\tilde{\mathbf{\Omega}}_t(s) \\ R_t(0) \text{ donné} \end{cases} \tag{2.2}$$

où

$$\mathbf{\Omega}_t \;=\; (1-t)\,\mathbf{\Omega}_0 \;+\; t\,\mathbf{\Omega}_1.$$

Afin d'obtenir une solution discrète de cette équation nous utilisons une méthode d'Euler (c.f. [Bar77]) appropriée. Nous explicitons et justifions la méthode de résolution employée dans l'annexe C. Celle-ci nous fournit une solution discrète approchée (nous supprimons l'indice t dans ce qui suit) :

$$\begin{cases} R_0^n = R(0) \\ R_{i+1}^n = R_i^n \,\exp(\frac{1}{n}\tilde{\mathbf{\Omega}}(\frac{i}{n})) \end{cases}$$

où exp est l'application exponentielle présentée dans l'annexe C.

Dans la pratique nous utilisons une équation de propagation similaire :

$$R_{i+1}^n = R_i^n \,\exp(\frac{1}{n}\tilde{\mathbf{\Omega}}_i^n) \tag{2.3}$$

où $\mathbf{\Omega}_i^n$ converge vers $\mathbf{\Omega}$ au sens de la définition 2.3. En combinant cette propriété de convergence et celle de la méthode d'Euler (c.f. [Bar77], p 88), il est facile de montrer que R_i^n converge vers la solution du système 2.2.

Pour compléter l'algorithme d'interpolation, il nous faut donc obtenir des approximations du VVRI pour les repères associés aux deux courbes Γ_0 et Γ_1.

VVRI du RTRP

On peut vérifier aisément que le VVRI du RTRP est de la forme $v\kappa\mathbf{b}$ où v est la vitesse, κ la courbure et $\mathbf{b}$ la binormale à la courbe *exprimée dans le repère local*. Notons que ce produit est toujours défini pour une courbe régulière de classe C^2 car là où la (bi)normale n'est pas définie, la courbure est nulle.

Par ailleurs on vérifie assez simplement que :

$$\lim_{u\to 0} \frac{\arccos\langle \mathbf{t}(s), \mathbf{t}(s+u)\rangle}{u} = v(s)\kappa(s)$$

7. En annulant la $3^{\text{ème}}$ composante on retombe sur le repère de Frenet tourné de $\pm\frac{\pi}{2}$.

Ce qui montre que si α_i^n est l'angle entre les tangentes $\mathbf{t}_i^n$ et $\mathbf{t}_{i+1}^n$ (voir figure 2.1), $n\alpha_i^n$ converge vers $v(s)\kappa(s)$. On en déduit que

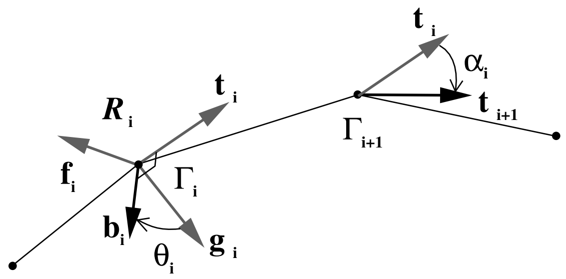


FIG. 2.1 – les angles α_i^n et θ_i^n (les indices n sont supprimés sur la figure).

$$\mathbf{\Omega}_i^n = n\alpha_i^n\, \tilde{\mathbf{b}}_i^n \tag{2.4}$$

fournit une bonne approximation du VVRI. On a par conséquent, d'après l'équation 2.3 :

$$R_{i+1}^n = R_i^n \, \exp(\alpha_i^n\, \tilde{\mathbf{b}}_i^n). \tag{2.5}$$

Compte tenu des propriétés de l'exponentielle, cela signifie que R_{i+1}^n se déduit de R_i^n, comme annoncé dans la section précédente, par rotation autour de la binormale de l'angle α_i^n entre les deux tangentes $\mathbf{t}_i^n$ et $\mathbf{t}_{i+1}^n$. On retrouve ainsi les résultats de [Klo86] sans aucun calcul.

Par ailleurs, le VVRI du RTRP s'exprime dans le repère local sous la forme :

$$\mathbf{\Omega} = v\kappa \begin{pmatrix} 0 \\ -\sin\theta \\ \cos\theta \end{pmatrix}$$

où θ est une primitive de la torsion τ (c.f. [Bis75]). Si on note $(\mathbf{t}_i^n, \mathbf{f}_i^n, \mathbf{g}_i^n)$ les vecteurs du repère R_i^n, une approximation θ_i^n de l'angle θ s'obtient simplement comme on le voit sur la figure 2.1 :

$$\theta_i^n = (\widehat{\mathbf{g}_i^n, \mathbf{b}_i^n})$$

d'où

$$\mathbf{\Omega}_i^n = n\alpha_i^n \begin{pmatrix} 0 \\ -\sin\theta_i^n \\ \cos\theta_i^n \end{pmatrix}.$$

Nous voyons ainsi qu'il est possible d'interpoler linéairement le produit $v\kappa$ (resp. $v\tau$) de la vitesse par la courbure (resp. par la torsion) en interpolant linéairement les angles $\alpha^n_{t,i}$ et $\theta^n_{t,i}$. Nous obtenons ainsi la même interpolation (de l'arc géométrique) qu'avec le repère de Frenet.
En munissant (Γ^n_i) du RTRP discret nous avons donc deux possibilités :

1. ou bien interpoler linéairement le VVRI $\mathbf{\Omega}^n_i$,
2. ou bien interpoler linéairement les angles $\alpha^n_{t,i}$ et $\theta^n_{t,i}$.

Dans les deux cas, la courbe interpolée est toujours munie du RTRP, ce qui peut être un avantage compte tenu de la propriété de minimisation de la rotation du RTRP.

VVRI du repère de Frenet

Le VVRI du repère de Frenet est de la forme $v(-\kappa\mathbf{n}+\tau\mathbf{b})$ où $\mathbf{n}$ est la normale à la courbe et τ sa torsion. Cependant, sa version discrète

$$\mathbf{\Omega}^n_i = -n\alpha^n_i\,\tilde{\mathbf{n}}^n_i + n\beta^n_i\,\tilde{\mathbf{b}}^n_i$$

où $\beta^n_i = (\widehat{\mathbf{b}^n_i, \mathbf{b}^n_{i+1}})$, ne permet pas de retrouver le repère de Frenet comme il a été défini dans la section précédente. Il faut ainsi calculer le VVRI discret indirectement à partir des matrices $R_i^{-1}\,R_{i+1}$ obtenues par construction sur les courbes discrètes initiales. Plus précisément on a

$$\tilde{\mathbf{\Omega}}^n_i = \log((R^n_i)^{-1}\,R^n_{i+1})$$

log étant l'inverse de exp au voisinage de l'identité.

Étant donné le caractère intrinsèque du procédé d'interpolation que l'on cherche à simuler, on peut se demander s'il n'existe pas un calcul plus direct de l'interpolation des repères qui ne fasse pas intervenir le VVRI. Dans la section suivante nous montrons en effet qu'un calcul simple permet de se passer du calcul du VVRI.

2.2.4 Interpolation des repères (seconde version)

La relation 2.3 nous incite à ajouter à l'équation 2.1 de propagation des points une équation de propagation des repères :

$$R^n_{i+1} = R^n_i\,r^n_i \tag{2.6}$$

On obtient ainsi deux suites $(r^n_{0,i})_{0\le i\le n}$ et $(r^n_{1,i})_{0\le i\le n}$ de rotations correspondant aux repères adaptés $(R^n_{0,i})_{0\le i\le n}$ et $(R^n_{1,i})_{0\le i\le n}$. Lorsque n est assez grand, $r^n_{0,i}$ et $r^n_{1,i}$ sont proches de l'identité. On peut par conséquent appliquer l'interpolation locale évoquée dans la section 1.1. Plus précisément, on peut définir $r^n_{t,i}$ comme le point de paramètre t sur la géodésique (de $SO(3,\mathbb{R})$) joignant $r^n_{0,i}$ et $r^n_{1,i}$.

On en déduit progressivement $R^n_{t,i}$ à l'aide de 2.6 et de la rotation $R^n_{t,0} = R_{t,0}$ obtenue à partir d'un chemin qui relie $R_{0,0}$ à $R_{1,0}$. Dans la pratique ce chemin est calculé à l'aide d'une (de la) géodésique les joignant.

En annexe D on montre que ce schéma correspond effectivement à l'interpolation linéaire des VVRI. Cela tient en partie au fait que $n(R^n_i)^{-1}(R^n_{i+1} - R^n_i) = n(r^n_i - Id)$ converge vers $\tilde{\mathbf{\Omega}}$ si R^n_i converge vers une solution de $R' = R\tilde{\mathbf{\Omega}}$.

Calcul avec les quaternions

L'usage des quaternions pour représenter les rotations est très répandu en infographie. Comme il est rappelé dans [BCGH92, HP93], les quaternions présentent l'avantage d'être définis par quatre coordonnées au lieu des neuf coefficients d'une rotation (autrement dit les quaternions sont naturellement plongés dans $\mathbb{R}^4$ tandis que les rotations le sont dans $\mathbb{R}^9$).

Cet usage se justifie par l'existence d'un revêtement simple (à deux feuillets) des rotations par les quaternions unitaires (i.e. de norme égale à 1). Ce revêtement est de plus une homothétie locale de rapport 2 : les géodésiques autour d'un quaternion sont envoyées sur les géodésiques autour de la rotation correspondante mais de longueur double. Notons que le terme géodésique peut se référer ici aussi bien à la structure de groupe de Lie de ces ensembles qu'à leur structure de sous-variété riemanienne usuelle (c.f. également l'annexe C à ce sujet).

Ce revêtement associe au quaternion unitaire $(w,x,y,z) = (w,\mathbf{V})$ la rotation :

$$r = Id + 2\,w\tilde{\mathbf{V}} + 2\,\tilde{\mathbf{V}}^{\mathbf{2}}$$

Inversement, la rotation d'angle α autour du vecteur $\mathbf{b}$ correspond à l'un des deux quaternions

$$\pm\,(\cos(\alpha/2),\ \ \sin(\alpha/2)\,\mathbf{b})$$

Ainsi, en choisissant deux quaternions $q^n_{0,i}$ et $q^n_{1,i}$ correspondant aux deux rotations $r^n_{0,i}$ et $r^n_{1,i}$, on peut effectuer le calcul de $r^n_{t,i}$ en terme de quaternion (on prendra soin de choisir $q^n_{0,i}$ et $q^n_{1,i}$ "proches", c'est-à-dire tels que leur produit scalaire dans $\mathbb{R}^4$ soit positif). On a alors

$$q^n_{t,i} = slerp(q^n_{0,i}, q^n_{1,i}, t)$$

où *slerp* est la procédure d'interpolation des quaternions définie dans [Sho87].

Quaternions et RTRP : le calcul avec les quaternions est particulièrement simple dans le cas du RTRP puisque l'angle et l'axe des rotations r_i^n est connu à l'avance. On a ainsi :

$$q_i^n = \pm\left(\cos(\alpha_i^n/2),\ \ \sin(\alpha_i^n/2)\,\mathbf{b_i^n}\right)$$

où α_i^n et $\mathbf{b_i^n}$ sont définis comme précédemment.

2.3 Extension

On peut reprocher aux modes d'interpolation précédents d'être trop contraints et de ne pas laisser assez de possibilités à l'utilisateur. Afin de donner plus de liberté à l'utilisateur nous proposons de modifier l'interpolation par l'intermédiaire d'une torsion supplémentaire et d'une interaction sur la vitesse locale de transformation. Ces modifications sont obtenues par la spécification de deux champs scalaires le long de chaque courbe à interpoler.

2.3.1 Torsion des repères

Dans notre implémentation, l'utilisateur a la possibilité d'associer interactivement des valeurs réelles en certains points des courbes. Ces valeurs sont ensuite interpolées en chaque point des courbes (au moyen d'une spline) et interprétées comme des angles. Chaque repère est ensuite tourné autour de la tangente en utilisant ces angles. Ainsi le repère R adapté à une courbe est remplacé par

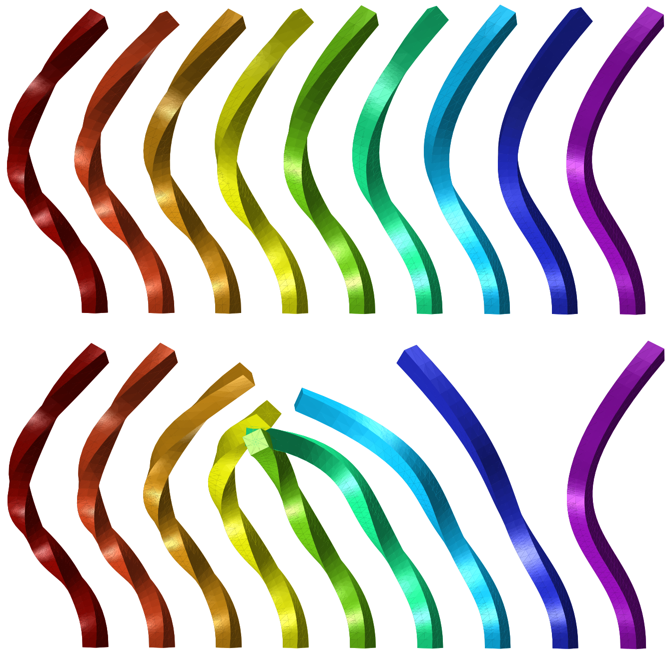
$$\overline{\mathcal{R}} = \mathcal{R}\, r_\varphi$$

où r_φ est la rotation d'angle φ autour de l'axe des x. Le VVRI $\overline{\mathbf{\Omega}}$ de $\overline{\mathcal{R}}$ est relié au VVRI $\mathbf{\Omega}$ de R :

$$\overline{\mathbf{\Omega}} = r_{-\varphi}\,\mathbf{\Omega} - \varphi'\vec{\imath}$$

Pour interpoler deux repères "tordus" $\overline{\mathcal{R}}_0$ et $\overline{\mathcal{R}}_1$ nous pouvons soit interpoler linéairement les quantités $\mathbf{\Omega}$ et φ soit interpoler linéairement $\overline{\mathbf{\Omega}}$. Dans le premier cas la géométrie de la courbe interpolée n'est pas modifiée mais possède une "torsion interne" qui peut être utile si les repères servent par ailleurs de système de coordonnées comme nous le verrons dans la dernière partie de ce manuscrit. La figure 2.2 montre l'effet de la torsion sur une interpolation suivant que l'on interpole séparément ou non $\mathbf{\Omega}$ et φ (la torsion varie de 0 à 4π le long de la courbe initiale). Afin de visualiser la torsion des repères nous avons habillé les courbes d'un cylindre dont les sections carrées sont attachées à chaque repère.

Notons que cette torsion supplémentaire permet de munir une courbe de n'importe quel repère adapté, puisque deux repères adaptés se déduisent l'un de l'autre par rotation autour de la tangente.



La torsion est interpolée séparément.

La torsion est incorporée dans le VVRI.

FIG. 2.2 – Influence d'une torsion supplémentaire.

2.3.2 Contrôle de la vitesse de transformation

Le second champ scalaire est utilisé pour contrôler la vitesse d'interpolation. Plus précisément nous remplaçons le temps t dans les formules d'interpolation linéaire en un point quelconque par une fonction de t et de la valeur du champ scalaire au point considéré. De sorte qu'une quantité x_i sera interpolée par une formule du type :

$$x_{t,i} = (1 - f(t,i))\, x_{0,i} + f(t,i)\, x_{1,i}$$

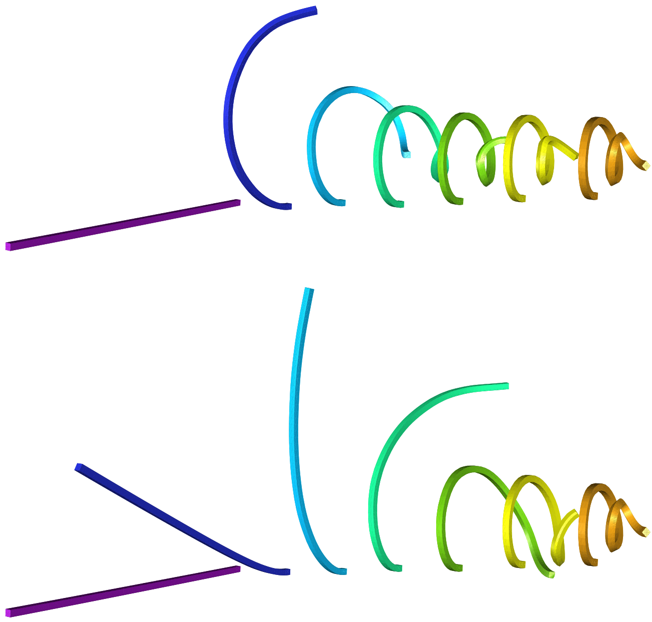
f vérifiant :

$$f(0,i) = 0 \quad \text{et} \quad f(1,i) = 1$$

Un exemple de modification de la vitesse d'interpolation est illustré figure 2.3.

2.3.3 Redressement automatique

Le mode d'interpolation défini dans la première partie permet également de redresser automatiquement une courbe, c'est-à-dire de transformer (à longueur constante) une courbe en une géodésique. Il suffit pour cela de rétracter le repère R_0 sur l'identité. $R_t(s)$ est donc le point de paramètre t sur la géodésique (dans $SO(3,\mathbb{R})$) qui joint $R_0(s)$ à l'identité (c.f.



Interpolation “normale”.

La vitesse d'interpolation augmente graduellement le long de la courbe.

Fig. 2.3 – Influence du contrôle de la vitesse d'interpolation. Les courbes interpolées sont visualisées aux temps t = 0, 0.1, 0.2, 0.35, 0.5, 0.75, 1.

figure 2.4 gauche). Sur le plan algorithmique cela correspond, en se référant à la seconde méthode d'interpolation, à calculer $r^n_{t,i}$ par :

$$r^n_{t,i} = slerp(r^n_{0,i}, Id, t)$$

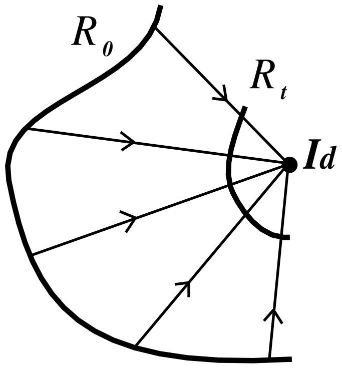
Un exemple de redressement est illustré sur la figure 2.4.

2.4 Implémentation et résultats

Dans notre implémentation nous n'avons utilisé que le RTRP en raison de sa propriété de “minimisation de la torsion”. Nous avons ainsi deux modes d'interpolation , mis à part les paramètres supplémentaires de la section 2.3 :

1. L'un utilise l'interpolation à l'aide des quaternions (c.f. section 2.2.4). Nous appelons Q cette méthode par référence à son mode de calcul.

2. l'autre utilise l'interpolation linéaire des angles α_i et θ_i (c.f. la fin du §VVRI du RTRP, de la section 2.2.3). Nous appelons A cette méthode par référence à son mode de calcul.



Rétraction de R_0 sur l'identité

Les courbes interpolées sont visualisées aux temps t = 0, 0.2, 0.4, 0.6, 0.8, 1.

FIG. 2.4 – Redressement automatique d'une courbe

Nous résumons ci-dessous les étapes de calcul de l'interpolation de deux courbes échantillonnées $(\Gamma_{0,i}, \mathbf{t}_{0,i})$ et $(\Gamma_{1,i}, \mathbf{t}_{1,i})$ pour chacune des méthodes Q et A.

Calcul de l'interpolation par la méthode Q :

initialisation

Pour chaque courbe échantillonnée :

1. calculer les angles $\alpha_i = (\widehat{\mathbf{t_i}, \mathbf{t_{i+1}}})$, les binormales $\mathbf{b_i} = (\mathbf{t_i} \wedge \mathbf{t_{i+1}}) / \parallel \mathbf{t_i} \wedge \mathbf{t_{i+1}} \parallel$ et les quaternions $q_i = (\cos(\alpha_i/2), \sin(\alpha_i/2)\mathbf{b_i})$,
2. choisir un repère $\mathcal{R}_0 = (\mathbf{t_0}, \mathbf{f_0}, \mathbf{g_0})$ et le propager à l'aide de l'équation 2.5,
3. calculer les coordonnées locales $\Delta_i = \mathcal{R}_i^{-1}(\Gamma_{i+1} - \Gamma_i) = \mathcal{R}_i^{\top}(\Gamma_{i+1} - \Gamma_i)$

Puis on obtient la courbe interpolée à chaque instant t :

interpolation à l'instant t

1. interpoler $(\Gamma_{t,0}, \mathcal{R}_{t,0})$ à partir de $(\Gamma_{0,0}, \mathcal{R}_{0,0}), (\Gamma_{1,0}, \mathcal{R}_{1,0})$ et t (on peut utiliser les quaternions),
2. calculer pour chaque i: $q_{t,i} = slerp(q_{0,i}, q_{1,i}, t)$ et $\Delta_{t,i} = lerp(\Delta_{0,i}, \Delta_{1,i}, t)$,
3. reconstruire les $\Gamma_{t,i}$ à l'aide des équations 2.6 (convertie en terme de quaternion) et 2.1

Calcul de l'interpolation par la méthode A :

initialisation

Pour chaque courbe échantillonnée nous reprenons l'initialisation de Q et ajoutons le calcul des angles $\theta_i = (\widehat{\mathbf{g_i}, \mathbf{b_i}})$ dans l'étape 3 :

1. calculer les angles $\alpha_i = (\widehat{\mathbf{t_i}, \mathbf{t_{i+1}}})$, les binormales $\mathbf{b_i} = (\mathbf{t_i} \wedge \mathbf{t_{i+1}}) / \parallel \mathbf{t_i} \wedge \mathbf{t_{i+1}} \parallel$ et les quaternions $q_i = (\cos(\alpha_i/2), \sin(\alpha_i/2)\mathbf{b_i})$,
2. choisir un repère $\mathcal{R}_0 = (\mathbf{t_0}, \mathbf{f_0}, \mathbf{g_0})$ et le propager à l'aide de l'équation 2.5,
3. calculer les coordonnées locales $\Delta_i = \mathcal{R}_i^{-1}(\Gamma_{i+1} - \Gamma_i) = \mathcal{R}_i^{\top}(\Gamma_{i+1} - \Gamma_i)$ et les angles $\theta_i = (\widehat{\mathbf{g_i}, \mathbf{b_i}})$

interpolation à l'instant t

1. reprendre l'étape 1 de la méthode Q,
2. interpoler linéairement les quantités $\alpha_{t,i}, \theta_{t,i}$ et $\Delta_{t,i}$,
3. reconstruire les $\Gamma_{t,i}$ à l'aide de ces données en utilisant les équations 2.4, 2.3 et 2.1.

Nous présentons quelques interpolations de courbes sur les figures 2.5 et 2.6. Les courbes sont obtenues par échantillonnage de splines cubiques et nous avons habillés les courbes d'un cylindre à base carrée afin de visualiser la torsion des repères. Les mêmes courbes initiales et finales sont utilisées dans les figures 2.5 et 2.6. La figure 2.5 montre une interpolation avec la méthode A, vue de face et de dessus. La figure 2.6 montre une interpolation avec la méthode Q vu de face pour deux niveaux d'échantillonnage différents. La courbe sous-jacente à la figure du dessus comporte 40 points de discrétisation, tandis que celle sous-jacente à la figure du dessous en comporte 10. La comparaison de ces deux images montre la très bonne stabilité de notre algorithme.

Vue de face.

Vue de Dessus.

FIG. 2.5 – Interpolation par la méthode A.

2.5 Conclusion

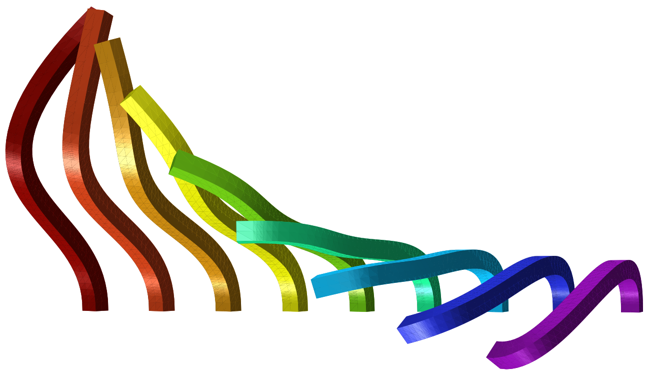
Nous avons présenté une nouvelle technique d'interpolation de courbes polygonales tridimensionnelles. Pour cela nous nous sommes appuyés sur un raisonnement dans le "continu" (c.f. chapitre 1) et nous avons établi avec précision une *analogie* avec un modèle combinatoire. Cette analogie utilise une notion originale de *modèle* décrite dans la section 2.1.

Les propriétés du procédé continu font que notre méthode est particulièrement bien adaptée pour le contrôle local (des courbures) et global (des longueurs) de la géométrie des courbes interpolées.

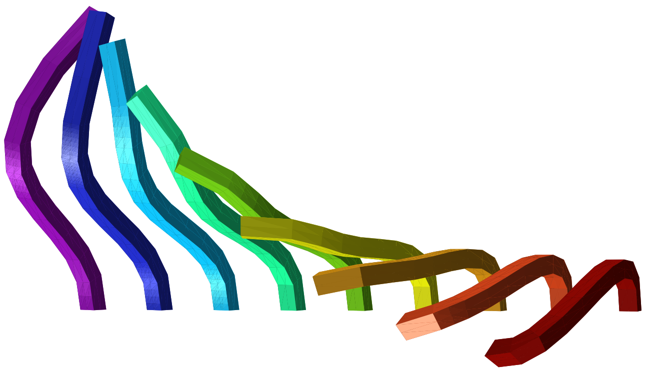
Par ailleurs, notre souci de définir dans le cas continu des procédés les plus intrinsèques possibles nous a permis d'obtenir des algorithmes les plus simples possibles et nécessitant le moins de paramètres possibles.

Afin d'obtenir une plus grande variété de transformations, nous offrons à l'utilisateur la possibilité de contrôler la torsion et la vitesse de transformation des courbes.

Notons enfin que l'extrême simplicité des algorithmes résultants permet d'obtenir des animations en temps interactif.



Vue de face. Les courbes ont 40 points.



Les courbes ont 10 points.

FIG. 2.6 – Interpolation par la méthode Q.

Deuxième partie

Aide à la création d'axe

Comme nous l'avons précisé dans l'introduction générale, les métamorphoses que nous présentons dans la dernière partie utilisent un axe défini dans chaque objet à métamorphoser. Nous avons montré comment animer ces axes, une des étapes nécessaires de la métamorphose. Dans cette partie nous étudions un moyen pour construire automatiquement ces axes dans les objets, c'est à dire un moyen d'associer une courbe squelette à un objet de forme allongée.

Chaque axe correspondant à un objet doit permettre de le paramétrer. Les contraintes permettant de définir un axe dépendent donc du mode de paramétrisation choisi. Dans notre cas les objets sont paramétrés de la manière suivante : une courbe γ permet de paramétrer un objet si celui-ci peut se décomposer en trois parties, une partie cylindrique fermée par deux calottes hémisphériques. Les calottes doivent respectivement être étoilées par rapport aux deux extrémités de la courbe et la partie cylindrique doit admettre une paramétrisation de la forme :

$$(s,\theta) \;\mapsto\; \gamma(s) + r(s,\theta)\,(\cos\theta\,\mathbf{e_2}(s) + \sin\theta\,\mathbf{e_3}(s))$$

où $(\mathbf{e_2}, \mathbf{e_3})$ est un repère normal à γ et r une fonction positive.

L'existence d'une telle décomposition n'est assurée que pour un certain type d'objet que nous appelons **objet étoilé par rapport à l'axe** γ.

La difficulté pour montrer qu'un objet est étoilé par rapport à un axe et pour construire automatiquement un tel axe semble indiquer que cette hypothèse d'étoilement est très restrictive. Néanmoins elle permet de définir des métamorphoses très simplement comme nous le verrons dans la dernière partie, et constitue de plus une extension par rapport à l'approche de [KPC91] qui suppose l'objet étoilé par rapport à un point. C'est pourquoi nous avons conservé ce type d'objet.

Dans la pratique nous n'avons pas réussi à formaliser un algorithme qui réponde à ces deux questions :

1. dire si un objet peut être étoilé par rapport à un axe,

2. dans l'affirmative, proposer un tel axe.

Par contre nous proposerons un moyen de fournir une courbe pour un objet quelconque qui pourra servir de point de départ à un utilisateur. Ceci explique le titre de cette partie qui se restreint à une *aide* à la création d'axe.

Par ailleurs nous avons jugé l'approche que nous avons développée suffisamment intéressante en elle-même pour la présenter ici.

Cette partie s'organise de la manière suivante :

– nous présentons d'abord la notion d'*axe médian* abondamment utilisé en analyse d'image et en reconnaissance de formes et nous montrons en quoi cet axe médian n'est pas adapté à notre besoin,

- puis nous exposons, dans un esprit analogue au premier chapitre de la partie I, les principes qui guident notre approche,
- nous développons ensuite un algorithme de décomposition d'un objet polyédrique en bandelettes cylindriques à partir duquel nous obtenons un axe,
- nous concluons sur des voies possibles de recherche pour obtenir un axe qui réponde à la condition d'étoilement.

Chapitre 1

Squelettisation et axe médian

La représentation des objets à partir de squelettes a largement été utilisée en analyse d'image pour la reconnaissance de forme. Un squelette permet en effet de réduire l'information géométrique ou topologique d'un objet et il est plus simple de tester la ressemblance des squelettes d'objets que celle des objets eux-mêmes.

Dans une certaine mesure il est même possible de parler de codage puisqu'à partir d'un squelette et d'une fonction positive définie sur ce squelette on peut reconstruire une forme. Des algorithmes ont étés proposés pour des formes simples, en particulier en CAO [GD95].

Comme il est décrit dans l'état de l'art introductif de [MR95] il existe plusieurs façons de définir le squelette d'un objet, également appelé *axe médian*. Voici les principales définitions que l'on trouve dans la littérature :

1. c'est le centre des boules maximales incluses dans l'objet,

2. c'est le lieu des points de rencontre des fronts d'une onde sphérique se propageant à vitesse constante à partir de la frontière de l'objet (analogie avec le feu de prairie),

3. c'est l'ensemble des points crêtes (ridge points) pour l'application qui à un point intérieur à l'objet associe sa distance au complémentaire de l'objet.

Ces définitions sont valides en toutes dimensions et fournissent des ensembles squelettes sensiblement équivalents (des précautions doivent cependant être prises concernant la régularité de l'objet et de sa frontière). Chaque définition donne lieu à des algorithmes de construction spécifiques. Ainsi la première mène à la construction du diagramme de Voronoï pour des ensembles de points qui approximent la surface de l'objet (c.f. [Sch89]). La seconde conduit à des opérations d'amincissement sur des images discrètes, tandis que la dernière induit le recourt à des méthodes de recherches de courbes extrémales dans un champ de potentiel à l'aide de modèles de courbes déformables (snakes) (c.f. [LL89]).

L'étude des propriétés du squelette est assez délicate. Elle a été en partie menée à l'aide des outils de la *morphologie mathématique* ([Ser82, Mat88]).

Comme l'étude précise et préalable de ce que l'on cherche à construire est de première importance nous décrirons dans ce qui suit les grandes lignes de l'approche proposée en morphologie mathématique.

1.1 Axe médian et morphologie mathématique

Nous résumons ici, à partir de [Ser82, Pre], les notions topologiques de bases utilisées en morphologie mathématique et exposons brièvement les principaux résultats de [Mat88].

Ces résultats portent sur la connexité du squelette et la semi-continuité de l'application qui associe son squelette à un objet. Le terme de semi-continuité n'a de sens que si nous définissons les ensembles source et but de l'application squelette et une topologie sur ces ensembles.

Les objets que nous étudions sont des sous-ensembles d'un espace topologique E (dans la pratique $\mathbb{R}^n$). Il est pratique pour des études théoriques de considérer des classes d'objets particulières liées à la topologie de E. On peut ainsi s'intéresser aux sous-ensembles fermés, ouverts, ou encore compacts. Chacune de ces classes d'objets peut être munie d'une topologie en relation avec celle de E :

– si l'on considère l'ensemble noté $\mathcal{F}$ des fermés, en supposant que E est localement compact (i.e. tout point admet un voisinage compact) et séparable (i.e. E admet une base dénombrable d'ouverts), on peut définir une topologie sur $\mathcal{F}$ appelée *topologie en tout ou rien* que nous explicitons en appendice,

– si l'on considère l'ensemble noté $\mathcal{G}$ des ouverts, on peut se placer dans les conditions précédentes et remarquer que l'application "complémentaire", qui passe au complémentaire dans E, réalise une bijection de $\mathcal{G}$ sur $\mathcal{F}$. La topologie sur $\mathcal{G}$ est alors l'image réciproque de celle de $\mathcal{F}$ par cette application ou encore la topologie la moins fine sur $\mathcal{G}$ qui la rende continue. Cette remarque simple pourra intéresser le lecteur de [Ser82] ou [Mat88] où la topologie sur $\mathcal{G}$ est abondamment utilisée sans jamais être définie (sic).

– si l'on considère l'ensemble des compacts non vides, il est commode de supposer que E est un espace métrique. Dans ce cas on peut définir une distance sur l'ensemble des compacts, la distance de Hausdorff. Dans [Sch89] on utilise en fait une autre topologie (sur *tous* les compacts, vides ou non), la *topologie myope*, dont la définition est proche de celle sur $\mathcal{F}$ (mais plus fine que celle induite par $\mathcal{F}$).

Pour ne pas alourdir le texte nous donnons en appendice E des précisions sur les notions de convergence, limite et semi-continuité pour la topologie en tout ou rien.

Deux résultats sur le squelette Lorsque $E = \mathbb{R}^n$ on définit le squelette d'un objet comme l'adhérence des centres des boules maximales incluses dans l'objet (une boule est maximale si elle n'est strictement incluse dans aucune autre boule elle-même incluse dans l'objet).

Dans [Mat88] on étudie la continuité de l'application squelette. Suivant qu'on définisse cette application sur $\mathcal{F}$ ou $\mathcal{G}$ ou que l'on considère des boules ouvertes ou fermées on obtient des résultats différents. Ainsi l'application squelette définie sur $\mathcal{F}$ à partir de boules fermées n'est continue ni supérieurement ni inférieurement comme le montre l'exemple de la figure 1.1 extrait de [Mat88]. Par contre l'application squelette définie sur $\mathcal{G}$ à partir de boules

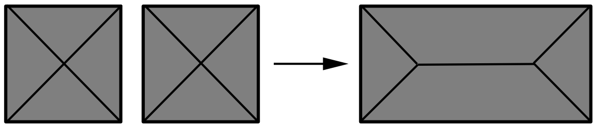


FIG. 1.1 – La réunion d'un carré C unitaire et de son translaté $C_{1+1/n}$, le long d'une de ces arêtes, converge dans $\mathcal{F}$ vers un rectangle lorsque la translation de longueur $1 + 1/n$ tend vers 1. Cependant la limite supérieure du squelette de ce $C \cap C_{1+1/n}$ n'est pas contenue dans le squelette du rectangle, et la limite inférieure ne contient pas le squelette du rectangle.

ouvertes est semi-continue inférieurement (de $\mathcal{G}$ dans $\mathcal{F}$). La démonstration utilise la caractérisation (3) (de l'appendice E) de la semi-continuité inférieure et le fait que pour toute suite convergente $G_i \mapsto G$ dans $\mathcal{G}$ et toute boule maximale B dans G, il existe une suite de boules maximales dans les G_i qui converge vers B (dans $\mathcal{G}$).

Le second résultat porte sur la connexité du squelette et stipule que si G est un ouvert connexe ne contenant aucun demi-espace (i.e. de boule de rayon infini) alors le squelette est connexe.

L'étude de la continuité de l'application squelette permet de mieux prévoir le comportement des algorithmes correspondants et en particulier leur stabilité par rapport aux données numériques. Cependant le résultat précédent de semi-continuité paraît assez difficile à interpréter d'autant plus que les espaces $\mathcal{F}$ et $\mathcal{G}$ ne sont pas métrisables.

1.1.1 Algorithmes de construction de l'axe médian

Il existe deux grandes familles d'algorithmes de construction du squelette d'un objet. L'une utilise une formulation à partir des opérations de base de la morphologie mathématique

que sont l'érosion et la dilatation (c.f. [Ser82]), l'autre relie le squelette à la triangulation de Delaunay d'une discrétisation de la frontière de l'objet.

La première formulation se prête bien à des calculs dans des espaces discrets $\mathbb{Z}^2$ (resp. $\mathbb{Z}^3$). L'idée est d'amincir l'objet progressivement pixel par pixel (resp. voxel par voxel) jusqu'à atteindre le squelette de l'objet (voir par exemple [HP84]).

La seconde utilise le fait que les sphères de Delaunay[1] associées à une triangulation ne contiennent aucun sommet de l'ensemble triangulé en leur intérieur. Quand l'échantillonnage de la frontière est très fin cela signifie intuitivement que les sphères de Delaunay sont tangentes à la frontière de l'objet en plusieurs points et sont donc maximales.

Ainsi M. Schmitt montre dans [Sch89] que pour tout objet compact dont la frontière est une réunion d'hypersurfaces de classe $\mathcal{C}^3$: *l'ensemble des centres des sphères de Delaunay associées à toute suite d'échantillonnage ϵ-dense de la frontière de l'objet*[2] *converge au sens de la topologie en tout ou rien (et au sens de la topologie myope) vers le squelette de l'objet quand ϵ tend vers 0.* Dans le cas d'objets plans à frontière assez régulière le squelette est un graphe planaire localement fini. M. Schmitt relie la topologie du diagramme de Voronoï d'une discrétisation de la frontière avec celle du squelette de l'objet. Dans le cas tridimensionnel nous n'avons pas de résultat mais il semble que pour des objets suffisamment réguliers le squelette soit une union finie de variétés à bord de dimension 1 ou 2.

Il existe néanmoins de nombreux algorithmes pour calculer le squelette à partir du diagramme de Voronoï aussi bien en 2D [MR95, Cho95] (dans ces articles on détermine le diagramme de Voronoï de polygones avec des côtés rectilignes ou non, et pas simplement de leurs sommets) qu'en 3D [GYKD91].

1.2 Conclusion

Si l'axe médian se prête bien à des traduction algorithmiques, il ne correspond cependant pas à ce que nous recherchons, à savoir une courbe (c.f figure 1.2). On peut alors penser à extraire une courbe intéressante à partir de l'axe médian. La première idée qui vient à l'esprit est de prendre le squelette du squelette. Cependant la tâche est ici un peu plus ardue si l'on veut utiliser un algorithme de type Delaunay, car les surfaces à trianguler (correspondant aux différentes branches du squelette) peuvent être gauches. Pour bien faire il faudrait utiliser une métrique induite sur chaque morceau du squelette. En fait, en supposant que les courbures des morceaux sont grandes par rapport à leur degré d'échantillonnage, on peut utiliser une heuristique comme dans [ST92] : partant de 3 sommets, on regarde si la plus petite sphère

1. Pour un état de l'art sur les diagrammes de Voronoï et les triangulations de Delaunay voir [Aur90]

2. Un échantillonnage est dit ϵ-dense si toute boule de rayon ϵ centrée sur la frontière de l'objet contient au moins un échantillon

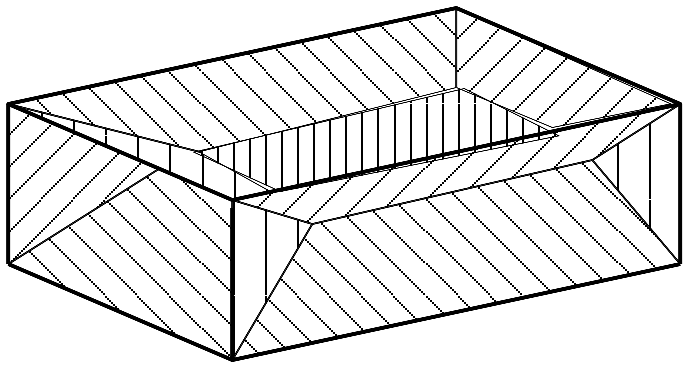


FIG. 1.2 – Squelette d'un parallélépipède.

englobante contient d'autres sommets et dans la négative on poursuit la triangulation en conservant le triangle formé par ces trois sommets.

Une telle 'déviation' de la notion d'axe médian paraît assez arbitraire et ne permet pas de justifier son emploi. C'est pourquoi nous n'avons pas poussé plus avant dans cette direction.

Notons que d'autres définitions de squelettes ont été proposées et étudiées en 2D ([Ros86, Pon88]) mais elles s'appliquent à des objets tellement particuliers (enveloppe de famille à un paramètre de cercles, balayage d'une courbe par un segment...) qu'elles sont inutilisables dans la pratique.

Dans le chapitre suivant nous exposons une autre méthode pour associer une structure filaire à un objet. Il s'agira d'un diagramme, parfois appelé diagramme de Reeb, issu de la théorie de Morse [Gra71] sur les surfaces. Cette notion est d'ailleurs évoquée dans la plupart des articles de L. Kunii et Y. Shinagawa tels que [SKK91, SK91, KST92, IKSU92]...Chaque article est accompagné de l'exemple donné figure 1.3 du diagramme de Reeb de la fonction hauteur sur un tore. Nous présenterons un diagramme similaire fondé sur une autre application que la hauteur. Pour arriver à cette notion nous développons d'abord quelques idées d'ordre général d'une manière analogue à la première partie.

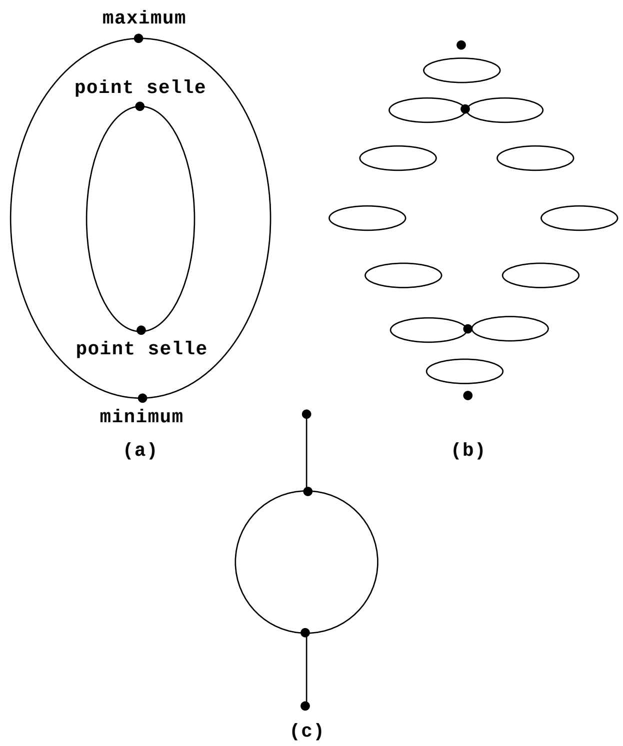


FIG. 1.3 – (a) Un tore, (b) ses sections et (c) son diagramme de Reeb.