



HAL
open science

Calcul du rayon de courbure d'une séquence d'ADN

Franck Rapaport

► **To cite this version:**

Franck Rapaport. Calcul du rayon de courbure d'une séquence d'ADN. [Stage] A04-R-276 || rapaport04a, 2004, 27 p. inria-00099893

HAL Id: inria-00099893

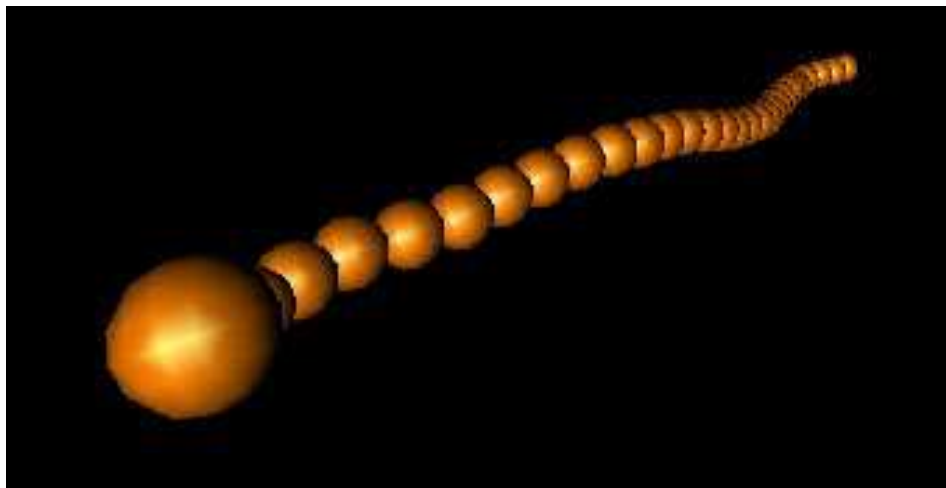
<https://inria.hal.science/inria-00099893>

Submitted on 26 Sep 2006

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

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

Calcul du rayon de courbure d'une séquence d'ADN



Jeudi 24 Juin 2004

stagiaire : Franck RAPAPORT

encadrant : Isabelle DEBLED-RENNESON

jury : Noëlle CARBONELL
Didier GALMICHE
Jean-Yves MARION
Dominique MERY
Jeanine SOUQUIERES

Université HENRI POINCARÉ
Ecole Doctorale IAEM-LORRAINE / DEA Informatique de Lorraine
LORIA / Equipe ADAGE

Table des matières

1	Le problème biologique	1
1.1	Structure de l'ADN	1
1.2	Les travaux des biologistes	4
1.2.1	CURVATURE	4
1.2.2	D'autres modèles de calcul	4
1.2.3	Pertinence de notre approche	5
2	Introduction à la géométrie discrète	6
2.1	Géométrie discrète	6
2.2	Droites discrètes 2D	6
2.3	Code de Freeman	8
2.4	Droites discrètes 3D	9
2.5	Calcul de la courbure en un point d'une courbe discrète 2D	11
3	Segments flous et courbure d'ordre d	13
3.1	Segments flous 2D	13
3.2	Segments flous 3D	14
3.3	Courbure d'ordre d	15
3.4	Torsion discrète	16
4	Expérimentations	18
4.1	Présentation du programme	18
4.2	Courbure de courbes connues	20
4.3	Courbure de séquences d'ADN	23
5	Conclusion et perspectives	24
	Bibliographie	25

Chapitre 1

Le problème biologique

1.1 Structure de l'ADN

La conformation spatiale de la molécule d'acide désoxyribonucléique (ou ADN) est un facteur primordial pour la génétique. En effet, la forme de la macromolécule peut nous aider à déterminer les sites de certaines réactions chimiques (par exemple l'accrochage de protéines spécifiques) en nous permettant d'extraire les endroits facilitant ces réactions. Dès lors, il devient plus facile d'identifier le rôle de ces sites.

Du point de vue chimique, l'ADN est une longue molécule formée par un enchaînement de blocs appelés nucléotides et composés d'un sucre (la désoxyribose), d'un groupement phosphate et d'une base nitrogenée parmi quatre bases possibles (l'adénine, la thymine, la guanine ou la cytosine). Les groupements phosphate et sucre de deux nucléotides successifs se lient, formant un long polymère. Depuis 1953 [14], on sait que les brins sont complémentaires, c'est-à-dire qu'à chaque base sur un brin correspond une base analogue sur l'autre (l'adénine correspond à la thymine, la guanine à la cytosine), comme le montre l'image ci-après.

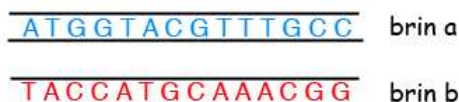


FIG. 1.1 – Une séquence d'ADN

La forme de l'ADN est celle d'une échelle tordue en double hélice, les liaisons hydrogène entre les paires de nucléotides analogues formant les barreaux de l'échelle, comme le montre l'image 1.2, propriété de l' *Image Library of Biological Macromolecules* [15]. Les deux brins étant complémentaires, il suffit déjà de connaître la séquence de bases d'un seul d'entre eux pour connaître la composition chimique de la molécule. Dès lors, on peut représenter le code génétique par une suite de lettres caractérisant les nucléotides, par exemple leurs initiales. C'est sous cette forme qu'on le stockera.

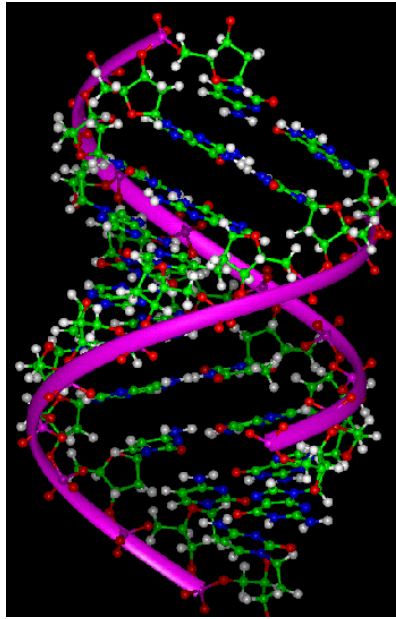


FIG. 1.2 – Une section d'ADN du virus *bacteriophage lambda*

Penser que la forme de l'ADN est identique quelle que soit la séquence est un peu naïf. En réalité, la double hélice est soumise à un *polymorphisme structurel* ainsi qu'à des *déformations locales*. Le polymorphisme structurel implique que selon le milieu dans lequel elle évolue, la même séquence d'ADN existe sous trois formes, A, B et Z qui diffèrent par l'espacement des nucléotides et la forme générale de la double hélice. La forme B étant de loin la plus répandue, nous ne nous intéresserons qu'à celle-ci.

Les déformations locales sont de petites variations de la structure de l'ADN mais sans que celle-ci soit modifiée en profondeur. Or, représenter ces déformations dans des segments d'ADN est difficile car elles sont trop vastes pour être visibles à l'échelle atomique mais trop petites pour qu'on puisse les observer sur des modélisations faites à l'aide de modèles élastiques, bons descripteurs de comportements macroscopiques.

Nous allons donc adopter un autre modèle appelé *modèle tubulaire* (ou *rod model* [17] en anglais). Ces modèles ne s'intéressent qu'à la forme générale de l'ADN, c'est-à-dire à la courbe formée par les milieux des paires de nucléotides, comme le montre l'image 1.4. Il suffit alors de suivre l'axe du modèle tubulaire pour obtenir les déformations locales de la structure et donc la forme de la molécule.

Cependant, cette représentation bien épurée ne permet pas encore une analyse automatique de la structure. En effet, pour qu'un ordinateur puisse se consacrer à cette tâche, il faut que l'on en ait extrait des données chiffrées à partir desquelles une machine pourra travailler. L'objectif de notre travail a donc été de trouver un estimateur mathématique adapté à l'analyse de cette structure. Dans la prochaine section, nous verrons les estimateurs adoptés par des équipes de biologistes au cours de différents travaux.

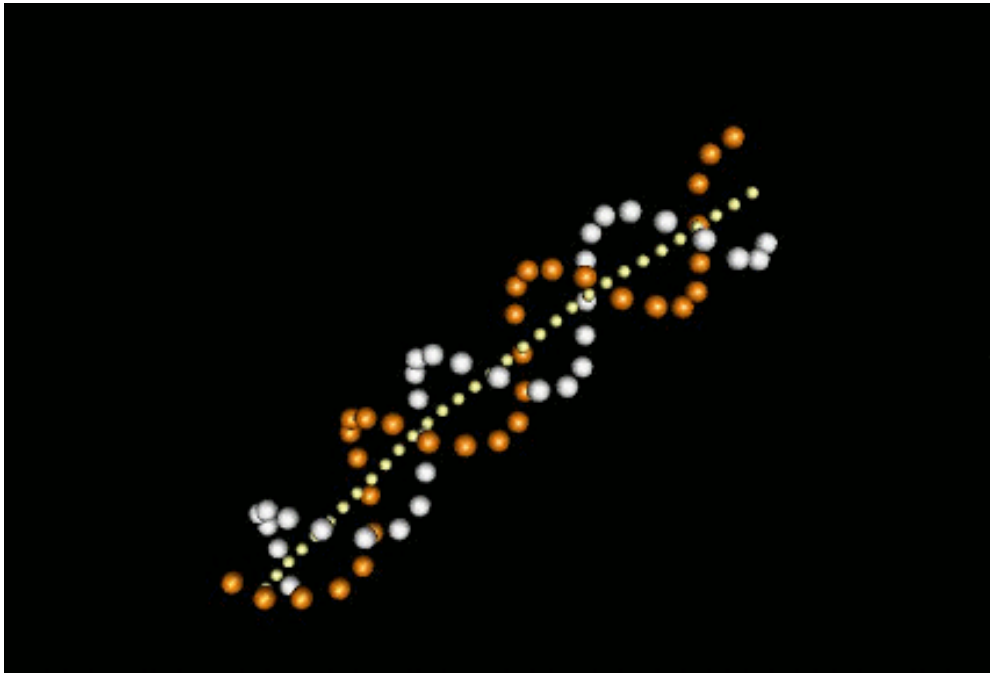


FIG. 1.3 – La modélisation de l'ADN : les points extérieurs sont les emplacements des nucléotides de chaque brin et les points intérieurs les points du modèle tubulaire (modélisation obtenue à l'aide du logiciel *CURVATURE*)

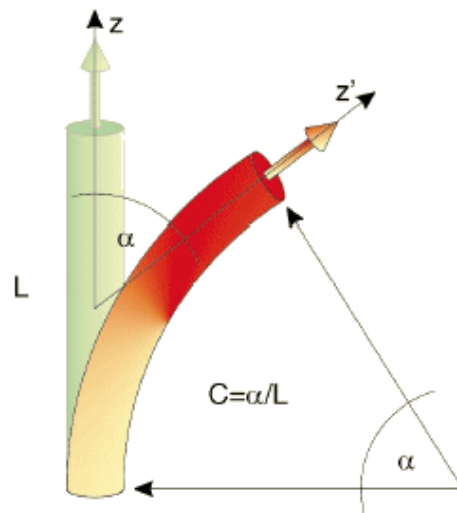


FIG. 1.4 – Schématisation du modèle tubulaire (C est la courbure du tube)

1.2 Les travaux des biologistes

1.2.1 CURVATURE

Pour calculer la structure d'un modèle tubulaire, Alexander Bolshoy [1] [20] [12] a divisé la taille du tubes en petits segments cylindrique dont le nombre de paire de bases est constant (en l'occurrence 2, on dit que son modèle est *dinucléotidique*). Il y a 16 combinaisons de deux lettres possibles, ou 10 si l'on admet qu'il y a symétrie des bases complémentaires (c'est-à-dire que la déformation induite par un segment AA est analogue à celle d'un segment TT). On peut alors diviser toute séquence en modules dinucléotidiques et, puisqu'on connaît la forme de chaque module, en déduire la forme de la structure associée à la séquence.

Alexander Bolshoy a déterminé la forme de chaque segment à l'aide d'expériences biologiques et en a déduit un modèle dinucléotidique [1]. Il a ensuite écrit un programme du nom de *CURVATURE* qui, à partir d'une séquence entrée en paramètre, rend les coordonnées de chacune des extrémités de ces segments dinucléotidiques et donc la forme du modèle tubulaire [20].

Ce programme peut aussi approximer le rayon de courbure du modèle tubulaire en chaque point I_n . Pour cela, on lui donne un entier a qui représentera un paramètre de fenêtre. Il prend les coordonnées du point I_{n-a} situé a bases en amont du point I_n et les coordonnées du point I_{n+a} situé a bases en aval du point I_n . Il calcule le rayon r du cercle passant par ces trois points et considère que r est le rayon de courbure au point I_n .

Comme on peut l'imaginer, cette méthode de calcul n'est pas une très bonne approximation du rayon de courbure réel du modèle tubulaire. Ainsi, selon la valeur du paramètre de fenêtre choisi, les résultats seront très différents. De plus, l'auteur prend des paramètres de l'ordre de cent. Il va donc négliger les petites variations de la structure. Il a ainsi réussi à montrer la forte courbure de l'ADN dans certaines séquences promotrices, particulièrement celles de la bactérie E. Coli [12].

Le modèle tubulaire fourni par *CURVATURE* est un modèle réaliste qui a été utilisé par de nombreux biologistes pour différents travaux [11] et que nous réutiliserons nous-même par la suite.

1.2.2 D'autres modèles de calcul

Le modèle fourni par *CURVATURE* n'est pas le seul modèle tubulaire existant. En effet, selon les expériences, les biologistes n'ont pas tous obtenu la même forme pour les segments dinucléotidiques. Ainsi, Cacchione a obtenu des résultats différents de ceux de Bolshoy dans le calcul des structures [2]. Il a aussi établi un modèle du rayon de courbure, mais ses calculs n'étant pas basés sur des théories mathématiques mais uniquement sur des calculs probabilistes, nous ne les étendrons pas ici.

Richard E. Dickerson a écrit un logiciel du nom de *FREEHELIX* [10], qui remplace son ancien logiciel *NEWHELIX*, dédié à l'analyse de portions d'ADN très courbées. Seulement, au lieu de s'intéresser à l'évolution du rayon de courbure, Dickerson préfère se pencher sur l'évolution des différents angles de la molécule par rapport à son axe principal. Il appelle ces trois angles *roll*, *slide* et *twist* et ils permettent de caractériser entièrement la molécule. Cependant, le principal désavantage de cette approche est qu'elle oblige à considérer trois estimateurs à la fois et que le modèle ne peut s'appliquer qu'aux molécules fortement courbées.

Munteanu et al. [17] soulignent aussi que si les modèles actuels ne sont pas exacts,

on pourrait encore obtenir de meilleurs résultats en augmentant la taille des segments considérés . Ainsi, ils considèrent qu'un modèle trinuécléotidique (chaque segment de la taille de trois paires de bases) devrait permettre d'obtenir des résultats très proches de ceux obtenus sur la molécule réelle, mais un tel modèle n'existe pas encore.

Masotti et al. [16] ont récemment proposé une méthode pour retrouver la courbure d'un brin d'ADN à partir d'un échantillon physique de la molécule (avec une déviation de 1 pour cent), ce qui prouve que le problème intéresse encore et toujours les biologistes.

Différents visualisateurs de séquences d'ADN ont été créés, on citera entre autres ADN-Viewer [13], qui présente la particularité de laisser le choix entre les modèles de Cacchione et Bolshoy. De la même manière, DIAMOD [11] de Dlakic et Harrington propose un visualisateur qui s'appuie sur différentes modélisations de la molécule. Les auteurs proposent même une comparaison de ces différents modèles.

1.2.3 Pertinence de notre approche

Il existe donc différents types de méthodes biologiques pour estimer la forme de l'ADN :

- Celles qui, telles la méthode de Bolshoy, nécessitent un paramètre et donnent des résultats très approximatifs.
- Des méthodes exactes comme celle de Dickerson, mais dont les résultats sont si complets qu'ils sont difficiles à analyser. En effet, cette méthode ne fournit pas un mais trois estimateurs complémentaires et qu'il faut considérer les trois à la fois pour pouvoir interpréter la courbe.
- Des méthodes probabilistes, comme celle de Cacchione qui ne permettent pas de déterminer avec certitude la courbure de la molécule.

Cependant, aucune de ces méthodes n'est à la fois rigoureuse mathématiquement et facile à exploiter. Le rayon de courbure semble approprié pour une bonne estimation qualitative de la forme de la molécule, cependant, la nature des modèles tubulaires, qui sont représentés par une série de points déconnectés, empêche le calcul par des méthodes usuelles qui nécessiteraient la connaissance de l'équation complète de la courbe. Nous allons donc utiliser des méthodes liées à la géométrie discrète, à la fois rigoureuses mathématiquement et pourtant appropriées pour l'étude de ce type de courbe puisque comme nous allons le voir, la géométrie discrète se consacre à l'étude d'ensemble de points. Cependant, pour pouvoir mieux comprendre ces méthodes, nous allons d'abord étudier quelques bases de la géométrie discrète avant de regarder comment calculer le rayon de courbure.

Chapitre 2

Introduction à la géométrie discrète

2.1 Géométrie discrète

Par définition, la géométrie discrète est l'étude de l'espace discret et de ses applications. Mathématiquement, cela signifie qu'au lieu de se placer dans \mathbb{R}^n comme en géométrie usuelle, on s'intéressera aux phénomènes de \mathbb{Z}^n . Une façon concrète de comprendre ce nouvel espace est de se le représenter comme une grille dont chaque point (que l'on appellera *voxel* dans l'espace tri-dimensionnel et *pixel* dans l'espace bi-dimensionnel) devient une case.

Le choix de cet espace n'est pas anodin : il est en effet mieux adapté aux raisonnements informatiques puisque, du fait de sa structure physique, un ordinateur ne peut travailler qu'avec des nombres finis, et donc dans un espace discret. Puisqu'on utilise des méthodes mathématiques basées sur un espace analogue, notre raisonnement sera plus facile à transcrire en algorithmes informatiques. Cependant, cette limitation impose de redéfinir toutes les notions couramment admises.

En effet, le problème principal posé par la transposition dans un espace discret est la disparition de toute notion de continuité, notion intimement liée à la densité de l'espace de travail. Il est donc nécessaire de trouver de nouveaux moyens de caractériser les figures géométriques les plus élémentaires, et donc principalement les droites.

2.2 Droites discrètes 2D

La notion de droite discrète arithmétique a été définie par JP Réveillès en 1991 [19].

Définition 1 Dans \mathbb{Z}^2 , une droite de pente $\frac{a}{b}$ (avec $b \neq 0$ et $\text{pgcd}(a,b) = 1$), de borne inférieure μ et d'épaisseur arithmétique ω est l'ensemble des points (x,y) de \mathbb{Z}^2 vérifiant la double inéquation :

$$\mu \leq ax - by < \mu + \omega \quad (2.1)$$

On notera une telle droite $D(a,b,\mu,\omega)$. Une droite discrète est donc définie par le couple d'inéquations $\mu \leq ax - by$ et $ax - by < \mu + \omega$, c'est l'ensemble des points

entiers compris entre les droites réelles d'équation $ax - by = \mu$ et $ax - by = \mu + \omega - 1$, appelées respectivement *droite d'appui supérieure* et *droite d'appui inférieure* [6]. Les points entiers de ces droites sont appelés *points d'appui*, respectivement supérieurs et inférieurs, de la droite discrète.

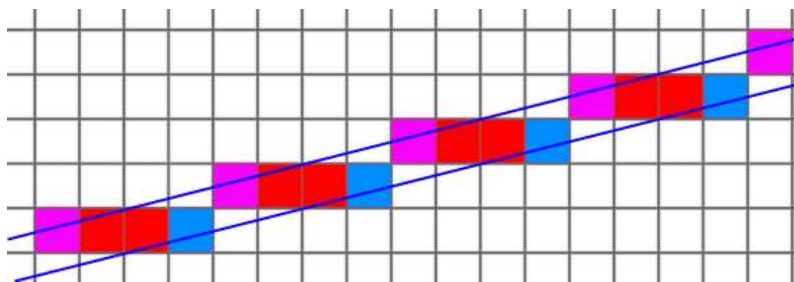


FIG. 2.1 – Une droite discrète d'équation $0 \leq x - 4y < 4$ accompagnée de ses droites et points d'appui (supérieurs en rose et inférieurs en bleu)

On s'intéressera à deux types de droites discrètes : les droites discrètes *épaisses* et les droites dites *naïves*, vérifiant $\omega = \sup(|a|, |b|)$. On notera une telle droite $D(a, b, \mu)$.

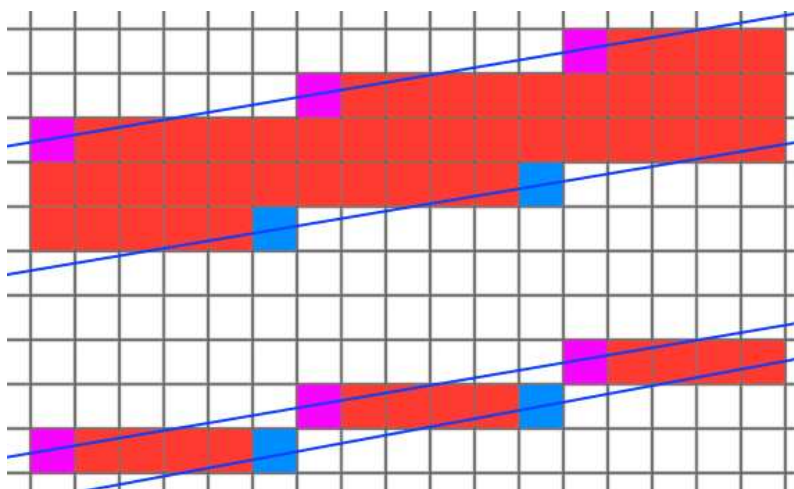


FIG. 2.2 – La droite épaisse $D(1, 6, -42, 17)$ et la droite naïve $D(1, 6, 0)$, accompagnées de leurs droites et points d'appui

Ce sont les droites d'appui qui sont la base d'un algorithme incrémental de reconnaissance de segments de droites naïves, c'est-à-dire un algorithme qui, à partir d'un segment Σ de la droite naïve D et des coordonnées d'un point M connecté à une extrémité de Σ , reconnaît si $\Sigma \cup M$ est un segment de droite naïve. Il s'appuie sur le théorème suivant :

Théorème 1 *Considérons $r(M)$ le reste au point $M(x, y)$ en fonction de D défini par :*

$$r(M) = ax - by \quad (2.2)$$

1. *Si $\mu \leq r(M) < \mu + b$, alors $M \in D(a, b, \mu)$
 $\Sigma \cup M$ est un segment reconnu de la droite D .*
2. *Si $r(M) = \mu - 1$, alors M est faiblement extérieur à D
 $\Sigma \cup M$ est un segment reconnu de la droite de pente donnée par le vecteur S_0M avec S_0 premier point d'appui supérieur de D .*
3. *Si $r(M) = \mu + b$, alors M est faiblement extérieur à D
 $\Sigma \cup M$ est un segment reconnu de la droite de pente donnée par le vecteur I_0M avec I_0 premier point d'appui inférieur de D .*
4. *Si $r(M) < \mu - 1$ ou $r(M) > \mu + b$ alors M est fortement extérieur à D
 $\Sigma \cup M$ n'est pas un segment de droite discrète*

Cet algorithme rapide, incrémental et linéaire est expliqué dans un article de Jean-Pierre Réveillès et Isabelle Debled-Renneson [9], mais nous ne le détaillerons pas ici car il nécessiterait l'introduction d'autres notions plus complexes.

Définition 2 *On appellera vecteur de décalage positif (respectivement vecteur de décalage négatif) de $D(a, b, \mu, \omega)$ noté V_{+1} (respectivement V_{-1}), un vecteur qui relie deux points de D dont les restes sont différents de $+1$ (respectivement -1) et tel que $0 \leq x_{V_{+1}} \leq b$ (respectivement $0 \leq x_{V_{-1}} \leq b$).*

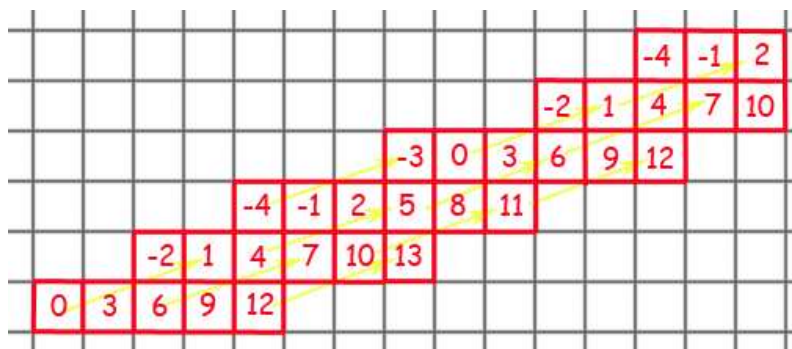


FIG. 2.3 – Représentation du vecteur de décalage positif de la droite épaisse $D(3, 8, -4, 18)$. Les valeurs des restes sont indiqués sur chaque pixel du segment.

2.3 Code de Freeman

Pour représenter de manière plus simple une séquence de points discrets 8-connexes de \mathbb{Z}^2 (par exemple une droite discrète naïve), H. Freeman a proposé d'associer à chaque direction principale un entier compris entre 0 et 7 d'après le schéma suivant :

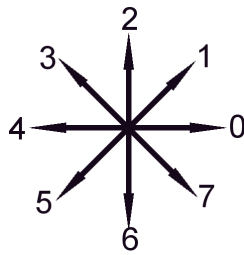


FIG. 2.4 – Les valeurs du code de Freeman

On peut alors diviser l'espace en huit parties, appelées octants. Le premier octant sera délimité par les directions 0 et 1, le deuxième par les directions 1 et 2, etc.

Définition 3 On appelle code de Freeman d'une courbe, le point d'origine P_0 et la suite (α_i) avec i variant de 0 à n , où α_i est la direction selon laquelle P_{i+1} est obtenu à partir de P_i . Ce code, qui représente de façon unique et exacte les courbes 8-connexes, a un avantage certain : il permet en effet d'effectuer facilement des changements d'octants de ces droites, puisqu'une rotation est alors équivalente à une addition dans le code. Dès lors, il devient très facile de transposer tous les calculs dans le premier octant, et donc de limiter la complexité des algorithmes en ne considérant qu'un seul cas.

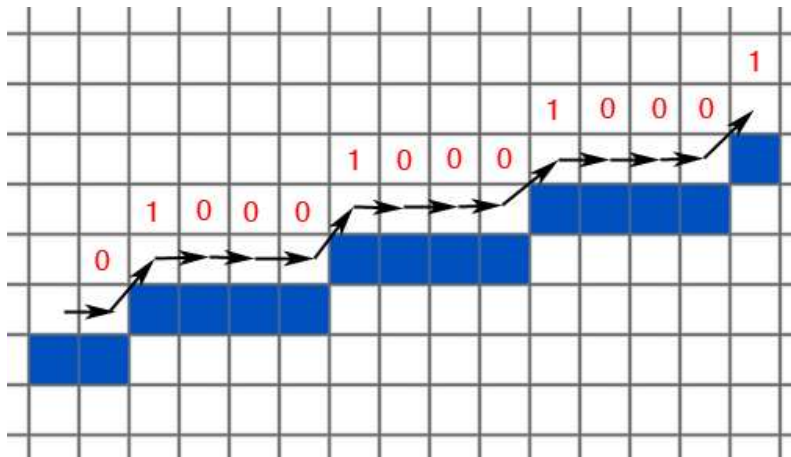


FIG. 2.5 – Une droite discrète et son code de Freeman

2.4 Droites discrètes 3D

Définition 4 Une droite discrète 3D est une courbe tri-dimensionnelle dont les projections dans deux plans orthogonaux sont des droites discrètes 2D. La droite discrète 3D, D , de vecteur directeur $V(a,b,c)$ avec $(a,b,c) \in \mathbb{Z}^3$ et $a \geq b \geq c$ est définie par l'ensemble des points vérifiant les inéquations :

$$D \begin{cases} \mu \leq cx - az < \mu + e \\ \mu' \leq bx - ay < \mu' + e' \end{cases} \quad (2.3)$$

avec $\mu, \mu', e, e' \in \mathbb{Z}$. μ et μ' sont les bornes inférieures de D et e et e' ses épaisseurs arithmétiques.

Les droites qui ne vérifient pas $a \geq b \geq c$ peuvent être définies de la même façon en permutant a, b et c dans la définition.

On appellera droite naïve 3D une droite discrète qui vérifie $e = e' = a$. On notera une telle droite $D(a, b, c, \mu, \mu')$.

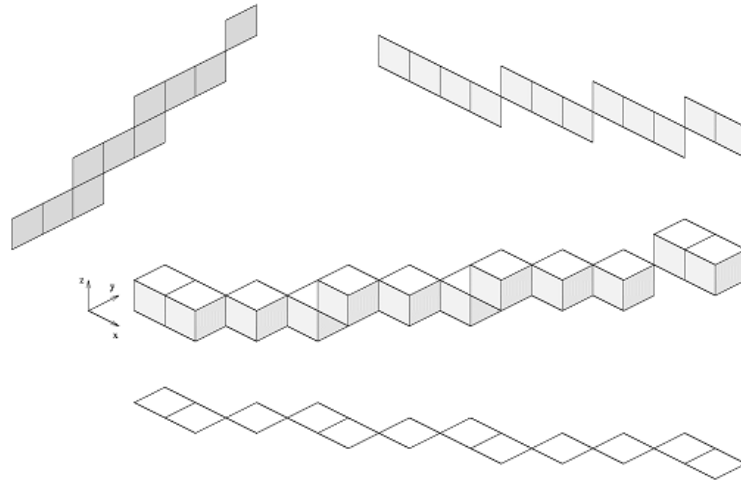


FIG. 2.6 – Un segment de la droite discrète tridimensionnelle $D(10, 7, 3, 0, 0)$

Il existe des algorithmes simples pour construire la droite discrète tridimensionnelle joignant deux points donnés de l'espace discret. Dans [6] et [4], un algorithme linéaire pour construire une droite discrète 3D naïve entre deux points $A(x, y, z)$ et $B(x', y', z')$ tels que

$$\begin{cases} (x - x') \geq (y - y') \geq (z - z') \\ x \geq x' \\ y \geq y' \\ z \geq z' \end{cases} \quad \text{a été présenté. Nous avons programmé}$$

cet algorithme, et l'avons étendu à tout l'espace en utilisant à chaque fois les rotations et les symétries nécessaires pour se retrouver dans les conditions sus-nommées.

Il est possible de montrer qu'un ensemble de points D est une droite discrète 3D si et seulement si il existe au moins deux plans orthogonaux dans lesquels les projections de D sont des droites discrètes 2D. On aura donc juste à appliquer l'algorithme incrémental de reconnaissance de droite (mentionné dans la section précédente) en parallèle sur chacune des deux projections dans les plans principaux pour vérifier si l'ensemble est une droite 3D. On remarquera donc que l'algorithme de reconnaissance des droites 3D est un algorithme linéaire et incrémental.

2.5 Calcul de la courbure en un point d'une courbe discrète 2D

En géométrie classique, la courbure en un point s'obtient à partir de l'équation de la courbe. En géométrie discrète, nous n'avons que les coordonnées des points, et la formule usuelle, basée sur des calculs différentiels n'est plus utilisable. Le plus simple est donc de revenir à la définition première de la courbure.

Pour une courbe 8-connexe, ce cercle peut être approximé en recherchant les plus longs segments appartenant à la courbe à gauche et à droite du point considéré (ces plus longs segments sont alors appelés *demi-tangentes*). Le cercle circonscrit au triangle formé par les extrémités de ces segments constitue alors une bonne approximation du cercle osculateur, comme David Coeurjolly l'a démontré dans sa thèse [3].

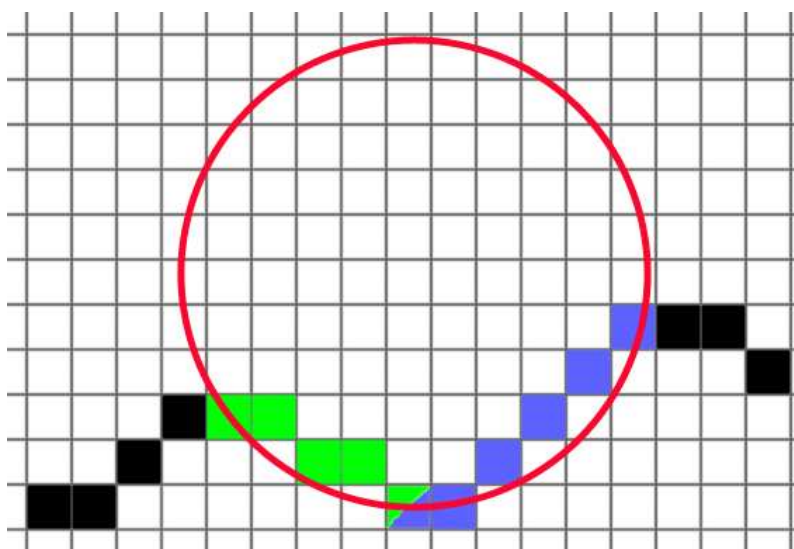


FIG. 2.7 – Approximation d'un cercle osculateur

A partir des extrémités des segments de droite obtenus, on peut calculer une bonne approximation de la courbure :

Théorème 2 Soit P le point considéré, P_g l'extrémité de la demi-tangente à gauche et P_d l'extrémité de la demi-tangente à droite. On note s_1 la distance $\|\overrightarrow{PP_d}\|$, s_2 la distance $\|\overrightarrow{PP_g}\|$ et s_3 la distance $\|\overrightarrow{P_gP_d}\|$. On a alors :

$$R_c = \frac{s_1 * s_2 * s_3}{\sqrt{(s_1 + s_2 + s_3)(s_1 - s_2 + s_3)(s_1 + s_2 - s_3)(s_2 + s_3 - s_1)}} \quad (2.4)$$

La courbure Γ est telle que :

$$\Gamma = \frac{1}{R_c} \quad (2.5)$$

Des méthodes de calcul de la courbure discrète ont aussi été mises en place pour les courbes tridimensionnelles par David Coeurjolly [3] [5].

En revanche, pour une courbe non connexe, ce qui est souvent le cas si on travaille sur des relevés expérimentaux, on ne peut pas calculer le plus long segment de droite à gauche ou à droite. Par conséquent, nous avons contourné le problème en rendant la droite connexe en reliant chaque couple de points successifs par un segment de droite discrète. Cependant, en faisant ainsi, on modifie les informations. En effet, dans le plan discret, ce n'est pas parce que trois points A, B et C sont alignés que l'union des segments [AB] et [BC] est un segment. Aussi, nous allons travailler avec des segments flous qui tolèrent du bruit dans les données.

Chapitre 3

Segments flous et courbure d'ordre d

3.1 Segments flous 2D

Définition 5 Les segments flous sont une extension des segments de droite discrète 2D. On dit qu'un ensemble S_f (avec $\text{Card}(S_f) \geq 2$) de points 2D consécutifs et 8-connexes est un segment flou d'ordre d si et seulement si il existe une droite discrète $D(a, b, \mu, \omega)$ contenant tous les points de S_f telle que $d \leq \frac{\omega}{\max(|a|, |b|)}$. On appellera une telle droite droite englobante du segment flou. [8]

On remarquera qu'un segment flou d'ordre 1 est une droite discrète naïve.

Soit S_f un segment flou d'ordre d qui couvre l'intervalle $[x_G, x_D]$ de l'axe des abscisses. $\bar{D}(a, b, \mu, \omega)$, une droite englobante de S_f , est dite strictement englobante en haut (respectivement en bas) si :

- D contient au moins trois points d'appui dans l'intervalle $[x_G, x_D]$.
- S_f contient au moins un point d'appui supérieur (respectivement inférieur) de D .

D sera dite strictement englobante si D est strictement englobante en haut et en bas.

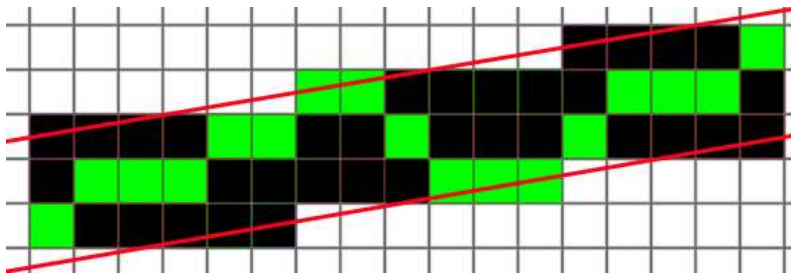


FIG. 3.1 – Un segment flou de droite englobante $D(1, 6, -42, 17)$

De la même façon que pour les droites discrètes, il existe un théorème qui permet de déterminer, à partir d'un segment flou S_f et d'un point connecté à S_f , M , si $S_f \cup M$ est un segment flou [8] :

Théorème 3 Soit un segment flou S_f qui couvre l'intervalle $[0, l - 1]$ sur l'axe des x et $D(a, b, \mu, \omega)$ une droite strictement englobante de S_f . Dans ce cas, l'ordre de S_f est $\frac{\omega}{b}$. Soit M un point entier connecté à S_f et d'abscisse égale à l ou $l - 1$.

1. Si $\mu \leq r(M) < \mu + \omega$, alors $M \in D$.

$S_f \cup M$ est un segment flou d'ordre $\frac{\omega}{b}$ qui a D pour droite strictement englobante.

2. Si $r(M) \leq \mu - 1$ alors M est extérieur à D .

$S_f \cup M$ est un segment flou d'ordre $\frac{\omega'}{b}$ qui a $D'(a', b', \mu', \omega')$ pour droite strictement englobante avec

- b' et a' coordonnées du vecteur $\overrightarrow{P_{r(M)+1}M}$, $P_{r(M)+1}$ étant le point de D de reste $r(M) + 1$ et $x_{P_{r(M)+1}} \in [0, b - 1]$.

- $\mu' = a'x_M - b'y_M$

- $\omega' = a'x_{L_L} - b'y_{L_L} - \mu' + 1$ avec $L_L(x_{L_L}, y_{L_L})$ dernier point d'appui inférieur de la droite D présent dans S_f

De plus, on peut aussi calculer les vecteurs de déplacement de D' :

$$\overrightarrow{V'_{-1}} = (b' - b, a' - a) + [\frac{b}{b'}](b', a')$$

$$\overrightarrow{V'_{+1}} = (b, a) - [\frac{b}{b'}](b', a')$$

3. Si $r(M) \geq \mu + \omega$, alors M est extérieur à D .

$S_f \cup M$ est un segment flou d'ordre $\frac{\omega'}{b'}$ qui a $D'(a', b', \mu', \omega')$ pour droite strictement englobante avec

- b' et a' coordonnées du vecteur $\overrightarrow{P_{r(M)-1}M}$, $P_{r(M)-1}$ étant le point de D de reste $r(M) - 1$ et $x_{P_{r(M)-1}} \in [0, b - 1]$.

- $\mu' = a'x_{U_L} - b'y_{U_L}$ avec $U_L(x_{U_L}, y_{U_L})$ dernier point d'appui supérieur de la droite D présent dans S_f

- $\omega' = a'x_M - b'y_M - \mu' + 1$

De plus, on peut aussi calculer les vecteurs de déplacement de D' :

$$\overrightarrow{V'_{-1}} = (b, a) - [\frac{b}{b'}](b', a')$$

$$\overrightarrow{V'_{+1}} = (b' - b, a' - a) + [\frac{b}{b'}](b', a')$$

Là aussi, on peut se baser sur ce théorème pour déterminer un algorithme incrémental et linéaire de reconnaissance de segment flou d'ordre d [8]. En effet, on observe qu'à chaque ajout de point, les caractéristiques du segment flou peuvent être entièrement recalculées. Il suffit donc d'exclure les points qui augmente l'ordre du segment flou au-dessus de l'ordre désiré. Pour simplifier (et donc accélérer) encore l'algorithme, on peut même prendre directement pour les valeurs de U_L et L_L les derniers points d'appui de D sans même vérifier s'ils sont dans S_f , mais la droite obtenue ne sera englobante qu'en haut ou en bas. Nous avons programmé cet algorithme.

3.2 Segments flous 3D

De la même façon que la notion de droite discrète 2D a été étendue vers les droites discrètes 3D, nous avons introduit la notion de segment flou 3D :

Définition 6 Un segment flou 3D d'ordre d est une séquence de points 26-connexe telle qu'il existe deux plans orthogonaux dans lesquels la projection de la séquence de points est un segment flou 2D d'ordre d .

Nous avons aussi introduit un algorithme de reconnaissance des segments flous 3D. De la même façon que pour les droites discrètes 3D, cet algorithme repose sur l'algorithme de reconnaissance des segments flous 2D : il suffit de regarder si chacune des projections dans les plans principaux est un segment flu 2D. En vérifiant les deux projections simultanément, on a là aussi un algorithme incrémental et linéaire, exactement l'analogue de celui existant pour les droites discrètes. Nous avons programmé cet algorithme en nous appuyant sur le code de l'algorithme de reconnaissance de segment flu 2D.

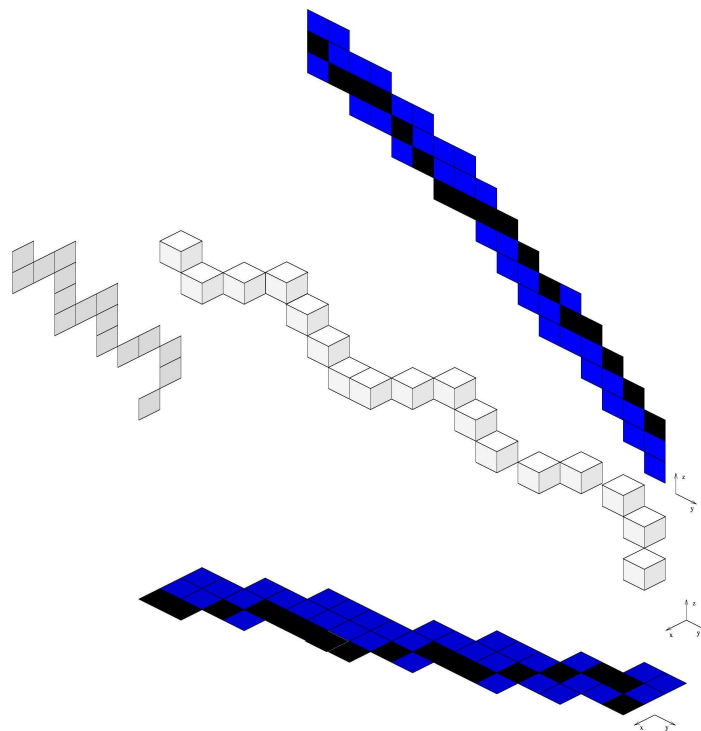


FIG. 3.2 – Un segment flu 3d d'ordre 3 accompagné de ses droites englobantes

3.3 Courbure d'ordre d

Dans le cas de relevés physiques, la courbe obtenue est souvent bruitée. Dès lors, la technique de calcul de la courbure discrète usuelle ne fonctionne plus, elle nous donne des résultats qui ne sont pas cohérents avec le domaine physique car l'algorithme de détection de droite est sensible à la moindre perturbation. Il suffit donc de remplacer l'algorithme de reconnaissance de droite par un algorithme de reconnaissance de segment flu pour ne plus être gêné par ces perturbations. On obtient alors une bonne approximation de la courbure du phénomène, l'ordre d'approximation étant donné par l'ordre des segments flous.

Nous avons introduit la notion de courbure d'ordre d en 3 dimensions :

Définition 7 La courbure d'ordre d en un point d'une courbe 2d-connexe est obtenue en approximant le cercle osculateur par le cercle circonscrit au triangle formé par les plus longs segments flous d'ordre d (qu'on appellera alors demi-tangentes d'ordre d) à gauche et à droite du point considéré.

De la même façon que pour la courbure usuelle, on obtient alors la valeur de Γ_d , courbure d'ordre d , à l'aide de la formule suivante :

Théorème 4 Soit P le point considéré, P_{d_G} l'extrémité de la demi-tangente d'ordre d à gauche et P_{d_D} l'extrémité de la demi-tangente d'ordre d à droite. On note s_1 la distance $\|\overrightarrow{PP_{d_D}}\|$, s_2 la distance $\|\overrightarrow{PP_{d_G}}\|$ et s_3 la distance $\|\overrightarrow{P_{d_G}P_{d_D}}\|$. On a alors :

$$R_{c_d} = \frac{s_1 * s_2 * s_3}{\sqrt{(s_1 + s_2 + s_3)(s_1 - s_2 + s_3)(s_1 + s_2 - s_3)(s_2 + s_3 - s_1)}} \quad (3.1)$$

La courbure d'ordre d Γ_d est telle que :

$$\Gamma_d = \frac{1}{R_{c_d}} \quad (3.2)$$

Cette nouvelle notion de courbure d'ordre d réduit la précision des calculs mais présente trois avantages :

- Comme précisé précédemment, elle est plus adaptée à l'analyse de relevés physiques, puisque les segments flous sont peu sensibles au bruit
- Elle permet de travailler avec des algorithmes de complexité réduite puisque, comme l'algorithme de reconnaissance de droites "exactes", l'algorithme de reconnaissance de segments flous est incrémental et linéaire. Notre algorithme de calcul de courbure en un point sera donc là aussi linéaire.
- Cette nouvelle courbure permet de travailler sur des courbes non connexes. En effet, on peut alors rendre les courbes connexes en joignant les points successifs à l'aide de droites discrètes 3D. Les perturbations induites alors en remplaçant les morceaux de courbes manquants par des droites ne sont plus limitants.

Elle avait déjà été introduite dans l'espace bi-dimensionnel [7], mais jamais en dimension 3.

3.4 Torsion discrète

Si la courbure est un bon estimateur de la forme d'une courbe bidimensionnelle, elle ne suffit plus en trois dimensions. La meilleure façon de voir ce problème est de considérer une spirale. Le rayon de courbure R_c (et donc la courbure $\Gamma = \frac{1}{R_c}$) d'une spirale est constant et égal au rayon du disque associé à la spirale. La courbure ne permet donc pas de différencier une spirale d'un disque.

Pour pallier à ce problème, nous avons introduit un nouvel estimateur des courbes discrètes : la *torsion discrète*. La torsion usuelle est un estimateur utilisé en géométrie différentielle (la géométrie qui s'intéresse aux courbes de \mathbb{R}^n avec $n \geq 2$). On définit la torsion de façon suivante (d'après l'Atlas des Mathématiques [18]) :

Définition 8 Soit $\overrightarrow{b(s)}$ le vecteur défini par $\overrightarrow{b(s)} = \overrightarrow{t(s)} \wedge \overrightarrow{n(s)}$ où $\overrightarrow{t(s)}$ est le vecteur tangent à la courbe au point d'abscisse curviligne s et $\overrightarrow{n(s)}$ le vecteur normal principal à la courbe au point d'abscisse curviligne s . On appelle $\overrightarrow{b(s)}$ vecteur binormal à la courbe au point d'abscisse curviligne s .

On définit alors la torsion au point $M(s)$ par

$$\tau(s) = \lim_{h \rightarrow 0} \frac{\sphericalangle(\overrightarrow{b(s)}, \overrightarrow{b(s+h)})}{h} \quad (3.3)$$

où \sphericalangle est la fonction angle sphérique.

De manière générale, on retiendra que la torsion est l'angle entre les vecteurs normaux aux plans osculateurs à la courbe en deux points infiniment proches.

En géométrie discrète, on peut approximer le plan osculateur en un point d'une courbe par le plan qui contient les deux demi-tangentes à droite et à gauche de ce point. Dès lors, on pourra considérer que la direction du vecteur binormal est donnée par le produit vectoriel des vecteurs directeurs de ces deux segments. On peut directement parler de torsion d'ordre d pour peu que les demi-tangentes soient des demi-tangentes d'ordre d . Nous avons donc défini la torsion discrète d'ordre d de la façon suivante :

Définition 9 Soit P_n le point considéré, P_{n+1} son successeur dans la courbe, P_{n_dG} l'extrémité de la demi-tangente d'ordre d à gauche et P_{n_dD} l'extrémité de la demi-tangente d'ordre d à droite. On appelle $\overrightarrow{g_{n_d}}$ et $\overrightarrow{d_{n_d}}$ les vecteurs définis respectivement par :

$$\overrightarrow{g_{n_d}} = \frac{\overrightarrow{P_n P_{n_dG}}}{P_n P_{n_dG}} \quad \overrightarrow{d_{n_d}} = \frac{\overrightarrow{P_n P_{n_dD}}}{P_n P_{n_dD}} \quad (3.4)$$

On définit alors le vecteur binormal d'ordre d en P_n par :

$$\overrightarrow{b_{n_d}} = \overrightarrow{g_{n_d}} \wedge \overrightarrow{d_{n_d}} \quad (3.5)$$

Dès lors, la torsion d'ordre d au point d'indice n est donnée par :

$$\tau_d(n) = \sphericalangle(\overrightarrow{b_{n_d}}, \overrightarrow{b_{(n+1)_d}}) \quad (3.6)$$

C'est-à-dire :

$$\tau_d(n) = \arccos\left(\frac{\overrightarrow{b_{n_d}} \cdot \overrightarrow{b_{(n+1)_d}}}{\|\overrightarrow{b_{n_d}}\| \|\overrightarrow{b_{(n+1)_d}}\|}\right) \quad (3.7)$$

Nous avons donc ainsi défini un nouvel estimateur pour une courbe discrète 3D. Cet estimateur, dont l'analogie est couramment utilisé en géométrie différentielle depuis le 19^{ème} siècle (au début sous le nom de seconde courbure), est un bon complément de la courbure et permet d'analyser des courbes tri-dimensionnelles là où la courbure n'était appropriée que pour les courbes bi-dimensionnelles.

Chapitre 4

Expérimentations

4.1 Présentation du programme

Nous avons développé un programme en C++ qui permet, à partir d'un fichier contenant les coordonnées des points non forcément connectés d'une courbe 3D, de calculer la courbure de la courbe en chacun de ces points à l'aide des méthodes précédemment expliquées dans ce document. Pour cela, à partir des points réels, il obtiendra une courbe discrète en multipliant ces points par un entier qu'on appellera *grain* puis en prenant la partie entière du nouveau nombre entier obtenu. Il suffit alors de relier les nouveaux points entiers par des droites discrètes naïves pour obtenir une courbe discrète connexe.

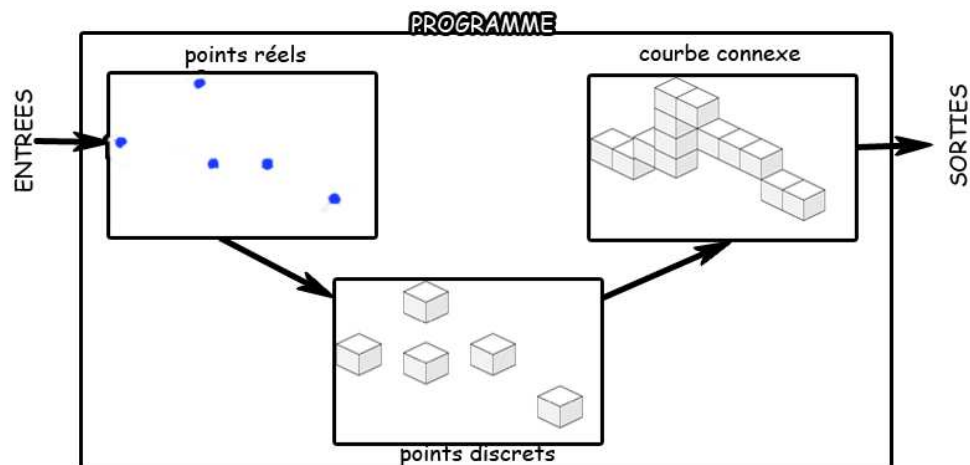


FIG. 4.1 – Les transformations que le programme effectue sur les points réels

Les entrées sont les suivantes :

- Un fichier texte qui contient les coordonnées des points de la courbe considérée. Ce fichier est formaté comme les fichiers en sortie du logiciel CURVATURE :

chaque ligne représente un point, et sont données dans l'ordre les coordonnées X, Y et Z de ce point.

- Un entier représentant ce qu'on appelle le *grain* ou la *résolution*. C'est l'inverse de la taille d'une unité. Par exemple, si la résolution est 10, le point de coordonnées (5,432 ; 6 ; 7,86) deviendra le point de coordonnées (54 ; 60 ; 79) pour le programme. Cette notion a un sens dans la mesure où les points obtenus en sortie de CURVATURE ne sont de manière générale, pas entiers. Les coordonnées d'un point seront donc multipliées par ce grain et approximées par l'entier le plus proche avant d'être traitées par le programme.
- Un entier qui représente l'ordre des segments flous à considérer.

En échange, le programme rend les données suivantes :

- Un fichier texte qui contient sur chaque ligne l'indice du point, sa courbure, le nombre de points considérés pour chaque demi-tangente ainsi que la torsion
- Deux fichiers lisibles avec gnuplot ¹ qui représentent les graphes de courbure et de torsion en fonction de l'indice du nucléotide. Ces fichiers ont en abscisse l'indice du nucléotide (qui correspond à une abscisse curviligne) et en ordonnée respectivement la courbure et la torsion.
- Eventuellement un fichier VRML ² qui est une représentation en 3d (lisible avec des programmes gratuits et répandus sous Linux et avec Internet Explorer sous Windows) de la courbe rentrée en entrée. On peut éventuellement lui demander de tracer les droites entre les points si la courbe est non connexe.

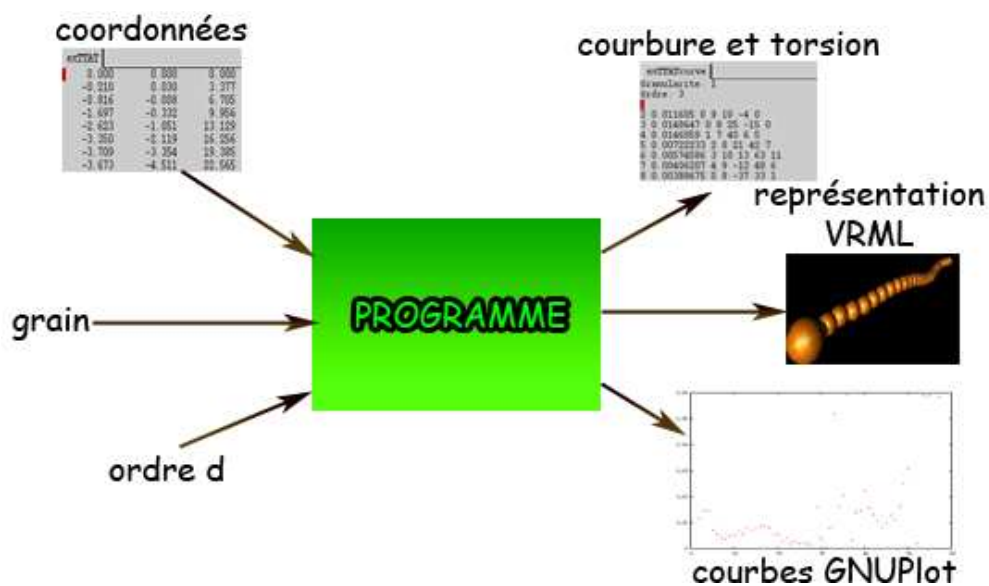


FIG. 4.2 – Les entrées et sorties du programme

¹<http://www.gnuplot.info/>

²<http://cic.nist.gov/vrml/vbdetect.html>

4.2 Courbure de courbes connues

Pour tester la validité du programme, nous avons commencé par lui faire analyser des courbes simples dont le rayon de courbure est connu. Nous avons procédé à ces tests pour différents grains et différents ordres.

Pour n'importe quelle droite naïve 3D (nous avons analysé un panel d'une quinzaine d'ensembles de points de droites naïves, connexes, non connexes et dans différents plans) quel que soit l'ordre ou le grain, le résultat est exact, puisque le programme trouve une courbure nulle partout.

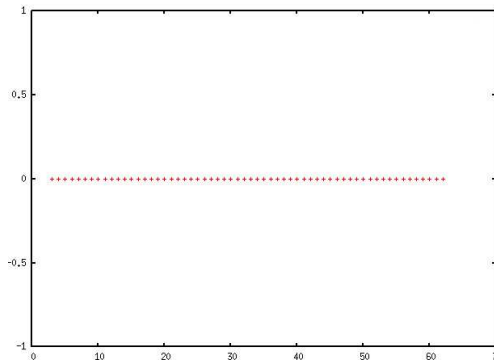


FIG. 4.3 – Le résultat du rayon de courbure discret sur une droite

On pourra aussi remarquer que pour toutes les courbes planes (cercles, droites, assemblages de cercle et cela dans n'importe quel plan), la torsion est constamment nulle ce qui était le résultat attendu puisque le plan osculateur est constant (et égal au plan auquel la courbe appartient).

Pour une hélice circulaire, c'est-à-dire une courbe obéissant à l'équation :

$$\begin{cases} x = R \cos t \\ y = R \sin t \\ z = lt \end{cases} \quad (4.1)$$

On a pour rayon de courbure $R_c = \frac{R}{R^2 + l^2}$. Ce sont bien les résultats que nous constatons sur les figures 4.6 , 4.4 et 4.5. Les différentes irrégularités ont deux sources : l'approximation des points réels par des points entiers mais aussi l'approximation des espaces entre points par des droites, les courbes d'origine étant non connexes. La figure 4.5 aide à se rendre compte de l'intérêt de la torsion. En effet, la courbure d'un disque de rayon 40 est identique à celle d'une hélice circulaire de rayon 40 et de pas 1. C'est donc la torsion (nulle pour le premier et comprise entre 0 et 10° pour le second) qui aidera à faire la différence.

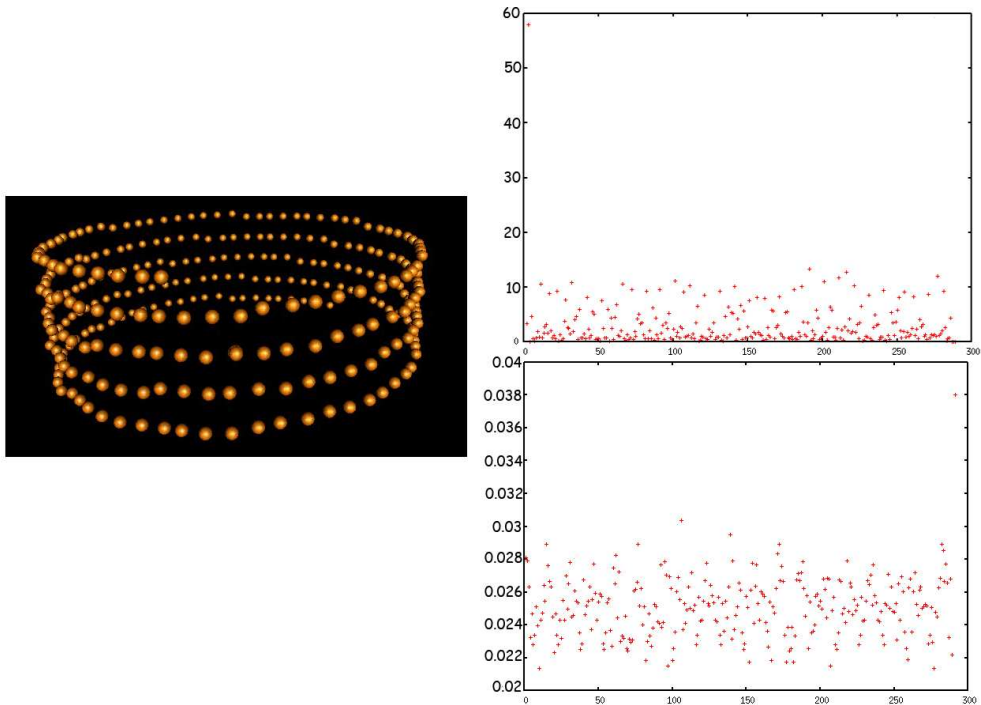


FIG. 4.4 – Torsion (en haut) et courbure (en bas) d’une hélice discrète de rayon constant égal à 40 et de pas $l=1$. La courbure est à peu près constante égale à 0.025 (courbure théorique : 0.0249) et la torsion oscille entre 0 et 10° . L’irrégularité des résultats a les mêmes sources que sur la courbe précédente

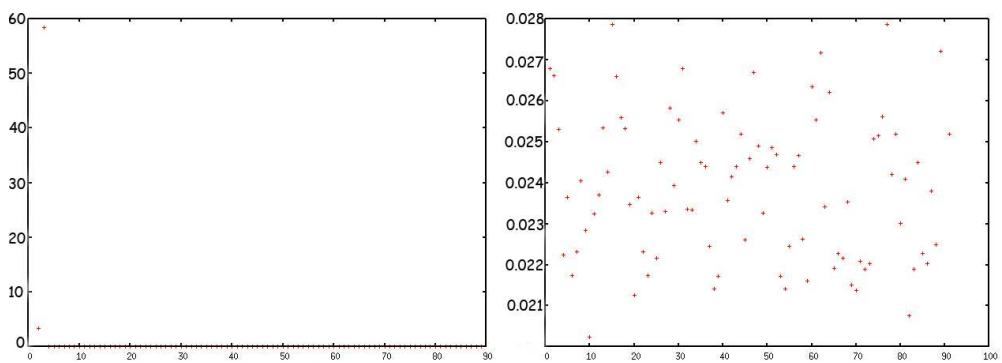


FIG. 4.5 – Torsion (à gauche) et courbure (à droite) d’un disque discret de rayon 40. Comme on peut le voir, la courbure n’est pas discernable de la courbure de l’hélice de rayon constant. C’est donc la torsion (ici nulle) qui permettra de faire la différence, ce qui justifie l’introduction de ce nouvel estimateur

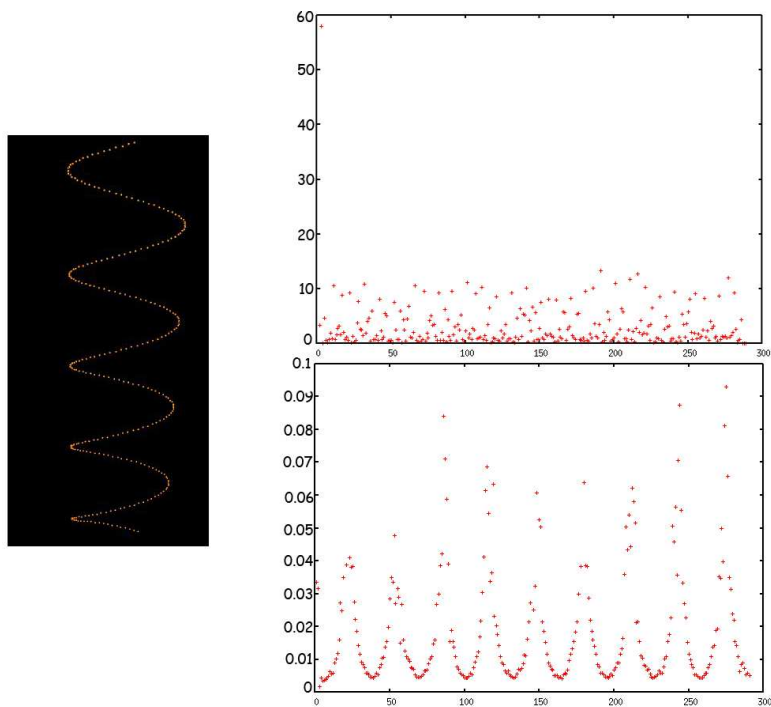


FIG. 4.6 – Torsion (en haut) et courbure (en bas) d’une hélice discrète et elliptique de demi-grand axe 55, demi-petit axe 15 et pas 15. La courbure est périodique, ce qui est logique puisque les coordonnées sur les axes x et y le sont. L’irrégularité des résultats vient de l’approximation des points réels par des points entiers mais aussi du fait que la courbe d’origine était non connexe et donc que les ”trous” ont été approximés par des droites.

4.3 Courbure de séquences d'ADN

Par manque de temps, nous n'avons pu faire tourner notre programme sur autant de séquences d'ADN que nous l'aurions souhaité et donc le valider d'un point de vue biologique. Cependant, l'exemple de la figure 4.7 confirme que nos estimateurs réagissent aux points particuliers.

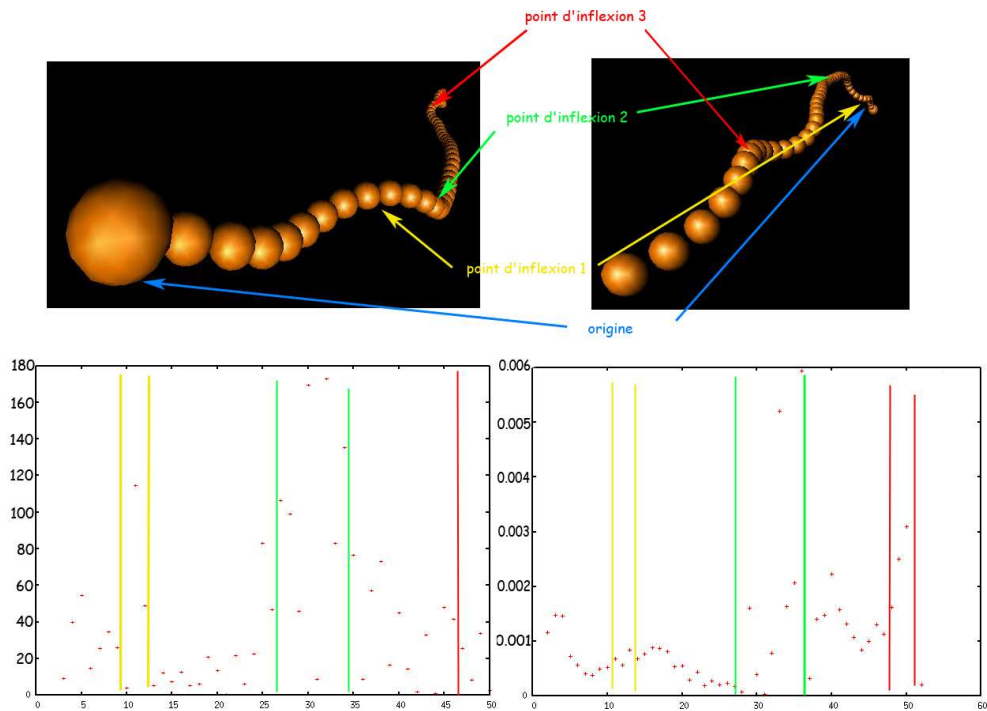


FIG. 4.7 - Torsion (à gauche) et courbure (à droite) pour la séquence TTTTTTTTTTTTTTTTTTATATATTATATATATATATTTTTTTTTTTTTTTTTTT fabriquée au hasard. Il y a trois points d'inflexion de la courbe 3D, le premier point ne se traduit que sur la torsion, le second sur torsion et courbure et le troisième principalement sur la courbure (ce qui est lié au fait que comme on approche de la fin de la courbe, on commence à manquer de points pour les calculs)

Nous avons désormais pour objectif de comparer les résultats obtenus par notre programme avec ceux obtenus par les méthodes habituelles des biologistes (entre autres CURVATURE).

Chapitre 5

Conclusion et perspectives

Au cours du stage de DEA, j'ai fait des recherches sur la structure de l'ADN, recherchant entre autres quels étaient les estimateurs choisis par les biologistes pour caractériser la forme d'une séquence. Je me suis aussi penché sur la géométrie discrète pour comprendre un peu mieux les bases de celle-ci, et comprendre comment calculer le rayon de courbure. Avec l'aide d'Isabelle Debled-Rennesson, ma responsable de stage, j'ai étendu les notions de segments flous et de courbure discrète d'ordre d à la dimension 3. Nous avons aussi introduit la notion de torsion discrète et plus encore de torsion à l'ordre d . J'ai programmé un logiciel qui regroupait toutes ces notions pour pouvoir l'appliquer sur des séquences d'ADN, ce qui était le but premier de ce stage. J'ai aussi appris à manipuler différents outils, en particulier VRML, xfig, gnuplot, L^AT_EX et The Gimp.

Malheureusement, par manque de temps, nous n'avons pu tester le code sur beaucoup de séquences d'ADN ni même nous pencher plus avant sur l'interprétation des résultats d'un point de vue biologique. Ceci est au moins en partie justifié par le fait que, à cause de la fin un peu plus tardive des programmes à Supélec, j'ai commencé mon stage en retard par rapport à mes collègues. Cependant, restant stagiaire deux mois supplémentaires au Loria, je compte bien rattraper ce léger retard.

Même si notre programme est d'ores et déjà fonctionnel, il nous reste encore un certain nombre de choses à faire pour améliorer ce programme :

- Il faut que nous trouvions plus de séquences d'ADN remarquables par leurs propriétés structurelles, dans l'optique de faire analyser ces molécules par notre code pour confirmer (ou éventuellement infirmer) la validité de notre approche. Pour cela, nous avons prévu de contacter différents biologistes spécialistes du problème pour qu'ils nous orientent dans notre recherche. Nous avons entre autres la volonté de nous pencher sur certains sites de la bactérie E.Coli sur laquelle Alexander Bolshoy dit avoir relevé des formes particulières de courbure.
- Nous pouvons aussi nous pencher sur un autre modèle de calcul de la structure 3D que celui d'Alexander Bolshoy (utilisé dans CURVATURE). En effet, nous avons choisi celui-ci parce que la forme des résultats fournis par CURVATURE était facilement exploitable mais aussi parce que ce modèle, cité dans de nombreux articles, est un modèle qui a fait ses preuves. Cependant, des modèles plus récents existent, et il est possible que certains d'entre eux fournissent une structure plus proche de la réalité. Notre calcul du rayon étant complètement indépendant du calcul du modèle tubulaire, leur intégration dans notre programme ne devrait pas poser trop de problèmes.

- L'algorithme que nous utilisons pour le calcul de la courbure n'est pas encore optimal. En effet, pour chaque nucléotide, nous recalculons le plus long segment flou à gauche et à droite indépendamment des résultats déjà obtenus. Il est sans doute possible d'imaginer un algorithme plus rapide qui prendrait en compte les résultats déjà calculés. Cependant, par manque de temps, nous avons préféré nous concentrer sur un algorithme un peu plus lent mais plus facile à mettre en oeuvre.
- L'outil que nous utilisons pour observer la forme des molécules d'ADN (le VRML) a pour principal avantage d'être multi-plateforme et d'utiliser des visualisateurs standardisés. Cependant, il est principalement conçu pour l'interactivité avec un univers tri-dimensionnel et non pas pour la restitution d'une forme en 3D sur le papier. On pourrait donc développer un visualisateur non multi-plateforme mais mieux approprié à notre problème (par exemple avec un système qui permettrait d'observer l'évolution de la courbure et de la torsion simultanément de la forme de la molécule).
- La forme de l'ADN change sensiblement selon le milieu dans lequel elle se trouve. En effet, la molécule est flexible et, dans le milieu cellulaire, comme elle n'est pas isolée, les interactions moléculaires font qu'elle n'est jamais tout à fait dans la forme du modèle tubulaire. Si l'étude de tous les cas possibles est bien trop complexe pour être envisagée, il pourrait en revanche être intéressant de s'intéresser à des cas très particuliers comme la forme de l'ADN lors de sa réaction avec certains enzymes ou protéines.
- Le but final est de mettre à la disposition de la communauté biologique un outil web complet et simple d'utilisation permettant à la fois la visualisation et l'analyse de la structure 3D associée à la modélisation d'une séquence d'ADN.

Bibliographie

- [1] A Bolshoy, P McNamara, RE Harrington, and EN Trifovon. Curved dna without a-a : experimental estimation of all 16 dna wedge angles. *Proc Natl Acad Sci USA*, March 1991.
- [2] S Cacchione, P De Santis, D Foti, and A znd Savino M Palleschi. Periodical polydeoxynucleotides and dna curvature. *Biochemistry*, October 1989.
- [3] D Coeurjolly. *Algorithmique et géométrie discrète pour la caractérisation de courbes et surfaces*. PhD thesis, Université Claude Bernard, Lyon, 2002.
- [4] D Coeurjolly, I Debled-Rennesson, and O Teytaud. Segmentation and length estimation of 3d discrete curves. *In Digital and Image Geometry, Lecture Notes in Computer Science*, 2243, 2002.
- [5] D Coeurjolly and S Svensson. Estimation of curvature along curves with application to fibres in 3d images of paper. *SCIA 2003*, 2003.
- [6] I Debled-Rennesson. *Etude et reconnaissance de droites et plans discrets*. PhD thesis, Université Louis Pasteur, Strasbourg, 1995.
- [7] I Debled-Rennesson. Estimation of tangents to a noisy discrete curve. *Vision Geometry XII, SPIE, Electronic Imaging, San Jose*, Janvier 2004.
- [8] I Debled-Rennesson, JL Rémy, and J Rouyer-Degli. Segmentation of discrete curves into fuzzy segments. *9th International Workshop on Combinatorial Image Analysis, Electronic Notes in Discrete Mathematics*, 12, May 2003.
- [9] I Debled-Rennesson and JP Réveillès. A linear algorithm for segmentation of digital curves. *International Journal of Pattern Recognition and Artificial Intelligence*, 9, December 1995.
- [10] RE Dickerson. Dna bending : the prevalence of kinkiness and the virtues of normality. *Nucleic Acids Research*, 26, 1998.
- [11] M Dlakic and RE Harrington. Diamod : display and modeling of dna bending. *Bioinformatics*, 14, 1998.
- [12] AE Gabrielian, D Landsmann, and A Bolshoy. Curved dna in promoter sequences. *In Silico Biology*, 1, 1999.
- [13] J Herisson and R Gherbi. Model-based prediction of the 3d trajectory of huge dna sequences. 2001.
- [14] <http://www.dnaftb.org/19/concept/index.html>. Dna from the beginning, chapitre 19 et suivants.
- [15] <http://www.imb.jena.de>. Image library of biological macromolecules.
- [16] D Masotti, E Ficarra, and E Macii. Techniques for enhancing computation of dna curvature molecules. *EEE International Symposium on Bioinformatics and Bioengineering (BIBE)*, May 2004.

- [17] MG Munteanu, K Vlahovicek, S Parthasarthy, I Simon, and S Pongor. Rod models of dna : sequence-dependent anisotropic elastic modelling of local bending phenomena. *TIBS*, 23, September 1998.
- [18] F Reinhardt and H Soeder. Torsion discrète. *Atlas des Mathématiques*, 1997.
- [19] JP Réveillès. *Géométrie discrète, calculs en nombre entiers et algorithmique*. PhD thesis, Université Louis Pasteur, Strasbourg, 1991.
- [20] ES Shpigelman, EN Trifonov, and A Bolshoy. Curvature : software for the analysis of curved dna. *Computer Application in the Biosciences*, 9, 1993.