



# SPLINES DYNAMIQUES GÉOMÉTRIQUEMENT EXACTES : SIMULATION HAUTE PERFORMANCE ET INTERACTION

## THÈSE

présentée et soutenue le 12 décembre 2007

pour l'obtention du

**Doctorat de l'Université des Sciences et Technologies de Lille**

**spécialité Informatique**

par

**Adrien Theetten**

Composition du jury :

Pr	YANNICK RÉMION	CRÉSTIC, Groupe SIC, LERI	Président du Jury
Pr	DINESH MANOCHA	University of North Carolina	Rapporteur
Dr	YOHAN PAYAN	Laboratoire TIMC-IMAG	Rapporteur
Dr	FRÉDÉRIC BOYER	Ecole des Mines de Nantes, IRCCyN	Rapporteur
D.R.	THIERRY GOUDON	Laboratoire Paul Painlevé	Examineur
Pr	MICHEL PETITOT	LIFL	Examineur
Pr	CHRISTOPHE CHAILLOU	LIFL, INRIA	Directeur
Dr	CLAUDE ANDRIOT	CEA LIST	Co-Encadrant
Dr	LAURENT GRISONI	LIFL	Co-Encadrant

UNIVERSITÉ DES SCIENCES ET TECHNOLOGIES DE LILLE

LABORATOIRE D'INFORMATIQUE FONDAMENTALE DE LILLE

U.F.R. d'I.E.E.A. – Bât. M3 – 59655 VILLENEUVE D'ASCQ CEDEX

Tél. : +33 (0)3 28 77 85 41 – Télécopie : +33 (0)3 28 77 85 37 – email : direction@lifl.fr



*À ma femme qui a enduré avec coeur mes pérégrinations géographiques, professionnelles  
et intellectuelles pendant ces trois dernières années...*



# Remerciements

Ces trois dernières années figureront certainement parmi les plus intenses de ma vie. Cette thèse a fait l'objet d'intenses réflexions, aiguillées avec talent par mes encadrants Laurent Grisoni du LIFL et Claude Andriot du CEA, ainsi que mon directeur de thèse Christophe Chaillou.

Je remercie tout particulièrement mes rapporteurs, Dinesh Manocha, Yohan Payan et Frédéric Boyer, de leur indulgence et de leurs critiques bénéfiques à l'égard de ces travaux. Ma gratitude va également aux autres membres du jury, qui ont apporté un regard neuf et pertinent sur les résultats obtenus.

Ceux-ci ne sont pas le fruit d'une seule personnes, mais de tout un ensemble d'hommes et de femmes compétentes et passionnées par la recherche. Christian Duriez et Xavier Merlhiot, par leurs conseils avisés, m'ont apporté les fondements nécessaires et les pistes qui m'ont donné force de proposition ; qu'il en soient vivement remerciés et aient conscience de ma pleine reconnaissance. Nombreux sont mes collègues qui m'ont également apporté des critiques constructives, des encouragements : Jérémie Dequidt, Maud Marchal, Stéphane Cotin, Julien Lenoir, Damien Marchal, Pierre-Jean Bensoussan, Sylvère Fontenau, Frédéric Blondel, Julien Vandaele, Jean-Philippe Deblonde, Mélisande Biet, Samuel Degrande, Géry Casiez et Betty Semail à Villeneuve d'Ascq ; Hatem Charfi, Laurent Mertz, Guillaume Saupin, Cyrille Colette, Mathieu et Julien à Fontenay-aux-Roses.

Villeneuve d'Ascq, le 8 avril 2008.



# TABLE DES MATIÈRES

TABLE DES MATIÈRES	1
INTRODUCTION	5
A    MOTIVATIONS ET ENJEUX	5
B    CADRE ET OBJECTIFS DE LA THÈSE	6
C    ORGANISATION DU DOCUMENT	7
I  ÉTAT DE L'ART DES DÉFORMATIONS ET INTERACTIONS D'OBJETS 1D	9
A    NOTIONS DE MÉCANIQUE DES MILIEUX CONTINUS	10
1    Hypothèses et terminologie de la mécanique des milieux continus	10
a    Milieu continu	10
b    Description du milieu	10
c    Contraintes et déformations	10
d    Linéarité et non linéarité	11
e    Domaines de déformations	12
2    Déformations réversibles : élasticité	12
a    Elasticités linéaire et non-linéaire	12
b    Paramètres mécaniques d'un matériau élastique	13
c    Relations élastiques	14
d    Modèles élastiques	14
3    Déformations irréversibles : plasticité, rupture et fatigue	15
a    Limites d'élasticité	16
b    Plasticité	17
c    Fatigue	17
d    Modèles de déformations plastiques	17
e    Modèles de fracture	18
B    SIMULATION D'OBJETS DÉFORMABLES 1D	19
1    Théorie des poutres	19
a    Définitions et terminologie	20
b    Historique	21
c    Théories et hypothèses usuelles	22
2    Déformations d'une poutre	23
a    Modélisation des efforts intérieurs	23
b    Relation entre la contrainte et le torseur de cohésion	24
c    Effort de traction-compression	25
d    Moment fléchissant	26
e    Couple de torsion	27
f    Effort tranchant	27
3    Discrétisation spatiale	28
a    Approches particulières	28
b    Approches chaînées	31
c    Approches continues	35
4    Résolution numérique	43

	a	Résolution quasi-statique . . . . .	43
	b	Résolution dynamique . . . . .	45
5		Gestion des interactions . . . . .	47
	a	Détection de collision . . . . .	47
	b	Réponse à la collision . . . . .	49
	c	Manipulation . . . . .	50
	d	Schémas de couplage . . . . .	50
		<b>BILAN . . . . .</b>	<b>51</b>
<b>II DÉFORMATIONS D'UNE SPLINE MATÉRIELLE</b>			<b>55</b>
A		<b>MODÉLISATION D'UNE SPLINE MATÉRIELLE</b>	<b>56</b>
1		Spline géométrique . . . . .	56
	a	Définitions et propriétés . . . . .	56
	b	Splines d'interpolation . . . . .	58
	c	Splines d'approximation . . . . .	59
	d	Splines précédemment utilisées pour la mécanique . . . . .	61
	e	Choix du modèle spline pour la mécanique . . . . .	61
2		Matérialisation de la spline . . . . .	65
	a	Section de la spline matérielle . . . . .	65
	b	Equation de la spline matérielle . . . . .	66
	c	Répartition de la masse . . . . .	67
	d	Repère local, courbure et torsion . . . . .	67
	e	Enveloppe géométrique de la spline matérielle . . . . .	70
B		<b>MODÈLE MÉCANIQUE SPLINE À DÉFORMATIONS RÉVERSIBLES</b>	<b>71</b>
1		Hypothèses simplificatrices de départ . . . . .	71
2		Equations de Lagrange . . . . .	72
3		Forces et moments généralisés de déformation d'une spline matérielle . . . . .	73
	a	Force généralisée de traction-compression . . . . .	74
	b	Moment généralisé de torsion . . . . .	75
	c	Moment généralisé de flexion . . . . .	75
4		Inertie d'une spline matérielle . . . . .	76
5		Résolution efficace du système d'équations . . . . .	77
	a	Schéma implicite linéarisé . . . . .	77
	b	Matrice de raideur du système . . . . .	78
C		<b>MODÈLE MÉCANIQUE SPLINE À DÉFORMATIONS IRRÉVERSIBLES</b>	<b>79</b>
1		Limite d'élasticité . . . . .	80
2		Déformations plastiques . . . . .	80
	a	Plasticité parfaite . . . . .	80
	b	Rupture . . . . .	81
	c	Plasticité, rupture . . . . .	82
D		<b>VALIDATIONS MÉCANIQUES</b>	<b>83</b>
1		Déformations . . . . .	83
	a	Validation de la traction : la chaînette . . . . .	83
	b	Validation de la flexion : spline matérielle encastree soumise à une force à son extrémité . . . . .	83
	c	Validation de la torsion . . . . .	86
	d	Sections variables . . . . .	86
	e	Etat initial déformé . . . . .	87
	f	Plasticité et rupture . . . . .	88
2		Dynamique . . . . .	88
3		Non-linéarités géométriques . . . . .	88
		<b>BILAN . . . . .</b>	<b>90</b>

<b>III</b>	<b>INTERACTIONS D'UNE SPLINE MATÉRIELLE</b>	<b>91</b>
A	CONTRAINTES . . . . .	92
1	Application de forces . . . . .	92
a	Forces ponctuelles . . . . .	92
b	Forces uniformes . . . . .	92
c	Frottements visqueux . . . . .	93
2	Multiplicateurs de Lagrange . . . . .	93
3	Contraintes sur un point . . . . .	95
a	Contrainte de position . . . . .	95
b	Contrainte d'orientation . . . . .	95
c	Contrainte de distance . . . . .	96
d	Contrainte de plan . . . . .	97
e	Contrainte d'axe . . . . .	97
f	Contrainte de point glissant . . . . .	98
g	Contrainte de point glissant orienté . . . . .	99
h	Contrainte d'abscisse glissante . . . . .	101
4	Contraintes sur plusieurs points . . . . .	101
a	Contrainte de positions relatives . . . . .	102
b	Contrainte d'orientations relatives . . . . .	102
c	Contrainte d'encastrement coplanaire relatif . . . . .	103
d	Contrainte de distance relative . . . . .	103
e	Contraintes barycentriques . . . . .	104
5	Manipulation et interaction par contraintes . . . . .	105
a	Modélisation de clips de fixation . . . . .	105
b	Couplage . . . . .	105
B	COLLISION . . . . .	107
1	Détection de la collision . . . . .	107
a	Sphères . . . . .	107
b	Cylindres droits . . . . .	107
c	Cylindres généralisés . . . . .	108
d	Segments de splines . . . . .	110
e	Bilan . . . . .	110
2	Réponse à la collision . . . . .	110
C	APPLICATIONS . . . . .	111
1	Câble simple sur portière de voiture . . . . .	111
2	Faisceau de câbles sur structure latérale de voiture . . . . .	111
3	Double hélice de l'ADN . . . . .	112
4	Pont . . . . .	113
5	Collier de perles . . . . .	113
	BILAN . . . . .	115
<b>IV</b>	<b>MODÈLE GÉOMÉTRIQUEMENT EXACT POUR LE TEMPS-RÉEL</b>	<b>117</b>
A	ARCHITECTURE ET ALGORITHMIQUE TEMPS-RÉEL . . . . .	118
1	Segments de splines . . . . .	118
2	Placement des inconnues du système . . . . .	119
a	Placement des points de contrôle . . . . .	119
b	Placement dynamique des contraintes . . . . .	119
c	Calcul des indices des points de contrôle et des contraintes dans le système . . . . .	122
3	Résolution du système matriciel à structure bande . . . . .	122
4	Matrices globales obtenues . . . . .	124
B	SIMULATION QUASI-DYNAMIQUE . . . . .	124
1	Entre simulation dynamique et simulation quasi-statique . . . . .	126

	a	Simulation dynamique . . . . .	126
	b	Simulation quasi-statique . . . . .	126
	c	Lien entre les simulations dynamiques et quasi-statiques . . . . .	127
	d	Transitions entre dynamique et quasi-statique . . . . .	127
	2	Heuristiques de passage . . . . .	128
<b>C</b>		<b>RÉCAPITULATIF ET RÉSULTATS . . . . .</b>	<b>131</b>
	1	Vue d'ensemble . . . . .	131
	2	Performances et limitations . . . . .	132
	a	Déformations . . . . .	132
	b	Contraintes . . . . .	134
	c	Récapitulatifs des diverses expérimentations . . . . .	135
		<b>BILAN . . . . .</b>	<b>135</b>
		<b>CONCLUSION . . . . .</b>	<b>137</b>
<b>A</b>		<b>RÉSUMÉ DES CONTRIBUTIONS . . . . .</b>	<b>137</b>
	1	Précision . . . . .	137
	2	Interaction . . . . .	138
	3	Performance . . . . .	138
<b>B</b>		<b>PERSPECTIVES . . . . .</b>	<b>138</b>
	1	Contrainte de déplacement relatif . . . . .	139
	2	Collision . . . . .	139
	3	Répartition adaptative des points de contrôle . . . . .	139
	4	Simulation de l'ADN . . . . .	140
	5	Simulation mécanique d'objets 3D à partir d'un squelette spline . . . . .	140
		<b>BIBLIOGRAPHIE . . . . .</b>	<b>141</b>
		<b>ANNEXES . . . . .</b>	<b>157</b>
<b>A</b>		<b>NOTATIONS . . . . .</b>	<b>158</b>
<b>B</b>		<b>LEXIQUE . . . . .</b>	<b>159</b>
<b>C</b>		<b>MATÉRIAUX . . . . .</b>	<b>160</b>
<b>D</b>		<b>PROPRIÉTÉS DES SECTIONS . . . . .</b>	<b>161</b>
	D.1	Section circulaire . . . . .	161
	D.2	Section tubulaire . . . . .	161
	D.3	Section elliptique . . . . .	161
	D.4	Section carrée . . . . .	162
	D.5	Section rectangulaire . . . . .	162
	D.6	Section en double T . . . . .	162
<b>E</b>		<b>EQUATION DE LA CHÂINETTE ET AUTRES RÉSULTATS RELATIFS . . . . .</b>	<b>163</b>
	E.1	Bilan des forces . . . . .	163
	E.2	Longueur de la chaînette . . . . .	164
	E.3	Flèche . . . . .	164
<b>F</b>		<b>LES OBJETS UNIDIMENSIONNELS DANS NOTRE ENVIRONNEMENT . . . . .</b>	<b>165</b>
	F.1	Matériaux et mécanismes . . . . .	165
	F.2	Etude et animation du vivant . . . . .	171

# Introduction

**Q**UEL est le point commun entre un câble, un cheveu, de l'herbe, une chaîne, un cathéter ou même un spaghetti ? Ce sont tous des objets dits 1D, appelés également unidimensionnels, ou monodimensionnels ; des objets pour lesquels une des trois dimensions de l'espace est nettement plus importante que les deux autres. Le fait d'avoir une dimension privilégiée permet de faciliter leur étude par le biais d'hypothèses simplificatrices. Celles-ci peuvent être notamment de nature géométrique, mécanique, électro-magnétique ou thermodynamique... Nous nous intéressons ici aux déformations et aux interactions des objets 1D avec leur environnement, dans le but de les simuler, c'est-à-dire reproduire et prédire leur comportement dans un monde virtuel. Nous souhaitons proposer un modèle général, qui convient à toutes les applications scientifiques existantes.

## A MOTIVATIONS ET ENJEUX

L'étude des objets monodimensionnels constitue un enjeu important dans de nombreux domaines.

Dans les transports, l'architecture et l'industrie, elle correspond à une problématique de *précision et de fiabilité*. Si la corde a été utilisée comme base de transport bien avant l'antiquité, son usage est bien longtemps demeuré empirique. Certes, de Vinci, Euler et Bernoulli ont esquissé les fondements théoriques d'une mécanique unidimensionnelle, dès la Renaissance. Mais ce n'est qu'à la fin du XIX<sup>ème</sup> siècle, après les travaux de Navier, Poisson et Green, que la science s'est immiscée de façon remarquable dans la conception architecturale : la Tour Eiffel en est l'un des exemples les plus flagrants. Elle a ensuite permis l'utilisation de plus en plus massive de câbles puis de poutres métalliques et bétonnées, dans des constructions de plus en plus audacieuses : les ponts du Golden Gate et de Normandie, l'opéra de Sydney et d'innombrables gratte-ciel. On n'envisage plus aujourd'hui de construire un bâtiment important sans une étude mécanique préalable. Celle-ci reste complexe et doit tenir compte de nombreux paramètres. Elle ne suffit d'ailleurs pas toujours : par exemple, la destruction du pont de Tacoma en 1940 a résulté de phénomènes oscillatoires non prévus lors de sa conception. C'est dans le domaine industriel que les corps unidimensionnels font actuellement l'objet d'intenses recherches : bras articulés de robot en milieu hostile pour l'être humain, positionnement de câbles dans des structures automobiles ou aéronautiques, et même imitation de l'appareil musculaire d'une anguille pour découvrir de nouveaux modes de transports aquatiques.

En ce qui concerne le monde des vivants, l'étude des objets 1D vise d'abord à mieux comprendre les mécanismes naturels du corps. L'acide désoxyribonucléique (A.D.N.) concentre une bonne partie des efforts des biologistes. Il contribue pour beaucoup à l'expression de notre patrimoine génétique. Ses déformations joueraient un rôle crucial dans le processus de réplication des gènes. D'un autre côté, l'étude des objets 1D cherche à reproduire les déformations des organes pour pouvoir opérer plus précisément sur le corps humain. En effet, de nombreux organes internes sont assimilables à des objets uni-

dimensionnels : vaisseaux sanguins, tendons, muscles, intestins, etc. La plupart des outils d'intervention chirurgicale le sont également : fils et épingles intervenant dans la suture de tissus, cathéters dans les vaisseaux sanguins. . . *Comportement physique plausible, interaction et temps interactif* sont cette fois nécessaires. Certains chercheurs s'intéressent à reproduire des phénomènes complexes comme la rupture systématique d'un spaghetti en trois parties, ou la montée en hélice des plantes grimpanes autour d'une tige.

L'animation vise elle à reproduire visuellement, voire même à amplifier, les comportements dynamiques d'objets tels que les cheveux, les poils, l'herbe ou les arbres, pour donner vie à un univers. L'objectif n'est pas donc pas d'être très précis, mais d'*obtenir le plus rapidement possible un résultat suffisamment crédible pour l'oeil humain*. Néanmoins, les animateurs se tournent de plus en plus vers la simulation physique pour satisfaire l'exigence croissante des spectateurs. L'utilisation des cheveux et des poils dans les films d'animation est généralement massive et requiert par conséquent de nombreuses heures de calculs pour quelques secondes d'animation : les 60.000 cheveux d'*Aki Ross*, du film *Final Fantasy*, les créatures de *l'esprit*, ou les 2.320.413 poils bleus de *Sully*, de *Monster Inc.*, ont monopolisé plusieurs semaines de calculs sur des milliers de processeurs. A l'opposé, dans les jeux vidéos, la simplification est généralement poussée à l'extrême pour obtenir des mouvements en temps-réel. Des moteurs physiques commencent cependant à intégrer des modules dédiés à la simulation de tels objets.

De nombreuses autres utilisations sont faites, nous invitons le lecteur à consulter l'annexe F pour une étude plus exhaustive.

## B CADRE ET OBJECTIFS DE LA THÈSE

De ces domaines scientifiques que nous venons d'énumérer, les motivations des chercheurs et leur modèle diffèrent parfois considérablement. Le cadre de cette thèse permet de préciser nos objectifs. Elle s'est effectuée au travers d'une cotutelle entre d'une part, le *Laboratoire Commande Interface(LCI)* du *Laboratoire d'Intégration des Systèmes Technologiques(LIST)* au sein du *Commissariat à l'Energie Atomique(CEA)* de Fontenay-aux-Roses d'une part, et de l'autre le projet *Alcove* de l'*Institut National de Recherche en Informatique et en Automatique(INRIA)* de Lille, financé entre autres par l'Institut de Recherche sur les Composants logiciels et matériels pour l'Information et la Communication Avancée (IRCICA).

Le LCI a une forte expérience dans le domaine de la robotique. Il s'est tourné ces dernières années vers la réalité virtuelle, pour simuler physiquement la manipulation d'objets, dans le cadre de l'étude de conception et du prototypage. Il réalise notamment des études de faisabilité pour des partenaires industriels comme *Renault*, *PSA*, *Dassault Aviation*, *EADS*. . . Parmi les applications proposées, certaines font intervenir la manipulation de câbles électriques et pneumatiques, de nature peu déformable.

Le projet *Alcove*, partie intégrante du *Laboratoire d'Informatique Fondamentale de Lille (LIFL)*, est issu de la communauté d'informatique graphique et de la réalité virtuelle. L'animation physique, essentiellement dans le domaine médical, constitue un domaine important de ses activités. Plusieurs travaux menés dans cette équipe ont conduit entre autres aux plate-formes logicielles *SPORE* et *SOFA*<sup>1</sup>. La simulation de l'opération de la cataracte ou des intestins constitue quelques-uns de ses faits d'armes. Cette thèse fait notamment suite aux travaux de *Lenoir* ([Leno4](#)) sur un modèle 1D déformable temps-réel à base de splines dynamiques. Son objectif principal était de fournir un modèle de fil chirurgical capable de reproduire le comportement de suture en temps-réel.

Trois critères nécessaires pour la simulation d'objets 1D ressortent de toutes les applications que nous venons d'énumérer : la *précision*, le *pouvoir d'interaction* et la *performance*. La précision est fondamentalement au coeur des préoccupations des sciences de la mécanique. De nombreux modèles tentent d'approcher visuellement les déformations d'objets,

<sup>1</sup><http://www.sofa-framework.org>

mais un environnement de production en industrie ne peut se satisfaire de telles approximations. Il est donc nécessaire de passer par des étapes d'intégration et de validation des paramètres mécaniques. Le potentiel d'interaction implique d'une part de pouvoir manipuler intuitivement et précisément l'objet dans l'espace de travail, et d'autre part un retour sur le travail de l'utilisateur au moins visuel, dans l'idéal également haptique. La performance est inhérente à la réalité virtuelle. Les calculs doivent s'effectuer suffisamment rapidement pour qu'il n'y ait pas de décalage temporel entre l'environnement créé et celui de l'utilisateur.

Comme nous le verrons dans l'état de l'art, le nombre d'articles traitant de la simulation d'objets unidimensionnels a considérablement crû ces dernières années, mais à notre connaissance, aucune méthode qui satisfasse à la fois ces trois exigences n'a été proposée. *Notre objectif est de proposer un modèle qui concilie à la fois précision et performance, tout en offrant un fort pouvoir d'interaction.* Nos travaux sont essentiellement fondés sur les articles précurseurs de Rémion, Nourrit, Nocent et Lenoir, traitant des splines matérielles dynamiques, et sur la théorie des poutres, mécanique des milieux continus spécifique aux objets 1D. Tout au long de ce mémoire, un effort pédagogique tout particulier est fait sur la clarté des justifications et des explications, afin qu'un informaticien non mécanicien puisse comprendre et implémenter facilement notre proposition de modèle déformable 1D.

## C ORGANISATION DU DOCUMENT

Ce mémoire est constitué de quatre chapitres.

Le chapitre I effectue une brève excursion dans la simulation des objets déformables 3D, plus générale mais moins adaptée à ce problème. Il dresse ensuite un état de l'art exhaustif de la simulation des objets déformables 1D. Plusieurs points sont abordés, notamment les déformations réversibles et irréversibles, l'application de contraintes et enfin la gestion de la collision.

Le chapitre II présente notre proposition de modèle 1D, à savoir la spline matérielle géométriquement exacte. Nous montrons qu'un objet 1D peut être modélisé de manière précise à partir d'un outil géométrique tel que la spline, à laquelle une section est ajoutée pour lui donner un volume, et d'outils mécaniques, donnés par la théorie des poutres. Nous proposons d'abord une méthode pour évaluer la précision mécanique apportée par l'interpolation spline et la comparer à celle obtenue par la méthode des éléments finis. Nous apportons ensuite une formulation géométriquement exacte des déformations en traction-compression, en flexion et en torsion d'une telle spline, puis nous déterminons un schéma de résolution adapté, stable et efficace. Nous terminons par quelques validations qui achèvent notre étude sur les déformations.

Le chapitre III s'intéresse aux interactions de la spline avec elle-même et son environnement. Nous rappelons d'abord comment ajouter au système des forces ponctuelles ainsi que de densités de force. Nous proposons ensuite une formulation des contraintes dures, compatible avec le mode de résolution que nous venons d'adopter, et plus efficace que celles existantes pour les splines. Nous apportons aussi de nouvelles contraintes, comme les contraintes de distance ou la modélisation de particules dont la trajectoire est imposée par la spline. Nous présentons également quelques utilisations des contraintes dures, pour la modélisation de clips de fixation ou la réalisation d'un couplage avec un autre modèle mécanique. Dans le cadre de la détection de collision, nous évaluons les différentes méthodes de représentation afin de déterminer celle qui offre la meilleure précision pour un coût de calcul acceptable pour le temps-réel. Quelques applications pratiques mettent enfin en valeur l'utilisation des contraintes.

Le chapitre IV détaille la méthode proposée pour obtenir une simulation performante. A partir d'un ensemble de points de contrôle et de contraintes, nous donnons d'abord un algorithme de construction d'un système à structure bande, qui assure la complexité

linéaire du modèle de spline matérielle. Afin de réduire le nombre de calculs au strict minimum, nous détaillons toutes les symétries exploitables du système. Nous proposons ensuite une nouvelle méthode de simulation, la simulation quasi-dynamique. Elle est fondée sur le passage automatique entre les équilibres quasi-statique et dynamique en cours de manipulation. Les commutations sont déterminées par des heuristiques, qui tiennent compte des propriétés des matériaux, des actions exercées sur l'objet et du temps de calcul. Enfin, nous donnons une vue générale de la simulation d'une spline matérielle. Nous évaluons également les performances de notre implémentation.

Nous terminons par un bilan de nos travaux, portant sur leurs atouts et leurs limitations, puis nous mentionnons un certain nombre de pistes actuelles et envisageables pour la suite de nos recherches.

## Chapitre I

# Etat de l'art des déformations et interactions d'objets 1D

*Pourquoi voyez-vous une paille dans  
l'oeil de votre frère, tandis que vous ne  
voyez pas une poutre dans la vôtre ?*

SAINT MATTHIEU

**Q**UEL que soit le domaine scientifique, la simulation mécanique d'un tel objet fait plus ou moins appel aux mêmes fondements théoriques. Les modèles issus sont donc comparables, voire transférables d'un monde à un autre, bien que les communautés concernées communiquent peu ou parfois pas du tout. Nous allons donc dresser l'état de l'art des méthodes existantes qui peuvent s'appliquer aux objets déformables 1D. Nous nous intéressons d'abord à la mécanique des milieux continus. Elle est utilisée pour la simulation d'objets déformables, dont nous donnons les principales méthodes. Nous mettons ensuite en avant les limitations d'une approche générale pour la modélisation d'objets déformables 1D. Celles-ci nous amènent à reprendre les hypothèses de la mécanique des milieux continus, et à les appliquer spécifiquement aux objets 1D, au travers de théories dédiées. Un état de l'art exhaustif des modèles déformables 1D permet alors de parvenir à un choix qui correspond le mieux à nos ambitions, qui portent sur trois critères principaux : validité physique, potentiel d'interaction et performance.

## A NOTIONS DE MÉCANIQUE DES MILIEUX CONTINUS

Nous allons nous intéresser à la science qui étudie les déformations des solides : la *mécanique des milieux continus*. Quelques connaissances sont nécessaires pour déterminer la qualité et la pertinence des modélisations existantes de déformations 3D. Nous présenterons d'abord quelques hypothèses de base, ainsi qu'un peu de terminologie associée. Nous distinguerons deux types de déformations, les déformations réversibles, ou élastiques, puis les déformations irréversibles, ou plastiques. Nous présenterons les principaux modèles généraux les traitant, et nous verrons qu'ils ne sont généralement pas adaptés à une simulation précise et efficace d'objets 1D.

### 1 Hypothèses et terminologie de la mécanique des milieux continus

Les modèles simulant le comportement d'objets déformables sont nombreux. Nous donnons ici quelques éléments permettant de classer les méthodes et d'en évaluer les limites.

#### a Milieu continu

Si l'on regarde la matière à l'échelle du nanomètre, la matière est granulaire, faite d'atomes. Mais à l'oeil nu, un objet solide semble continu, c'est-à-dire que ses propriétés semblent varier progressivement, sans à-coups. L'*hypothèse des milieux continus* consiste à considérer que quelque soit la déformation subie les atomes conservent leur voisinage. Une telle hypothèse permet d'avoir recours aux outils mathématiques reposant sur des fonctions continues et dérivables. Des hypothèses supplémentaires peuvent éventuellement être faites ; ainsi un milieu continu peut être homogène : ses propriétés sont les mêmes en tout point ; isotrope : ses propriétés ne dépendent pas du repère dans lequel elles sont observées ou mesurées. De nombreux matériaux utilisés dans l'industrie sont à la fois homogènes et isotropes (métaux usinés ou bruts de fonderie). Cependant, de nombreux matériaux ne sont pas isotropes (tôles laminées, pièces forgées, pièces tréfilées...); par ailleurs, l'utilisation de plus en plus fréquente des matériaux composites a amené à étudier les milieux qui ne sont ni homogènes (éponges), ni isotropes (fibres de verre, de carbone ou de kevlar maintenues dans une résine) mais pour lesquels l'hypothèse de continuité (tout au moins par morceaux) reste valable.

#### b Description du milieu

En mécanique continue, il existe deux représentations pour décrire un milieu. En *représentation lagrangienne*, les fonctions décrivant les grandeurs dépendent d'une particule et du temps. Cette représentation suit donc chaque particule à tout instant. En *représentation eulérienne*, les fonctions décrivant les grandeurs dépendent du point géométrique considéré et du temps. Le champ eulérien considère la grandeur portée par la particule qui occupe un point géométrique à un instant donné. Il est souvent associé aux équations de Navier. La représentation lagrangienne est souvent plus intuitive au départ, tandis que la représentation eulérienne conduit à des descriptions plus simples.

#### c Contraintes et déformations

Quand une force est appliquée sur un solide déformable, sa géométrie est altérée. Sa topologie peut également changer si la force est suffisamment grande, par exemple dans le cas d'une fracture. Afin de s'abstraire des dimensions des solides, on ne travaille pas avec des forces et des moments, ni des déplacements, mais avec des *contraintes*, notées  $\sigma$ , et des *déformations*, notées  $\epsilon$ . Une contrainte est une force ou un moment ramené à une unité de surface, tandis qu'une déformation est un déplacement par unité de longueur.

Les forces sont des contraintes en translation, tandis que les moments sont des contraintes en rotation ; elles s'expriment de la façon suivante :

$$\sigma = \begin{cases} \frac{\mathcal{F}}{A} & \text{(translation)} \\ \frac{Mr}{I} & \text{(rotation)} \end{cases} \quad (\text{I.1})$$

où  $\mathcal{F}$  est une force,  $A$  l'aire de la section,  $M$  un moment,  $r$  la distance entre le centre de gravité et le point d'application du moment, et  $I$  le moment d'inertie de la section.

La déformation est quantifiée par l'*allongement relatif*  $\epsilon$ , encore appelé *déformation* :

$$\epsilon = \frac{\Delta l}{l_0} \quad (\text{I.2})$$

$\Delta l$  étant la longueur initiale et  $l_0$  l'allongement ;  $\epsilon$  est sans unité.

#### d Linéarité et non linéarité

Généralement, quand on étudie les déformations d'un objet, on suppose que la déformation est petite par rapport aux dimensions de l'objet, que le matériau est élastique et que les déformations sont localement proportionnelles aux contraintes. Ces hypothèses sont utilisées dans le but de simplifier les équations en les linéarisant. Dans la plupart des cas, les déformations sont petites, les contraintes dans le domaine de validité de l'élasticité linéaire et les équations linéaires donnent des résultats corrects. Cependant, dans le cas où des *non-linéarités* structurelles interviennent, ces hypothèses ne sont plus valides. On doit par conséquent effectuer des analyses non linéaires. Les non-linéarités structurelles sont réparties en deux catégories : les non-linéarités matérielles et les non-linéarités géométriques.

Les *non-linéarités matérielles* interviennent quand la *loi de Hooke*, qui dit qu'une contrainte est proportionnelle à la déformation, ne s'applique pas. Cela inclut par exemple l'élasticité non linéaire, la plasticité et la rupture. Si la loi de Hooke est valable, on parle de *petites déformations*, sinon, de *grandes déformations*. Elle est souvent applicable dans le cas d'un matériau isotrope, c'est-à-dire un matériau dont le comportement est le même quelle que soit la direction de l'effort. C'est le cas par exemple du cuivre pour les fils électriques, de l'acier pour les câbles, du nylon pour les fils de chirurgie... Le fait de torsader ces matériaux leur offre une meilleure résistance dans le sens de la longueur (en traction), mais leur fait perdre la linéarité matérielle.

Les *non-linéarités géométriques* se produisent quand les déformations sont telles qu'elles changent la nature du problème. On se place dans le cadre des *petits déplacements* si les hypothèses géométriques linéaires sur la structure au repos restent valables au cours de la déformation. Par contre, s'il faut tenir compte de l'évolution de la géométrie de la structure au cours de la déformation, on considère alors des *grands déplacements*. On trouve également dans la littérature les termes *rotations* (JP99; MDM<sup>+</sup>02), *transformations* (BP05), employés pour désigner les mêmes approximations. Nous attirons l'attention du lecteur francophone sur la difficulté de traduction des mots anglais *strains* et *deformations*. *strain* désigne une déformation locale, tandis que *deformation* s'emploie pour une déformation globale. L'expression "small/large deformations" désigne les petits/grands déplacements, par opposition à "small/large strains" (PDA00; RJ07) pour les petites/grandes déformations. La prise en compte ou non des non-linéarités géométriques affecte particulièrement les éléments de type poutre. Dans le cas de l'étude de la déformation d'une poutre en flexion, on peut, pour simplifier les calculs, linéariser la courbure. Si la poutre très reste proche d'un état rectiligne, les résultats seront peu faussés par l'approximation. Les études linéaire puis non-linéaire d'une poutre encastree illustreront plus loin les précisions obtenues respectives (section b, page 85). Notons enfin qu'une étude en *petites perturbations* est définie par l'utilisation conjointe des hypothèses des petits déplacements et des petites déformations (Gar07).



(a) Compression (b) Torsion (c) Flexion (d) Fatigue

Fig. I.1 – Essais de matériaux.

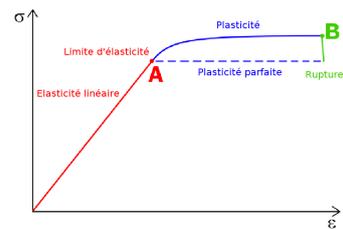


Fig. I.2 – Courbe schématique de contrainte-déformation

## e Domaines de déformations

Dans le cas général, les contraintes et les déformations sont manipulées sous forme de tenseurs  $\bar{\sigma}$  et  $\bar{\epsilon}$  (Cou80c; Gar07). Néanmoins, dans la plupart des cas, la relation entre une contrainte appliquée à un solide et la déformation résultante peut être décrite par une simple relation dans chacune des directions principales. Une *courbe de contrainte-déformation*, liant contrainte et déformation, caractérise le matériau. Elle peut être obtenue grâce à diverses méthodes plus ou moins complexes à mettre en oeuvre. Si les essais de traction ou de compression sont largement utilisés, les autres le sont dans une moindre mesure (figure I.1). Ils consistent à étirer ou comprimer le matériau jusqu'à la rupture. La force d'entrée est imposée donc connue, tandis que la déformation est mesurée au cours de l'expérience ; ces deux données permettent de tracer la courbe.

La figure I.2 représente schématiquement la courbe caractéristique de contrainte-déformation d'un matériau dans une direction. Le comportement du matériau dépend beaucoup de l'amplitude de la déformation. Les déformations peuvent être classées en trois domaines. Les déformations *élastiques*, se produisent en premier lieu. Une partie linéaire est éventuellement suivie d'une partie non-linéaire, suivant les matériaux. Les déformations *plastiques* arrivent en second lieu pour des déformations plus conséquentes. Enfin, quand la déformation atteint le point de *rupture*, la topologie locale change et la mécanique des milieux continus ne peut plus être utilisée pour modéliser le comportement du matériau. Les domaines de déformation que nous venons de décrire correspondent à deux types de comportement qui mettent en avant des problèmes différents et qui sont donc généralement simulés de manière spécifique. Les publications se concentrent généralement sur l'un ou l'autre. Nous allons donc dresser un tableau des méthodes existantes pour des déformations *réversibles* (élasticité) dans un premier temps, puis *irréversibles* (plasticité, fracture) dans un second temps.

## 2 Déformations réversibles : élasticité

La théorie de l'élasticité est une discipline fondamentale dans l'étude des milieux continus, initiée par Navier, Poisson et Green dans la première moitié du XIX<sup>ème</sup> siècle (Wes64). Elle consiste en un ensemble d'équations différentielles qui décrivent les relations entre les contraintes, les déformations et les déplacements de chaque point dans un matériau élastique.

### a Elasticités linéaire et non-linéaire

Quand l'amplitude des déformations du matériau est relativement petite, la loi de Hooke peut être appliquée. Elle stipule que les contraintes sont proportionnelles aux déformations. Le domaine d'élasticité contient seulement des déformations réversibles : quand la contrainte est relâchée, le matériau reprend sa forme de repos initiale. Une conséquence importante de la linéarité est le *principe de superposition* : si plusieurs systèmes de forces extérieures indépendants provoquent séparément des déformations élastiques et

des contraintes, leur application simultanée ou successive dans un ordre quelconque (à condition de rester dans le domaine élastique) donnera des déformations élastiques et des contraintes qui seront les sommes géométriques des déformations et des contraintes prises séparément.

Une partie élastique non-linéaire existe parfois, où l'approximation des grandes déformations est nécessaire (Cou80a). Cependant, un large éventail de matériaux que l'on retrouve dans les objets unidimensionnels sont suffisamment homogènes et isotropes pour que l'on puisse les considérer comme purement linéaires.

## b Paramètres mécaniques d'un matériau élastique

Plusieurs paramètres caractérisent un matériau élastique homogène et isotrope. Le rapport longitudinal entre une contrainte et la déformation correspondante est donnée par le *module d'Young* ou d'*élasticité longitudinale*  $E$  ; le rapport transversal est lui donnée par le *module de cisaillement* ou d'*élasticité transversale*  $G$ . Le *coefficient de Poisson*  $\nu$  permet de mesurer le rapport entre la déformation longitudinale et transversale. Les modules d'Young et de cisaillement, ainsi que le coefficient de Poisson sont liés par l'équation suivante :

$$G = \frac{E}{2(1 + \nu)} \quad (\text{I.3})$$

Les modules ont la dimension d'une pression et s'expriment en Pascals ( $1Pa = 1N.m^{-2}$ ), tandis que le coefficient de Poisson est sans dimension, sa valeur est théoriquement comprise entre  $-1$ , et  $0.5$ . Les valeurs expérimentales obtenues dans le cas d'un matériau parfaitement isotrope sont très proches de  $0.5$ . Pour un matériau quelconque, on obtient en moyenne  $0.3$ . Il existe également des matériaux à coefficient de Poisson négatif : on parle alors parfois de matériaux auxétiques (certains cristaux ou mousses). Deux de ces paramètres suffisent à caractériser un matériau donné. Certains sont donnés en annexe C. Nous invitons également le lecteur à consulter (LN91; How01) pour obtenir les valeurs d'un matériau donné.

Les *coefficients de Lamé* peuvent également être utilisés comme alternative aux paramètres décrits ci-dessus :

$$\lambda = \frac{E\nu}{(1 + \nu)(1 - 2\nu)}, \nu = G = \frac{E}{2(1 + \nu)} \quad (\text{I.4})$$

Néanmoins, ce sont le module d'Young et le coefficient de Poisson que l'on retrouve dans l'immense majorité des articles pour caractériser un matériau élastique homogène et isotrope. Pour les déterminer, on procède à diverses expériences.

Le module d'Young est généralement déterminé à l'aide d'un essai de traction ou de compression (figure I.1(a)). Au cours de l'essai, on applique une force connue sur le matériau et on la fait varier au cours du temps. On mesure les déformations correspondantes, ce qui permet d'établir la courbe de contrainte-déformation. Dans le cas de l'élasticité linéaire, on obtient une droite dont le coefficient directeur est le module d'Young, calculé donc par régression linéaire.

Pour certains objets, le coefficient de Poisson peut être déterminé à l'aide d'un procédé optique (Cou80c). Pour une poutre de section rectangulaire en flexion pure, la face supérieure se transforme en un paraboloïde hyperbolique dont les lignes de niveau sont des hyperboles ; l'angle  $\alpha$  des asymptotes de ces hyperboles est directement lié au coefficient de Poisson :  $\tan^2 \frac{\alpha}{2} = \nu$  L'expérience consiste à polir la face supérieure et à placer sur elle une plaque de verre coïncidant avec cette face. Le vide obtenu peut être mesuré en dirigeant normalement à la plaque un faisceau de lumière monochromatique. On obtient des franges hyperboliques, analogues aux anneaux de Newton, qui dessinent les lignes de niveau de la face supérieure de la poutre.

Les coefficients de Lamé servent également à décrire les relations entre les contraintes principales et les déformations principales dans un milieu continu élastique quelconque (Cou80b).

### c Relations élastiques

La loi de Hooke permet de passer des déformations aux contraintes à l'intérieur de l'objet. Elle s'énonce en trois dimensions, pour des petites déformations, en fonction des coefficients de Lamé :

$$\bar{\sigma} = 2\mu\bar{\varepsilon} + \lambda tr(\bar{\varepsilon})\bar{I} \quad (\text{I.5})$$

où  $tr$  désigne la trace d'une matrice et  $\bar{I}$  est la matrice identité. Si l'on peut ramener le problème à une direction (une compression uniaxiale par exemple), la loi de Hooke devient :

$$\sigma = E\varepsilon \quad (\text{I.6})$$

Ensuite, les contraintes internes sont équilibrées avec les forces de volume  $f_v$  qui s'exercent sur chaque particule de l'objet :

$$div(\bar{\sigma}) + f_v = \bar{0} \quad (\text{I.7})$$

Enfin, l'hypothèse sur les déplacements s'exprime au travers de la formulation du tenseur des déformations  $\bar{\varepsilon}$ . Pour des petits déplacements, sa formulation est linéarisée en supprimant le terme quadratique. Pour plus d'informations sur les tenseurs de déformations et de contrainte, nous renvoyons le lecteur au livre de Garrigues sur la mécanique des milieux continus (Gar07).

### d Modèles élastiques

Les lois de constitutions donnent donc des équations aux dérivées partielles (EDP) qui décrivent le comportement du matériau. Ces équations peuvent généralement être résolues à l'aide des différences finies (TPBF87) ou des éléments finis (Bat82).

Les différences finies manquent de stabilité pour simuler les non-linéarités, même avec des degrés de dérivation élevés (HKS98), comme le mentionne Bertails dans sa thèse (Bero6). Appliquées sur un maillage, elles sont en fait un cas spécial des éléments finis. En mécanique, les simulations à base d'éléments finis volumiques sont typiquement pré-calculées pendant plusieurs minutes, voire plusieurs heures, pour un résultat de quelques secondes. Une des premières méthodes à base d'éléments finis en informatique graphique a été proposée par Gourret *et al.* (GTT89) pour simuler les déformations d'une balle serrée dans une main. Bien qu'en quasi-statique (succession d'états d'équilibres statiques, donc non dynamique), cette méthode est inexploitable dans un cadre temps-réel, d'autant qu'elle se place dans le cadre des petites perturbations (petits déplacements et petites déformations). D'autres travaux ont par la suite fourni des simulations de plus en plus rapides et stables. Grâce à leur linéarité, une partie de la solution peut être pré-calculée. Des applications ont surtout été faites dans le domaine médical : chirurgie plastique (PRZ92), animation des muscles faciaux (CZ92), tissus mous pour la chirurgie (CDA99), etc. Avec ArtDefo (Accurate Real Time Deformable Objects), James *et al.* (JP99) ont également présenté un algorithme plus général qui utilise les éléments de surface pour déformer les objets en temps-réel.

Cependant, l'hypothèse des petits déplacements produit vite des résultats inappropriés, comme le soulignent Zhuang *et al.* (ZCoo) ainsi que Müller *et al.* (MDM<sup>+</sup>02) : des déformations globales importantes se traduisent notamment par une augmentation de volume et une amplification des déformations (figure d).

Récemment, des résultats plus réalistes ont été obtenus en se plaçant dans l'hypothèse des grands déplacements (PDAoo), mais avec un surcoût très important et l'introduction d'instabilités numériques. Seuls des problèmes de taille limitée peuvent être traités en

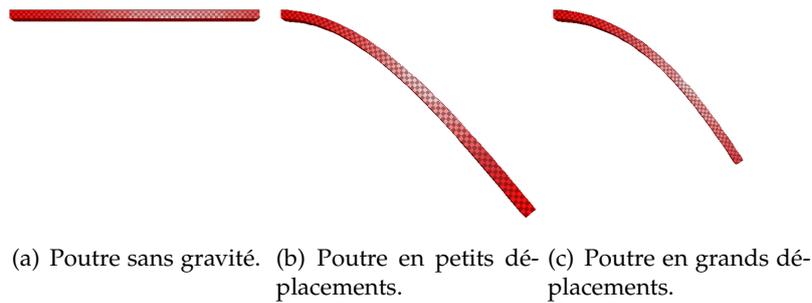


Fig. 1.3 – Mise en évidence des hypothèses de petits et grands déplacements, (ZCoo)

temps réel. Pour obtenir de meilleures performances, Zhuang *et al.* (ZCoo) approximent les calculs non-linéaires en distribuant la masse sur des points précis du solide. D'autres compromis ont également été proposés. Desbrun *et al.* (DSB99) séparent les forces dans des réseaux de masses-ressorts en deux composantes linéaires et non linéaires. La résolution s'effectue en deux étapes : la partie non-linéaire qui décrit les rotations est d'abord négligée pour calculer une rapide approximation de l'intégration implicite ; puis la première estimation est corrigée pour préserver le moment.

La multi-résolution de maillage et la simulation à pas de temps adaptatif peuvent quelque peu améliorer le compromis entre la précision et le coût en calculs (DDCB01; GKS02; CGC<sup>+</sup>02). Le gain de temps se fait néanmoins au prix de simplifications dans les déformations et d'un accroissement de la complexité du modèle.

Supposer de grandes rotations nécessite de recalculer des termes coûteux à chaque pas de temps (i.e. la matrice de raideur du système). Müller *et al.* (MDM<sup>+</sup>02) évitent ce recalcul par l'utilisation d'un champ de rotations linéaires, ce qui rend plus rapide et robuste une simulation en grands déplacements. Néanmoins, le bilan des forces n'est pas garanti nul, et des forces "fantômes" peuvent ainsi apparaître. Capell *et al.* (CGC<sup>+</sup>02) proposent de subdiviser un objet en plusieurs éléments à partir de son squelette. Pour chaque élément sont calculées des forces linéaires, ce qui limite les artefacts des petits déplacements mais introduit des discontinuités entre les éléments. Irving *et al.* (ISF07) ont présenté un modèle d'élément finis qui reproduit correctement le comportement local d'incompressibilité.

De nombreuses méthodes à base de particules ont fait leur apparition ces dernières années. Enfin, les méthodes sans maillage sont bien adaptées pour gérer de grandes rotations grâce à leur flexibilité et leur propriété de multi-résolution locale. D'un autre côté, un traitement spécial est requis pour modéliser les bords. De plus, ces méthodes sont plus coûteuses, puisque la connexité des noeuds est déterminée pendant la simulation et que l'évaluation des fonctions de forme nécessite une inversion de matrice. Müller *et al.* (MKN<sup>+</sup>04) ont cependant réussi à s'affranchir de ce problème en utilisant des schémas locaux.

Pour en savoir plus la simulation d'objets déformables 3D, nous renvoyons le lecteur aux états de l'art de Gibson *et al.* (GM97), de Nealen *et al.* (NMK<sup>+</sup>05), ainsi qu'à l'article de Rivers *et al.* (RJ07).

### 3 Déformations irréversibles : plasticité, rupture et fatigue

Nous nous intéressons maintenant aux déformations irréversibles du matériau. Elles comprennent d'abord les déformations plastiques, qui sont maintenues après relâchement des contraintes. Si l'effort est trop important, le matériau se rompt. La fatigue peut également être à l'origine d'une fracture du matériau. Nous allons exposer ces différentes notions dans cette sous-section.

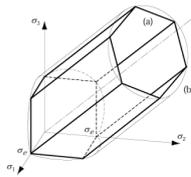


Fig. I.4 – Critères de Tresca (polyèdre) et de von Mises-Hencky (cylindre).

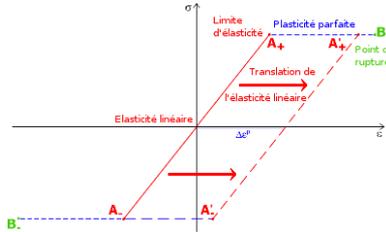


Fig. I.5 – Hystérésis de contrainte-déformation pour la plasticité parfaite.

### a Limites d'élasticité

Quand un matériau soumis à des efforts passe du comportement élastique au comportement plastique, on dit qu'il atteint sa limite d'élasticité. Cette limite peut être déterminée par des essais pour chaque déformation, mais il n'est pas toujours évident d'obtenir des résultats. Bien souvent, seule la limite en traction est disponible. La limite globale est alors déterminée à partir de celle-ci et des contraintes appliquées.

Considérons un élément de matière soumise aux contraintes principales  $\sigma_1, \sigma_2$  et  $\sigma_3$ , classées par ordre décroissant. On appelle *état limite de contrainte* toute combinaison ( $\sigma_1, \sigma_2$  et  $\sigma_3$ ) provoquant le dépassement de la limite élastique dans l'élément. De nombreux critères existent pour déterminer cet état, selon que le matériau est cassant ou ductile. Les deux critères les plus couramment utilisés pour déterminer la défaillance des matériaux ductiles soumis à des charges statiques sont celui de Tresca d'une part, et celui de von Mises-Hencky d'une autre.

Le *critère de Tresca*, nommé aussi *critère du plus grand cisaillement* ou *critère de la contrainte tangentielle maximale*, admet que la défaillance survient dès que la plus grande contrainte de cisaillement dépasse la valeur limite  $\tau_e$  déterminée par un essai de traction :

$$\tau_{max} \geq \frac{\tau_e}{2} \quad (I.8)$$

On définit le facteur de sûreté  $SF$  comme le rapport entre la valeur limite et la contrainte maximale mesurée. Pour le critère de Tresca, il est donc égal à :

$$SF = \frac{\tau_e}{2\tau_{max}} = \frac{\tau_e}{\sigma_1 - \sigma_3} \quad (I.9)$$

Le *critère de von Mises-Hencky*, ou critère d'énergie de distorsion, utilise une contrainte dite *effective* pour déterminer la défaillance. Cette contrainte  $\sigma'$ , appelée aussi *contrainte de von Mises*, est la contrainte de traction qui donnerait une distorsion d'énergie équivalente à celle étudiée :

$$\sigma' = \sqrt{\frac{(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2}{2}} \quad (I.10)$$

Le facteur de sûreté  $SF$  est ici égal à :

$$SF = \frac{\tau_e}{\sigma'} \quad (I.11)$$

Ces deux critères donnent les mêmes résultats pour des contraintes axiales, mais dans la plupart des autres cas le critère de Tresca est plus restrictif que celui de von Mises-Hencky. Etant également plus facile à calculer, le critère de Tresca est le plus largement utilisé en conception. La plus grande précision du critère d'énergie de distorsion le rend plus utile dans l'estimation des contraintes d'un matériau qui a connu une défaillance.

La figure I.4 montre les limites des deux critères dans l'espace. Si la contrainte est à l'intérieur du volume, la défaillance ne se produira pas ; dans le cas contraire, on s'attend à ce qu'elle se produise.

Si les matériaux ductiles sont les plus souvent utilisés, les matériaux cassants peuvent l'être également. Le comportement des matériaux cassants est différent en traction et en compression, d'où l'usage de deux autres critères : le critère de Coulomb-Mohr et le critère modifié de Mohr. A l'image des critères de Tresca et de von Mises-Hencky, le premier est légèrement plus restrictif et donc plus utilisé en conception, tandis que le deuxième sert à déterminer les raisons d'une défaillance. Pour plus de détails, nous invitons le lecteur à consulter ([Howo1](#)).

## b Plasticité

Si la contrainte appliquée sur la structure excède un premier seuil appelé *limite d'élasticité*, alors les déformations deviennent irréversibles ; quand une contrainte appliquée sur un matériau est relâchée, celui-ci reste partiellement déformé. On dit alors que les déformations sont plastiques. Après une déformation plastique quelconque, le point de repos et le domaine élastique de la courbe de déformation-contrainte sont déplacés dans le plan (figure [I.2](#)). Dans le cas d'une *plasticité parfaite*, la courbe est translatée selon l'axe des déformations. Si l'effort est relâché, le matériau repasse par une nouvelle phase d'élasticité et atteint un point de repos différent : on parle d'hystérésis (figure [I.5](#)). Si la déformation atteint un deuxième seuil appelé *point de rupture*, la structure perd toute ou partie de ses propriétés d'élasticité et de plasticité, voire se fracture en plusieurs éléments.

Un matériau peut également ne pas avoir de comportement plastique avant la rupture : on dit qu'il est *cassant*. Cela ne veut pas dire pour autant qu'il est fragile, mais quand un tel matériau attend le point de rupture, il éclate. Très peu de matériaux sont vraiment cassants (céramiques...) et se déforment généralement peu. A l'opposé, un matériau dit *ductile* réagit de manière élastique puis plastique avant de se fracturer. Les matériaux ductiles ont des comportements de fracture bien plus divers, à cause de l'énergie absorbée pendant la phase de plasticité.

On parle généralement de *défaillance* du matériau quand il atteint la limite d'élasticité ou le point de rupture. Celle-ci dépend du type de charge (statique ou variable) et du type de matériau (cassant ou ductile). Nous nous intéressons d'abord à la défaillance due à des charges statiques, puis à des charges variables.

## c Fatigue

Pour beaucoup de structures déformables, le mouvement désiré peut ne se produire qu'une unique fois et les critères décrit ci-dessous s'avérer suffisants pour décrire le comportement du matériau. Mais dans le cas des objets 1D qui nous intéressent, comme les câbles, leur forme varie constamment sous l'effet des charges extérieures. On parle alors de résistance à un nombre important de cycles, voire à une "vie infinie". Ces chargements répétés imposent des contraintes variables et peuvent résulter en *fatigue*. Une défaillance par la fatigue peut se produire pour des contraintes bien moindres que celles qui entraînent une défaillance statique. On ne peut se permettre qu'un objet tel qu'un câble, souvent constamment sollicité, connaisse une défaillance prématurée. Il est donc nécessaire d'établir des critères pour estimer la fatigue durant la conception. Quelques modèles existent dans ce sens ([Howo1](#)), mais ce phénomène présente un intérêt trop limité en simulation interactive. Il ne sera donc pas repris dans la suite de ce document.

Nous n'avons ici que peu traité le problème de la fracture. Des études plus complètes sont proposées par Anderson ([And95](#)), Nishioka ([Nis97](#)) et Howell ([Howo1](#)).

## d Modèles de déformations plastiques

A l'opposé de l'élasticité, relativement peu d'articles traitent le comportement plastique, qui modélise les déformations irréversibles d'un solide. Terzopoulos *et al.* ([TF88](#); [Ter86](#))

ont introduit le comportement non élastique dans la communauté graphique. Ils ont proposé des modèles physiques pour simuler la viscoélasticité, la plasticité et la fracture pour l'animation. Leur méthode définit des tenseurs pour les énergies de déformations de courbes, surfaces et volumes. Ce tenseur sert de base pour produire un modèle continu à partir des différences finies.

O'Brien *et al.* (OBH02) proposent une méthode peu coûteuse pour traiter la plasticité dans les objets ductiles. Contrairement aux objets cassants, la fracture des objets ductiles passent par une phase de plasticité supplémentaire. Ils déterminent la limite d'élasticité à l'aide du critère de von Mises-Hencky. La déformation plastique est modélisée à l'aide d'un tenseur de déformation plastique, qui correspond à un état d'équilibre que l'on modifie quand une contrainte dépasse le critère de limite d'élasticité. Müller *et al.* (MKN<sup>+</sup>04) reprennent cette modélisation élégante. Nous adopterons une approche semblable dans notre modélisation de la plasticité (section 2, page 80).

## e Modèles de fracture

La fracture est un domaine beaucoup plus abordé que la plasticité, sûrement parce qu'elle est beaucoup plus dure à imiter sans des modélisations physiques. Les papiers qui la traitent sont relativement nombreux.

Terzopoulos *et al.* (TF88) montrent que certains effets de fracture peuvent être reproduits en annulant les coefficients élastiques entre noeuds voisins, quand la distance entre ces noeuds dépasse un seuil. La déchirure de feuilles de papier et de vêtements illustrent leur propos. Norton *et al.* (NTB<sup>+</sup>91) ont présenté une technique à base d'un système de masse-ressort pour animer la fracture d'objets. Ceux-ci cassent quand ils sont soumis à de trop larges déformations. Un seuil de distance est également utilisé pour déterminer pour cette fois quand un ressort doit être supprimé. L'exemple d'une théière se brisant au sol illustre leur proposition. Smith *et al.* (SWB00) proposent une méthode relativement rapide à base de tétraèdres pour modéliser la fracture d'objets cassants. Les points de rupture sont déterminés à partir des forces de contraintes lagrangiennes, plutôt que des seuils de distance. Les vitesses des fragments résultants peuvent être également calculées à partir de ces forces.

Cependant, ces méthodes connaissent deux limitations liées à l'utilisation d'un maillage. La première est l'absence de précision dans la localisation de la fracture. Celle-ci est obligatoirement située entre deux noeuds de la discrétisation spatiale. L'autre limitation concerne la trajectoire de la fracture qui peut être fortement influencée par la construction du maillage de l'objet, d'autant plus si le maillage est régulier. Si le maillage est irrégulier, les artefacts seront plus ou moins masqués, et les fissures suivront de toute façon les bords de facettes prédéfinies. L'effet modélisé doit donc être d'une échelle bien plus importante que la discrétisation. O'Brien *et al.* (OH99; OBH02) tentent donc de remailler dynamiquement le modèle 3D au cours de la simulation pour simuler la fracture d'objets cassants puis ductiles.

Maintenir un maillage conforme est cependant une tâche ardue et coûteuse quand la topologie de l'objet change fréquemment. Finalement, leur modèle donne des résultats très convaincants mais coûteux : de l'ordre d'une dizaine de minutes par seconde de simulation. Müller *et al.* (MMDJ01) approchent par contre le temps réel, en restreignant la résolution cette fois statique lors du traitement de la collision. L'utilisation de l'orientation des tenseurs de contrainte leur permet d'obtenir des fractures qui ne dépendent pas de la discrétisation du maillage. Ils n'ont par conséquent pas à remailler le modèle en cours de simulation. Irving *et al.* (ITFo4) donnent un mécanisme pour contrôler la déformation plastique d'un maillage à éléments finis, ce qui permet de guider un objet déformable vers une forme donnée tout en étant réaliste. Sukumar *et al.* (SMMB00) proposent eux d'enrichir le maillage d'éléments finis avec des méthodes à base de particules au niveau des fissures. Krysk et Belytschko (KB99) modélisent entièrement l'objet à l'aide de particules.

Ces méthodes ont rencontré un vif succès ces dernières années. Elles permettent d'éviter les complexes opérations de remaillage, ainsi que les problèmes de coupe d'éléments de la méthode des éléments finis. Pauly *et al.* (PKA<sup>+</sup>05) ajoutent un aspect multi-résolution qui renforce le réalisme des déformations et de la fracture. Wicke (WSG05) décrit une méthode également à base de particules qui peut reproduire le comportement de plasticité et de fracture sur des objets surfaciques comme des coques. Enfin, d'autres travaux s'intéressent à l'éclatement ou à l'explosion d'objets (MMA99; YOH00).

De nos jours, des modèles permettent de simuler interactivement des déformations visuellement convaincantes de matériaux plus ou moins mous. Ils tiennent compte de la flexion et de la torsion, de l'étirement, et parfois de la fracture, avec une complexité linéaire en temps de calcul. La plausibilité physique n'est cependant présente que pour des déformations à complexité relativement modestes. Un objet déformable 1D peut être simulé comme un objet déformable 3D, avec les méthodes que nous venons de mentionner. Néanmoins, elles nécessitent un nombre important d'éléments pour obtenir des résultats probants et sont par conséquent beaucoup trop coûteuses en termes de calculs pour de la réalité virtuelle. C'est pourquoi de nombreux modèles spécifiques aux objets déformables 1D ont été développés, que nous étudierons dans la section suivante.

## B SIMULATION D'OBJETS DÉFORMABLES 1D

Les objets 1D se présentent sous de nombreuses formes, avec des échelles, des matériaux et des aptitudes à se déformer différentes. Néanmoins, les études mécaniques menées sur un type d'objet 1D se révèlent généralement valables sur un autre. L'expérience montre que certaines communautés scientifiques traitent le même problème, certaines fois de façon très similaire, d'autres avec des approches complémentaires. Mais bien souvent elles s'ignorent : les biologistes travaillent sur l'ADN, les mécaniciens sur des structures composées de poutres, les graphistes sur les cheveux... et pourtant le comportement mécanique de tous les objets 1D peut être étudié avec les mêmes outils ! Une branche de l'élasticité leur est consacrée : il s'agit de la théorie des poutres. Nous verrons également que les motivations diffèrent selon les communautés et par conséquent les objectifs également : les mécaniciens s'attachent à la précision et la validité du comportement simulé tandis que l'animation de cheveux nécessite de la rapidité de calcul ; ces deux besoins sont rarement conciliables et c'est pourtant notre objectif.

Comme nous l'avons vu précédemment, les lois de la mécanique des milieux continus fournissent des équations, généralement aux dérivées partielles (EDP) ou différentielles ordinaires (EDO), liant souvent contraintes, déformations et déplacements. La théorie des poutres nous en donnera d'autres, plus spécifiques. Elles dépendent de grandeurs spatiales et temporelles *continues*. Bien souvent, on ne connaît pas de solution analytique générale de ces équations. Il faut donc les résoudre *numériquement*, c'est-à-dire trouver les valeurs des grandeurs qui les satisfont au mieux. Pour ce faire, il est nécessaire de discrétiser le problème à la fois en espace et en temps. Cette discrétisation constitue ici un enjeu majeur pour modéliser correctement les déformations d'un objet 1D : elle conditionne la précision et la stabilité de la simulation, ainsi que son aspect temps-réel.

Dans cette section, nous présenterons d'abord les grandes lignes de la théorie des poutres, puis nous étudierons successivement les discrétisations spatiales et temporelles des différents modèles 1D déformables de la littérature, pour terminer sur la modélisation des interactions avec leur environnement, ainsi que leur manipulation.

### 1 Théorie des poutres

Issue de la théorie de l'élasticité isotrope, la *théorie des poutres* permet de simplifier les problèmes rencontrés, en assimilant le solide étudié à une poutre, à savoir une structure

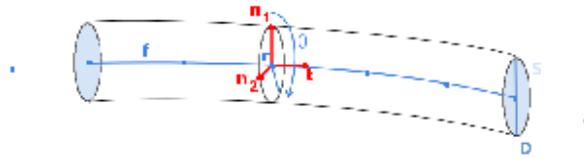


Fig. I.6 – Schéma d'une poutre.

élançée, dont une dimension est grande devant les deux autres. C'est ainsi que l'étude d'une aile d'avion ou la semelle d'un pont de bateau peut être abordée par la théorie des poutres.

Celle-ci s'applique, rappelons-le, dans un domaine d'efforts permettant au matériau, homogène et isotrope, de demeurer linéaire et élastique, ce qui implique, entre autres, qu'après déformation, le solide reprend sa forme initiale.

Dans cette sous-section, nous allons présenter les connaissances et résultats requis pour modéliser des objets 1D déformables à l'aide de la physique. Ils sont nécessaires d'une part pour bien comprendre les modèles étudiés dans la section suivante.

Pour les bases mécaniques en théorie des poutres, nous nous sommes assez largement inspirés de l'excellent traité de Jean Courbon sur la "théorie des poutres" (Cou80c) et du non moins bon cours en ligne de Jean Garrigues sur la "statique des poutres élastiques" (Gar99). Quant à l'historique, nous recommandons notamment les thèses de Chouaïeb (Choo3) et Primault (Prio3). Il n'existe par contre pas d'état de l'art complet sur les objets déformables 1D si ce n'est celui de Ward *et al.* (WBK<sup>+</sup>07), spécifique aux cheveux, que nous avons complété partiellement avec les thèses de Lenoir (Leno4) et (Bero6). Nous avons par conséquent effectué un important travail de recherche en termes de publications et travaux traitant de ce problème.

### a Définitions et terminologie

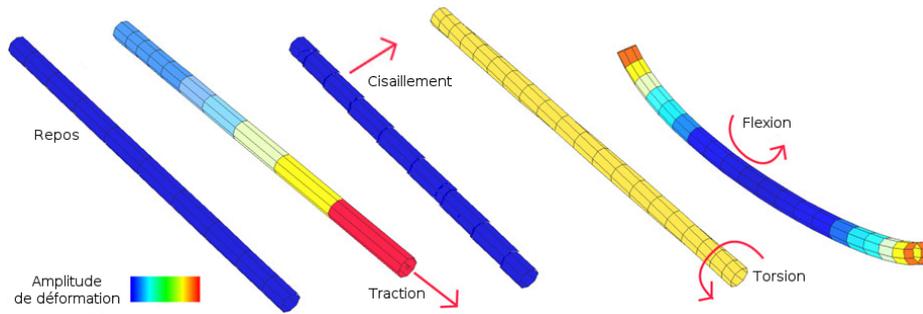
Soit une courbe orientée appelée *ligne moyenne* ou *fibre moyenne*  $f$ . Pour tout point  $G$  de cette fibre moyenne, on définit dans le plan normal une *section*  $S$  de contour fermé et de surface  $A$ , dont le centre de gravité est ce point  $G$ . Le volume engendré par l'ensemble des sections droites de la ligne moyenne définit une *poutre* (figure I.6).

Une poutre gauche est une poutre dont la fibre moyenne est une courbe gauche (non droite). Une poutre plane est une poutre dont la fibre moyenne est une courbe plane. Une poutre droite est une poutre dont la fibre moyenne est un segment de droite.

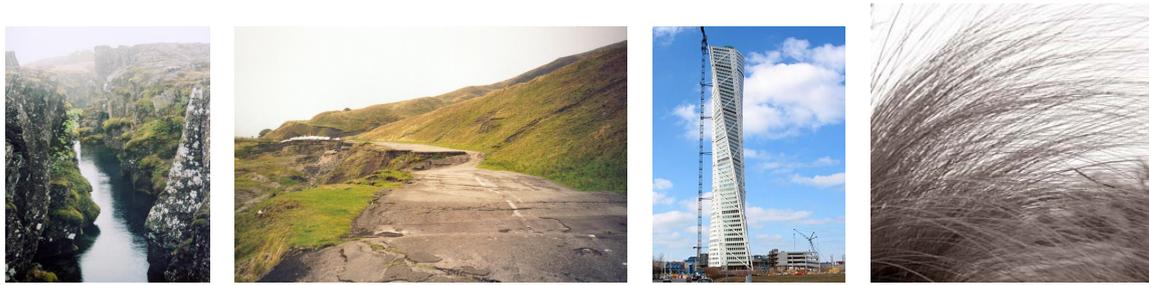
Si l'aire  $A$  est constante, la poutre est alors dite de *section constante*. Mais très souvent, en vue de proportionner les dimensions de la poutre aux efforts qu'elle doit supporter, l'aire  $A$  varie lorsque son centre de gravité décrit la fibre moyenne ; la poutre est alors dite de *section variable*. On appelle *prisme élémentaire* ou *élément de poutre* le volume de la poutre compris entre deux sections droites voisines.

Une poutre peut subir différents types de déformations : la *traction* et la *compression*, lorsque la fibre moyenne s'allonge ou se raccourcit ; le *cisaillement*, quand la section se déplace dans son plan ; la *torsion*, lorsque la section tourne autour de la ligne moyenne ; et la *flexion* quand la fibre moyenne est courbée. Ces différentes déformations sont illustrées à un niveau macroscopique dans la figure I.7.

Nous supposons que les forces et moments extérieurs sont appliqués à la fibre moyenne de la poutre. Une force appliquée en un point d'une section  $S$  de centre de gravité  $G$  peut être remplacée par une force équipollente appliquée en  $G$  et par un couple égal au moment de la force par rapport au point  $G$ .



(a) Les types déformations d'une poutre.



(b) Canal formé par extension des plaques en l'Irlande. (c) Cisaillement de la route A6187 au Royaume-Uni. (d) Le "Turning Torso" à Copenhague. (e) Herbe fléchie par le vent.

**Fig. I.7** – Les différents types de déformation d'une poutre, représentés de manière schématique et réelle.

## b Historique

L'étude des objets monodimensionnels, appelées poutres ou tiges selon les études, a une longue histoire.

Il est communément reconnu que c'est Galilée, au XVI<sup>ème</sup> siècle, qui a essayé le premier de développer une théorie des poutres, mais des études récentes prétendent que Léonard de Vinci est le premier à avoir fait des observations cruciales. Il a cependant manqué la loi de Hooke et de calcul pour compléter la théorie, tandis que Galilée a émis une mauvaise hypothèse : il pensait que pour un chargement transverse, la section d'une poutre développait une contrainte uniforme (Balo6). Au début du XVIII<sup>ème</sup> siècle, Parent a été le premier à obtenir la distribution correcte des contraintes et à établir une relation entre la contrainte et le moment de flexion. Vers 1750, Daniel Bernoulli et Leonhard Euler, travaillèrent sur la flexion d'une tige dans le plan, l'*elastica*, une tige étant une poutre mince, inextensible et élastique. En 1859, Gustav Robert Kirchhoff a généralisé leurs travaux dans le plan, en passant à des tiges en trois dimensions (Kir59). Kirchhoff a montré qu'une tige initialement droite, isotrope et uniforme et dont on néglige le cisaillement, admet des solutions hélicoïdales sous l'action de forces et de moments appliqués à ses extrémités. La théorie de Kirchhoff est fondée sur l'hypothèse que les contraintes dépendent linéairement de la courbure et de la torsion géométrique.

Mais c'est au début du siècle dernier, que les frères Cosserat ont proposé le premier modèle complet spécifique aux objets monodimensionnels, qui décrit, en plus de la flexion et de la torsion, les déformations de cisaillement et d'extension (CC09). Ce modèle est la base de nombreuses théories plus complexes de tiges, mais il sera pourtant ignoré par la communauté mécanique pendant plus de cinquante ans.

Pour un historique plus détaillé sur les poutres, nous invitons le lecteur à consulter (Balo6, Ant95, Choo3, Prio3).

### c Théories et hypothèses usuelles

Les différentes théories issues de l'étude des poutres tirent partie d'un certain nombre d'hypothèses pour simplifier les équations données par la théorie de l'élasticité. Ces hypothèses portent généralement sur le comportement de la section de la poutre et les déformations à prendre en compte parmi l'élongation, le cisaillement, la torsion et la flexion.

Deux grandes voies théoriques permettent d'aboutir aux équations liant efforts et déplacements d'une poutre. La première reprend la démarche habituelle de la mécanique des milieux continus en procédant aux approximations dues à la faiblesse des rapports des deux dimensions transverses sur la troisième (Ant72). Dans ce contexte, les équations d'équilibre sont déduites par intégration des équations tridimensionnelles sur les deux dimensions transverses. De plus, il est nécessaire d'intégrer dans l'épaisseur les lois constitutives tridimensionnelles impliquant un jeu de tenseurs tridimensionnels tels que le tenseur des déformations de Green-Lagrange et celui des contraintes de Piola-Kirchoff pour résoudre le problème en grandes transformations.

La seconde voie, au contraire de la première, considère d'emblée la poutre comme un milieu monodimensionnel réalisé par une succession continue de micro-solides rigides. C'est l'approche initiée par les frères Cosserat (CC09). Une tige est modélisée par une courbe géométrique 3D et des repères orthonormés associés à chacun des points de la courbe. Chaque élément possède donc 6 degrés de liberté, 3 en translation et 3 en rotation.

**Théorie des poutres de Timoshenko** Si les deux approches conduisent essentiellement aux mêmes équations d'équilibre, les lois de constitutions diffèrent et les équations de type Cosserat, généralement plus simples, ne peuvent pas forcément être déduites des équations tridimensionnelles. Ces dernières peuvent par contre prendre en compte une certaine déformation des sections, alors qu'une tige de Cosserat impose des sections planes (hypothèse de Navier-Bernoulli) et interdit une quelconque distorsion en leur sein.

Si aucune autre hypothèse n'est imposée, la tige de Cosserat est dite poutre de Timoshenko ou de Reissner (Rei73)<sup>1</sup> Cette théorie a été développée par de nombreux auteurs : Simo et Vu-Quoc (Sim85; Sim88; VQDE96) et en même temps Cardona et Géradin (CG88), puis Ibrahimbegovic (Ibr95), ont étendu sa résolution notamment à l'aspect dynamique, aux structures multicouches, à l'approche dite géométriquement exacte...

**Hypothèse de Bernoulli et théorie des poutres de Rayleigh** Dans le cas où la poutre est suffisamment fine, les sections peuvent être considérées orthogonales à la ligne moyenne au cours de la déformation. C'est l'hypothèse de Bernoulli ou hypothèse de Kirchhoff<sup>2</sup>, qui revient à négliger le cisaillement. Le modèle qui en découle est la *poutre de Rayleigh*. Les tiges de Kirchhoff (Kir59) considèrent en plus que la poutre est inextensible. Les équations correspondantes sont numériquement raides et non linéaires (HKS98).

**Hypothèse et théorie des poutres d'Euler-Bernoulli** Si l'on suppose en plus que l'énergie cinétique de rotation des sections est négligeable, le modèle est dit *poutre d'Euler-Bernoulli*. Cette hypothèse n'est valable que dans le domaine des faibles fréquences. Elle n'est donc pas adaptée aux vibrations (cordes d'instruments de musique par exemple) mais convient à la manipulation. Une conséquence de cette hypothèse est que toute rotation d'une section se propage immédiatement le long de la poutre. De plus une contrainte est nécessaire pour définir une configuration, sinon la solution est indéfinie.

<sup>1</sup>L'hypothèse de non-déformation de la section est généralement associée à Timoshenko pour les poutres, Mindlin et Reissner pour les plaques et Naghdi pour les coques

<sup>2</sup>L'hypothèse d'orthogonalité de la section par rapport à la ligne moyenne est elle associée à Bernoulli pour les poutres, Kirchhoff pour les plaques et Love pour les coques

**Hypothèses pratiques** Dans la pratique, on peut considérer un objet comme une poutre si les conditions suivantes sont respectées.

*Les dimensions transversales d'une poutre doivent être petites par rapport à sa longueur . Pour une poutre droite, le rapport de la hauteur de la section à la longueur de la poutre est en général compris entre 1/5 et 1/30, les valeurs les plus courantes étant comprises entre 1/10 et 1/20. Pour les poutres courbes ou arcs, ce rapport peut être notablement plus faible, et compris entre 1/50 et 1/100. Par contre, lorsque les forces appliquées à la poutre sont parallèles, la largeur de la section mesurée normalement à la direction des forces peut être plus élevée.*

*Le rayon de courbure de la fibre moyenne doit être grand par rapport à la dimension transversale de la poutre, mesurée dans le plan osculateur à la fibre moyenne. Si la fibre moyenne est plane, son rayon de courbure doit être supérieur à cinq fois la hauteur de la section mesurée dans le plan de la fibre moyenne.*

*Lorsque la poutre est de section variable, la variation de la section doit être lente et progressive.*  
Appliquons maintenant ces hypothèses pour évaluer les déformations d'une poutre.

## 2 Déformations d'une poutre

Dans cette sous-section, nous allons déterminer les relations entre les déformations de traction-compression, de torsion, de flexion et les forces et moments correspondants. Ils nous permettront de déterminer ensuite les énergies de déformations que nous utiliserons dans les équations d'équilibre.

### a Modélisation des efforts intérieurs

**Principe I.1** Le principe de Saint-Venant ([Cou80c](#); [Gar99](#)) s'énonce de la façon suivante : la contrainte en un point éloigné des points d'application d'un système de forces ne dépend que de la résultante générale et du moment résultant de ce système de forces.

La solution de Saint-Venant n'est valable que loin de la section de l'application des charges. On admet en pratique qu'elle est vraie à partir de quelques diamètres ([Gar99](#)).

Le principe de Saint-Venant permet donc de travailler avec des forces ou charges  $R$  et des couples ou moments  $M$  plutôt que des contraintes  $\bar{\sigma}$ , outils moins simples à manipuler.

**Définition I.1** Pour modéliser les efforts intérieurs d'une poutre, c'est-à-dire les efforts d'une partie de la poutre sur une autre au travers d'une section  $S$ , plus exactement sur un point de la fibre moyenne, on les exprime sous forme d'un torseur  $\mathcal{T}_S$ . Ce torseur est appelé torseur de cohésion :

$$\mathcal{T}_S = (\mathbf{R} \quad \mathbf{M})_S \quad (\text{I.12})$$

La charge  $R$  est décomposable en deux forces, l'une  $N$  normale à la section  $S$ , et l'autre  $T$  contenue en son plan. L'effort normal  $N$  engendre une traction ou une compression de la poutre, dans le sens de la fibre moyenne, qui s'allonge ou rétrécit. Par convention une compression est positive. L'effort tranchant  $T$  donne un effet de cisaillement : l'angle entre la section et la fibre moyenne évolue.

De même, le couple  $M$  est décomposable en deux moments, l'un  $C$  normal à la section  $S$  et l'autre  $M$  contenue en son plan. Le couple de torsion  $C$  engendre une torsion, rotation de la section autour de la fibre moyenne. Le moment fléchissant  $M$  impose une courbure de la fibre moyenne.

Le torseur de cohésion peut donc également s'exprimer en fonction de  $N, T, C$  et  $M$  :

$$\mathcal{T}_S = \begin{pmatrix} N & C \\ T & M \end{pmatrix}_S \quad (\text{I.13})$$

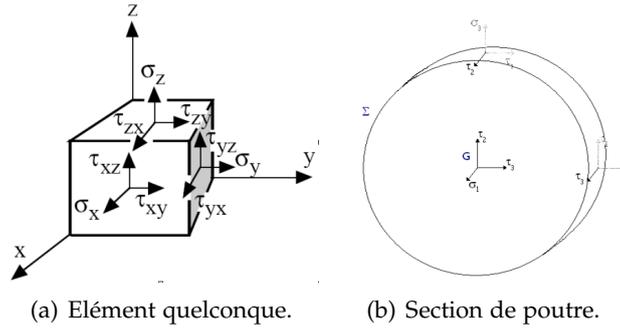


Fig. I.8 – Modélisation des contraintes.

L'effort tranchant  $T$  et le moment fléchissant  $M$  peuvent être projetés sur les axes  $y'$  et  $z'$  du repère local ; le torseur de cohésion s'exprime alors de la façon suivante :

$$\mathcal{T}_{(x',y',z')} = \begin{pmatrix} N & C \\ T_{y'} & M_{y'} \\ T_{z'} & M_{z'} \end{pmatrix}_{(x',y',z')} \quad (\text{I.14})$$

### b Relation entre la contrainte et le torseur de cohésion

Le tenseur de contraintes  $\bar{\sigma}$  sur un élément quelconque s'exprime de la façon suivante (figure I.8(a)) :

$$\bar{\sigma} = \begin{pmatrix} \sigma_{x'} & \tau_{xy'} & \tau_{xz'} \\ \tau_{yx'} & \sigma_{y'} & \tau_{yz'} \\ \tau_{zx'} & \tau_{zy'} & \sigma_{z'} \end{pmatrix}_{(x',y',z')} \quad (\text{I.15})$$

A l'équilibre,  $\tau_{xy'} = \tau_{y'x} = \tau_3$ ,  $\tau_{xz'} = \tau_{z'x} = \tau_2$  et  $\tau_{yz'} = \tau_{z'y} = \tau_1$  ; le tenseur de contrainte  $\sigma$  est donc symétrique :

$$\bar{\sigma} = \begin{pmatrix} \sigma_1 & \tau_3 & \tau_2 \\ \tau_3 & \sigma_2 & \tau_1 \\ \tau_2 & \tau_1 & \sigma_3 \end{pmatrix}_{(x',y',z')} \quad (\text{I.16})$$

De plus, pour une section de poutre (figure I.8(b)), on considère les composantes  $\sigma_2$ ,  $\sigma_3$  et  $\tau_1$  comme négligeables : cela revient à supposer que la contrainte de cisaillement s'exerce parallèlement à la fibre moyenne : c'est le principe de Saint-Venant .

Le tenseur des contraintes devient donc :

$$\bar{\sigma} = \begin{pmatrix} \sigma_1 & \tau_3 & \tau_2 \\ \tau_3 & 0 & 0 \\ \tau_2 & 0 & 0 \end{pmatrix}_{(x',y',z')} \quad (\text{I.17})$$

A l'aide des équations d'équilibre et de Beltrami, cette dernière hypothèse permet d'obtenir l'expression des composantes du torseur de cohésion (Gar99) :

$$N = \oint_S \sigma_1 dS \quad (\text{I.18})$$

$$T_{y'} = \oint_S \tau_3 dS \quad (\text{I.19})$$

$$T_{z'} = \oint_S \tau_2 dS \quad (\text{I.20})$$

$$C = \oint_S (y' \tau_2 - z' \tau_3) dS \quad (\text{I.21})$$

$$M_{y'} = \oint_S \sigma_1 z' dS \quad (\text{I.22})$$

$$M_{z'} = - \oint_S \sigma_1 y' dS \quad (\text{I.23})$$

Dans la suite de cette section, nous allons calculer ces intégrales pour les exprimer en fonction des déformations de la poutre. Nous nous plaçons dans le cadre de l'élasticité linéaire. Nous pouvons donc appliquer le principe de superposition, qui dit que l'on peut considérer individuellement chaque type de déformation puis sommer les contributions. Pour chaque type de déformation, à savoir la flexion, la torsion, nous exprimerons les composantes du torseur de cohésion en fonctions de la géométrie de la poutre, puis les énergies de déformation internes correspondantes. Nous expliquerons pourquoi le cisaillement n'affecte pas le déplacement de la ligne moyenne.

### c Effort de traction-compression

Réduisons le système des forces extérieures au seul effort normal  $N$ . Nous sommes en traction/compression pure. Dans le cadre de l'élasticité linéaire, la relation entre contrainte et déformation est linéaire. La contrainte normale  $\sigma_1$  peut donc s'exprimer de la sorte :

$$\sigma_1 = a + by' + cz' \quad (\text{I.24})$$

A l'aide des équations I.18, I.22 et I.23 :

$$\begin{aligned} \oint_S \sigma_1 dS &= N \\ \oint_S \sigma_1 y' dS &= 0 \\ \oint_S \sigma_1 z' dS &= 0 \end{aligned}$$

nous trouvons  $a = \frac{N}{A}$ , avec  $A$  l'aire de la section  $S$ . L'effort normal  $N$  est donc donnée par la formule :

$$N = A\sigma_1 \quad (\text{I.25})$$

Le calcul du déplacement d'un point de la poutre à l'aide de cette équation, permet d'exprimer l'allongement relatif  $\epsilon$  en fonction de l'effort normal :

$$\epsilon = \frac{N}{EA} \quad (\text{I.26})$$

L'expression de la densité d'énergie potentielle de déformation en traction/compression  $dW = W_s ds$  est donnée par :

$$dW = \sigma_1^2 2EdV = \frac{N^2}{2EA} ds \quad (\text{I.27})$$

d'où

$$W_s = \frac{N^2}{2EA} = \frac{EA}{2} \epsilon^2 \quad (\text{I.28})$$

Notons qu'une traction ou compression est accompagnée d'un rétrécissement ou d'un agrandissement du diamètre  $D$  :

$$\frac{\Delta D}{D} = -\nu \frac{\Delta L}{L} \quad (\text{I.29})$$

La quantité  $\frac{\Delta D}{D}$  est appelée *déformation transverse*.

#### d Moment fléchissant

Soumettons la section  $S$  de la poutre à une flexion déviée, c'est-à-dire que le système des forces extérieures à la section se réduit à un couple autour d'un axe appartenant au plan de la section  $S$  et passant par  $G$ . Si le couple est dirigé suivant l'un des axes centraux  $Gy'$  ou  $Gz'$ , on parle de flexion pure.

Dans le cadre de l'élasticité linéaire, la contrainte normale  $\sigma_1$  peut s'exprimer de la sorte :

$$\sigma_1 = a + by' + cz' \quad (\text{I.30})$$

A nouveau à l'aide des équations I.18, I.22 et I.23 :

$$\begin{aligned} \oint_S \sigma_1 dS &= 0 \\ \oint_S \sigma_1 y' dS &= -M_{z'} \\ \oint_S \sigma_1 z' dS &= M_{y'} \end{aligned}$$

nous obtenons  $a = 0$ ,  $b = -\frac{M_{z'}}{I_{z'}}$ ,  $c = \frac{M_{y'}}{I_{y'}}$ , avec  $I_{y'}$  et  $I_{z'}$  désignant les moments d'inertie de la section par rapport aux axes  $Gy'$  et  $Gz'$  :  $I_{y'} = \oint_S z'^2 dS$  et  $I_{z'} = \oint_S y'^2 dS$ . Les moments d'inertie de quelques types de section sont donnés dans l'annexe D.

La contrainte normale est donc égale à

$$\sigma_1 = \frac{M_{y'} z'}{I_{y'}} - \frac{M_{z'} y'}{I_{z'}} \quad (\text{I.31})$$

Dans le cadre d'une flexion pure, on peut exprimer les projections  $M_{y'}$  et  $M_{z'}$  sur les axes  $Gz'$  et  $Gy'$  du moment fléchissant  $M$ . le moment fléchissant  $M_{y'}$  pour une flexion pure par rapport à l'axe  $y'$  s'exprime :

$$M_{y'} = \frac{I_{y'}}{z} \sigma_1 \quad (\text{I.32})$$

De même, le moment fléchissant  $M_{z'}$  pour une flexion pure par rapport à l'axe  $z'$  est donc donné par la formule :

$$M_{z'} = \frac{I_{z'}}{y} \sigma_1 \quad (\text{I.33})$$

On peut retrouver la formule I.31 en superposant ces deux flexions pures.

En calculant le déplacement d'un point de la poutre à l'aide de cette équation, on déduit une conséquence importante de la flexion sur la courbure  $k$  :

$$k = \frac{M}{EI} \quad (\text{I.34})$$

L'expression de la densité d'énergie potentielle de déformation en flexion  $dW = W_s ds$  est donnée par :

$$dW = \sigma_1^2 2EdV = \frac{M^2}{2EI^2} ds \oint_S y'^2 dS \quad (\text{I.35})$$

d'où

$$W_s = \frac{M^2}{2EI} = \frac{EI}{2} k^2 \quad (\text{I.36})$$

En décomposant le moment de flexion  $M$ , on obtient à l'aide de l'équation I.31 la formulation suivante :

$$W_s = \frac{1}{2} \left( \frac{M_{y'}^2}{EI_{y'}} + \frac{M_{z'}^2}{EI_{z'}} \right) \quad (\text{I.37})$$

### e Couple de torsion

Considérons un élément de poutre  $dx'$  compris entre deux sections voisines. Pour une poutre droite, le déplacement relatif entre les deux sections est une rotation  $\theta$  autour de  $Gx'$ , égale à  $\frac{d\theta}{dx'}$ <sup>3</sup>. Un point de la section se déforme donc de

$$r \frac{d\theta}{dx'} \quad (\text{I.38})$$

La loi de Hooke permet d'établir la relation entre la contrainte et la déformation :

$$\tau = -Gr \frac{d\theta}{dx'} \quad (\text{I.39})$$

où  $G$  est le module d'élasticité transversale, où module de cisaillement. Le couple de torsion est la somme de toutes les contraintes sur la section : il a donc pour valeur

$$C = \oint_S \tau dS \quad (\text{I.40})$$

Pour une section circulaire, cela donne :

$$C = \int_0^R (r\tau) 2\pi r dr = -G \frac{d\theta}{dx'} \int_0^R r^2 2\pi r dr \quad (\text{I.41})$$

En désignant par  $J = \frac{\pi}{2} R^4$  le *module de rigidité à la torsion*, nous obtenons les formules :

$$\tau = \frac{Cr}{J} \quad (\text{I.42})$$

$$\frac{d\theta}{dx'} = -\frac{C}{GJ} \quad (\text{I.43})$$

Les modules de rigidité à la torsion de quelques types de section sont donnés à l'annexe D. La théorie élémentaire de la torsion consiste à admettre que  $J$  est égal au moment d'inertie polaire, ce qui conduit à des résultats complètement faux pour des sections non circulaires. Les raisons sont expliquées dans (Cou80c).

L'expression de la densité d'énergie potentielle de déformation en torsion  $dW = W_s ds$  est donnée par :

$$dW = \sigma_1^2 2G dV = \frac{C^2}{2GJ} ds \quad (\text{I.44})$$

d'où

$$W_s = \frac{C^2}{2GJ} = \frac{GJ}{2} \frac{d\theta}{dx'} \quad (\text{I.45})$$

### f Effort tranchant

Un *effort tranchant*, s'il peut provoquer un mouvement de rotation de torsion, ne provoque ni allongement de la ligne moyenne, ni variation de sa courbure. Néanmoins, Garrigues (Gar99) fait remarquer que si l'effort tranchant n'est pas nul, alors la variation du moment de flexion n'est pas nulle. Il y a donc toujours un moment de flexion qui s'ajoute à un effort tranchant. Les déplacements transversaux de la ligne moyenne sont induits par la flexion et non pas par l'effort tranchant.

Dans un grand nombre de traités de théorie des poutres, on suppose que deux sections voisines soumises à un effort tranchant glissent l'une par rapport à l'autre parallèlement à leur plan (Cou80c), mais le *problème de Saint Venant* montre que cette hypothèse est

<sup>3</sup>Nous verrons plus loin (section d, page 67) que pour une poutre gauche il faut ajouter la torsion géométrique.

incorrecte. La plupart du temps, on se rattrape en disant que les déplacements dus à l'effort tranchant sont négligeables devant ceux dus au moment de flexion qui lui est associé.

En fait, les seuls déplacements induits par l'effort tranchant sont le gauchissement de la section. C'est pour cela que l'hypothèse de Kirchhoff revient à négliger le cisaillement.

Ces quelques notions de la théorie des poutres nous permettent de cadrer les différents modèles de déformations d'objets 1D que nous allons maintenant étudier, en les classant selon la finesse de leur modélisation, c'est-à-dire la discrétisation spatiale.

### 3 Discrétisation spatiale

La discrétisation spatiale peut se faire à plusieurs niveaux : en amont, au niveau de la modélisation du système mécanique et en aval, au niveau du système d'équations à résoudre pour obtenir le mouvement.

La discrétisation en amont favorise des modèles plus simples et plus rapides à résoudre mais les éloigne d'autant de la réalité. La répartition de la masse et le type de déformation en est tributaire. En général, le type de modélisation spatiale est dicté par la nature de l'objet à simuler, mais peut l'être aussi par la puissance de calcul, par l'architecture disponible ou par analogie visuelle. Par exemple, les systèmes à particules tirent largement profit des performances des cartes graphiques actuelles, par rapport aux processeurs classiques (KSW04; Osh07).

La discrétisation en aval, elle, est nécessaire, afin de pouvoir résoudre numériquement les équations obtenues à partir du modèle défini. La méthode utilisée déterminera la précision spatiale des résultats obtenus, mais sera d'autant plus précise que coûteuse à calculer.

Nous avons classé les différents modes de discrétisation spatiale utilisés pour les objets déformables 1D en quatre catégories : les approches *ponctuelles* ou *particulaires*, qui proposent une discrétisation de la matière en un ensemble de points matériels, éventuellement reliés par des ressorts ; ensuite les approches *chaînées*, qui considèrent la matière comme une succession d'éléments rigides ou déformables interfacés par des liaisons mobiles ou fixes ; enfin les approches *continues*, qui partent directement des équations de la mécanique élastique des milieux continus ou de théories dédiées aux objets poutres.

#### a Approches particulières

L'utilisation de particules massiques liées par des ressorts est la démarche la plus intuitive pour simuler les déformations d'un objet. A un niveau macroscopique, les propriétés d'un ressort sont les mêmes qu'un matériau élastique au niveau microscopique. Comme les systèmes à base de ressorts donnent des équations simples, faciles et surtout rapides à résoudre, ils ont remporté - et remportent encore - un vif succès dans la communauté graphique.

Les premières méthodes de modélisation d'objets 1D déformables à partir de ressorts ont été proposées par Thingvold *et al.* (TC90) pour du modelage 3D physique, ainsi que par Rosenblum *et al.* (RCT91) pour l'animation de cheveux, Miller *et al.* (Mil91) pour la simulation de serpents et de vers-de-terre et Provot (Pro95). Dans ces approches, l'objet 1D est représenté par une suite de particules connectées par des ressorts longitudinaux, ainsi que des ressorts dénommés charnières. Ceux-ci sont non pas définis à partir d'un angle, mais de la position de plusieurs particules. Le degré de liberté en translation correspond à une élongation, tandis que celui en rotation donne une certaine résistance à la flexion. Ces méthodes se heurtent à des instabilités numériques dès que la raideur est un peu élevée. Plutôt que d'utiliser des ressorts en translation pour modéliser la flexion, Looock *et al.* (LS01) proposent de les remplacer par des ressorts angulaires qui donnent des forces d'opposition plus convaincantes (figure I.9). L'ajout d'un amortissement en parallèle

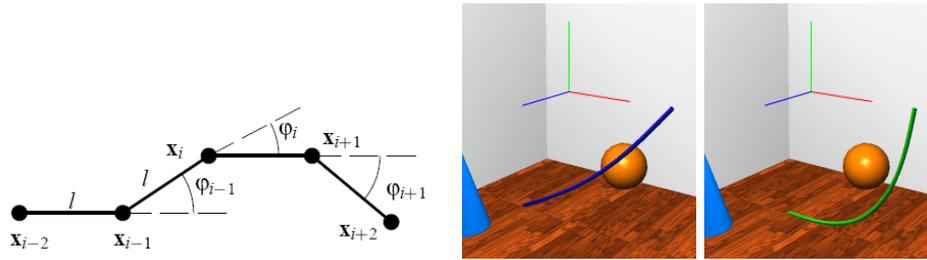
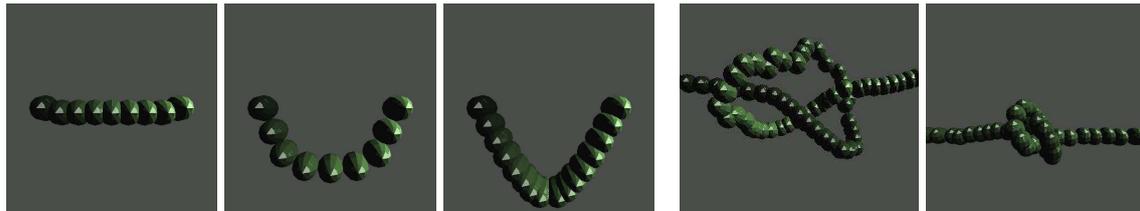


Fig. I.9 – Utilisation de ressorts angulaires pour la flexion, (LS01).



(a) Mauvaise répartition de la masse sur la spline,

(b) Réalisation de noeuds,

Fig. I.10 – Modèle de Phillips *et al.*, (PLK02).

des ressorts longitudinaux permet d'obtenir des simulations plus stables et surtout des performances de l'ordre de la milliseconde pour une dizaine de particules.

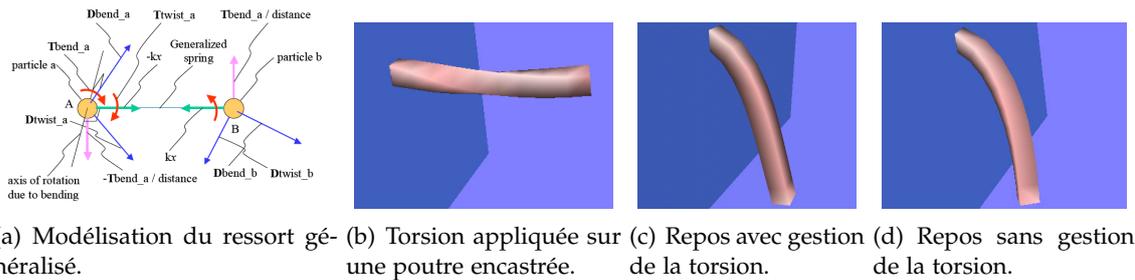
Plante *et al.* (PCP01) simulent des mèches bouclées, uniquement à l'aide de ressorts en extension. Des forces supplémentaires sont introduites pour donner une certaine résistance à la flexion, dépendant de l'épaisseur de la mèche. Comme la différence de longueur entre une mèche fort bouclée et une mèche lisse est importante, les ressorts sont autorisés à être relativement mous, et facilitent une résolution stable.

Witkin *et al.* (WW90) posent les bases d'une méthode rapide pour la simulation physique d'objets déformables, dans le cadre des petits déplacements. Ils présentent l'utilisation conjointe des équations de Lagrange, pour gérer l'énergie cinétique de l'objet, et de celle de ressorts pour gérer les déformations. Puisque les déplacements sont linéaires, les matrices du système sont constantes et peuvent être inversées une fois pour toutes au début de la simulation. L'introduction des contraintes lagrangiennes permet de joindre plusieurs objets, et de les déplacer selon des trajectoires établies. Elles sont gérées de manière stable à l'aide du schéma de Baumgarte. Force est de constater que les auteurs posent toutes les premières briques nécessaires à l'élaboration du modèle spline, tel que Rémion *et al.* (RNG99; Nou99) l'ont défini (voir section c, page 39).

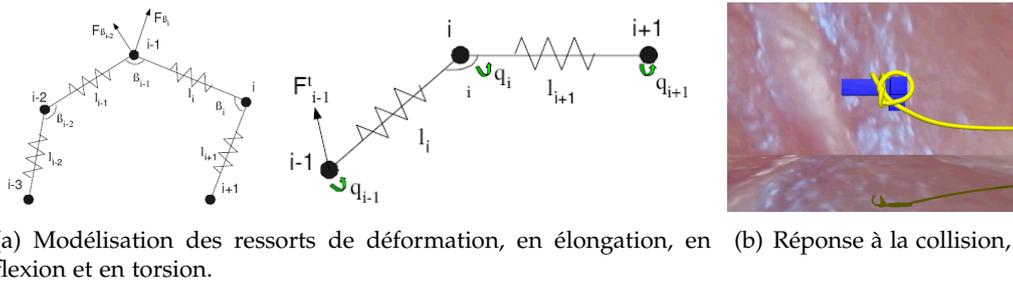
Modèle géré dynamiquement à partir de cette méthode, la corde de Phillips *et al.* (PLK02) est construite comme une spline dont les points de contrôle sont liés par des ressorts longitudinaux. La spline n'est ici qu'un habillage géométrique : elle n'intervient pas dans la mécanique, contrairement aux approches que nous verrons plus loin (section c, page 39). La description des déformations est cependant peu décrite. L'intérêt de ce papier réside dans la proposition d'une première méthode de multi-résolution pour un objet 1D particulière. La nouvelle répartition des masses, qui consiste à prendre la moitié de la masse de chaque voisin est cependant erronée (figure I.10(a)) et conduit à des oscillations et des discontinuités énergétiques. Les auteurs les atténuent en rajoutant de l'amortissement. La réponse à la collision à base de pénalité leur permet néanmoins d'obtenir la formation de noeuds (figure I.10(b)).

Notons que dans le cadre de la simulation de cheveux, Bertails *et al.* (BKCN03) se sont également intéressés à la multi-résolution de systèmes masse-ressort 1D. Celle-ci n'intervient cependant pas au sein d'un élément 1D, mais au niveau de la séparation et de la fusion des branches d'un arbre représentant un groupe de cheveux.

Très peu de modèles exclusivement à base de particules gèrent la torsion, pourtant



**Fig. I.11** – Modèle de ressort gérant les déformation d'élongation, de flexion et de torsion, (JLo4).

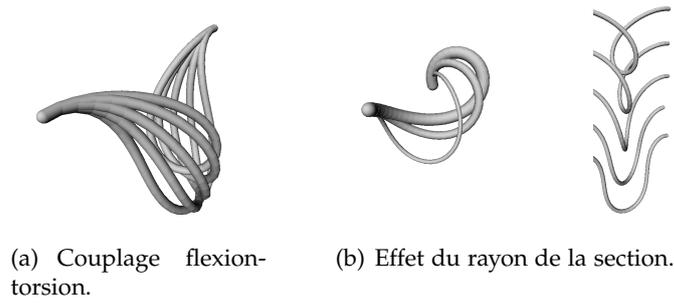


**Fig. I.12** – Modèle de fil chirurgical à base de ressorts, (WBD<sup>+</sup>05).

nécessaire à une représentation correcte des déformations. Jeong *et al.* (JLo4) ont proposé un *ressort généralisé* qui peut gérer non seulement la propriété classique d'un ressort, à savoir la traction-compression, mais aussi un moment d'opposition en flexion et un autre en torsion. Le moment de flexion est basé sur la différence d'angle entre les vecteurs de flexion des deux extrémités, tandis que le couple de torsion s'obtient en projetant les vecteurs de torsion sur le plan normal à l'axe du ressort (figure I.11). Le comportement d'opposition à la torsion est validé par la torsion à 180° d'une barre encastree et soumise à la gravité. En l'absence de gestion de la torsion, le ressort reste partiellement déformé au repos. Ce modèle manque néanmoins de validation au niveau de la précision. De plus, on ne connaît ni le temps de calcul, ni la stabilité du modèle avec une chaîne de ces ressorts.

Dans le cadre de la simulation d'un fil de suture, Wang *et al.* (WBD<sup>+</sup>05; WBVB05) proposent un modèle qui contient un ressort pour chaque type de déformation (figure I.12(a)). Plutôt que de faire uniquement le bilan des forces, ils effectuent également le bilan des moments, à laquelle est associée une variable d'angle dont la variation correspond à la torsion. Le système est résolu avec une intégration Euler explicite. Au vu des résultats, la stabilité ne semble pas être au rendez-vous, puisque le fil évolue dans un milieu très visqueux et qu'il ne subit pas ou très peu la gravité.

Dans le cadre de l'assemblage virtuel de pièces, conscients de l'importance de la torsion dans les déformations Gregoire *et al.* (GS06; GS07) proposent une modélisation de câble qui l'incorpore. Pour y parvenir, ils adaptent le modèle de Cosserat à un système masse-ressort. Le modèle de Cosserat (CC09) représente une tige par une ligne de micro-solides qui sont définis par une position et orientés par des vecteurs directeurs. La flexion et la torsion étant directement évaluées à partir des directeurs et non pas des positions, il n'y a pas d'approximation de ces grandeurs dans le calcul des déformations. On se place donc dans le cadre des grands déplacements. Les auteurs ont choisi de décrire l'orientation à l'aide de quaternions unitaires. Les quaternions sont équivalents aux matrices de rotation, mais ils n'ont que 4 coordonnées et 1 contrainte (norme unitaire), contre 9 composantes et 6 contraintes (vecteurs de base unitaires et orthogonaux deux à deux) pour les matrices de rotation. Un élément du câble est donc associé à une position (3 degrés de liberté) et à un quaternion (4 degrés de liberté). A cause de la difficulté à contraindre une tige entre ses extrémités, que nous évoquerons plus loin avec les travaux de Pai (Pai02) et Bertails (Bero6), les auteurs ont opté pour un système masse-ressort. La tige est donc



**Fig. I.13** – Comportements pris en compte dans le modèle de Grégoire *et al.*, (GS06).

modélisée par une séquence de masses ponctuelles situées sur la ligne moyenne, qui sont reliées par deux types de ressorts : des ressorts longitudinaux gèrent la conservation de la longueur et des ressorts angulaires la flexion. La torsion, considérée comme la somme de la torsion de Frenet  $\tau$  et la torsion matérielle pure  $\frac{d\theta}{ds}$ , est calculée à partir de la variation du quaternion associé **quat**. Pour s'assurer ensuite que la position et l'orientation sont correctement couplées, Grégoire *et al.* introduisent une énergie de cohérence, qui lie le vecteur tangent  $\mathbf{u}_i$  calculé à partir des positions  $r$  des particules et la direction initiale  $\mathbf{u}_{init,i}$ , modifiée par le quaternion d'orientation **quat** :

$$E_{coh,i} = k_{coh} \|(0, \mathbf{u}_i) - \mathbf{quat}_i \cdot (0, \mathbf{d}_{init,i}) \cdot \overline{\mathbf{quat}_i}\|^2 \quad (\text{I.46})$$

avec  $u_i$  le vecteur tangent du  $i^{\text{ème}}$  point, correspondant à l'orientation du quaternion *quat*. La contrainte de norme du quaternion est réajustée après chaque pas de simulation. Les résultats sont assez convaincants : le modèle permet de reproduire des comportements complexes, tels que le couplage flexion-torsion, ou l'effet du rayon de la section sur l'amplitude de la déformation (figure I.13). Ce modèle reste à notre connaissance le plus pertinent dans la catégorie des masses-ressorts. Néanmoins, le propriété d'oscillation intrinsèque aux ressorts se révèle être finalement gênante pour obtenir de bons résultats : la stabilité d'un tel système ne résiste pas aux fortes raideurs inhérentes aux câbles, particulièrement au niveau de la tension. Une solution consiste alors à utiliser des chaînes plus ou moins articulées d'éléments rigides ou souples, que nous allons présenter dans la sous-section suivante.

## b Approches chaînées

L'*algorithme de chaîne* (Nah91; How01) est appelé ainsi car il requiert la discrétisation de l'objet en éléments de type poutre et analyse chaque élément à la suite. Un élément est traité comme une poutre encastree au bout de l'élément précédent. L'inconvénient majeur qui en découle est la nécessité de fixer une condition initiale au minimum à l'une des extrémités, généralement sous la forme d'un encastrement. L'algorithme se déroule en deux étapes. Il consiste à effectuer le bilan des forces du côté libre de la chaîne, et à remonter le bilan jusqu'à la partie fixe. Le même chemin est alors parcouru en sens inverse pour repositionner les maillons de la chaîne, les uns par rapport aux autres. Nous classons les approches chaînées en deux catégories : celles qui sont constituées d'éléments rigides interfacés par des liaisons mobiles ; et celles qui juxtaposent bout-à-bout des éléments déformables. Les unes s'apparentent à l'étude dynamique directe des mécanismes poly-articulés en robotique, tandis que les autres sont issues de modèles spécifiques 1D, tels que les modèles de tige de Kirchhoff et de Cosserat.

**Dynamique directe** Hadap et Magnenat-Thalmann (HMT00; HMT01) mettent en évidence que les méthodes à base de masse-ressort, que nous avons vues précédemment, ne sont pas adaptées à la simulation d'objets 1D déformables, et en particulier le cheveu. En

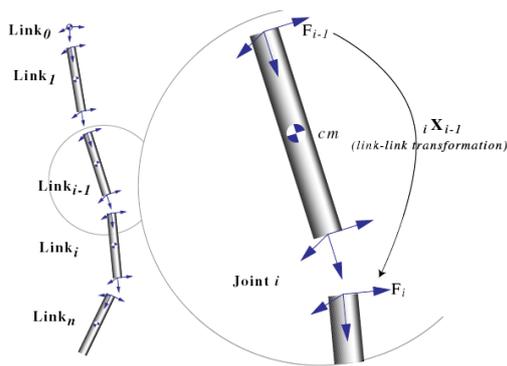


Fig. I.14 – Modélisation de cheveu par la dynamique directe, (HMT01).



(a) Point d'action du ressort (b) 50 liaisons actives (c) 5 liaisons actives sans l'algorithme (d) 5 liaisons actives avec l'algorithme

Fig. I.15 – Mise en évidence de l'algorithme adaptatif de dynamique directe, sur un pendule constitué de 50 liaisons, (RGL05).

effet, celui-ci est très fin mais possède une rigidité en tension très forte, ce qui est problématique numériquement : pour faire face à ce *problème raide*, des pas très petits doivent être utilisés. Comme le cheveu ne se déforme qu'en flexion et en torsion, les auteurs proposent deux solutions alternatives. La première est de contraindre les particules à ne pas agir en élongation, par exemple à l'aide de multiplicateurs de Lagrange (Bar96). La deuxième est de reformuler le problème en coordonnées réduites, c'est-à-dire en supprimant l'élongation des équations (Fea87). Les deux méthodes sont de complexité équivalente, mais la reformulation en coordonnées réduites est préférable parce qu'elle donne au système une topologie plus simple et parce qu'elle facilite la définition paramétrique de la flexion et de la torsion. C'est la raison pour laquelle le cheveu est ici modélisé comme une chaîne poly-articulée rigide. Les segments sont liés par des liaisons rotules, dont l'orientation est donnée par un quaternion. Pour résoudre l'action de forces sur le système, la *dynamique directe* est utilisée, d'après la méthode de Featherstone (Fea87). Chang *et al.* (CJY02) utilisent une approche similaire pour animer des cheveux directeurs guidant la chevelure. La simulation des déformations de 200 mèches constituées de 15 segments chacune prend 1 seconde par image. Pai *et al.* (PAK00) décrivent de manière approfondie plusieurs algorithmes de complexité linéaire utilisés en dynamique directe.

Redon *et al.* (RGL05) introduisent un algorithme adaptatif pour le calcul dynamique direct de corps articulés. Grâce à des métriques d'erreur, il peut automatiquement simplifier leur dynamique, à partir d'un nombre de degrés de liberté désiré et des points d'application des forces externes et des forces exercées au niveau des liaisons. Cette simplification permet de gagner un facteur 2 dans des situations complexes, mais aussi dans la simulation de câbles, d'autant qu'une manipulation ne déforme généralement que localement certaines portions (figure I.15).

Gayle *et al.* ont repris l'aspect adaptatif dans la modélisation de câbles (GLM06; GSR<sup>+</sup>07; KGL07), en rajoutant la prise en compte de la collision. Le temps de réponse à la collision est de l'ordre de la milli-seconde pour une centaine de joints actifs. La simulation complète ne prend que quelques milli-secondes par pas de temps. Néanmoins, le câble ainsi modélisé n'est pas réaliste mécaniquement : il n'offre de résistance ni à la flexion, ni la torsion. Son comportement mécanique, similaire à une chaîne, facilite la recherche de chemin au travers d'environnements, ce à quoi il est d'ailleurs destiné.

Même si la formulation à coordonnées réduites est efficace pour des systèmes multi-corps avec un grand nombre de segments et un nombre relativement petit de degrés de liberté, il est difficile de leur appliquer un schéma d'intégration implicite. Hadap *et al.* (Hado6) proposent un schéma implicite récursif adapté pour des systèmes chaînés à plusieurs embranchements, en utilisant des équations algébriques différentielles. La simulation de 9 tiges de 15 segments chacune tourne à une fréquence "interactive" de 2 à 4 Hz... Elle se situe donc loin des performances que nous espérons.

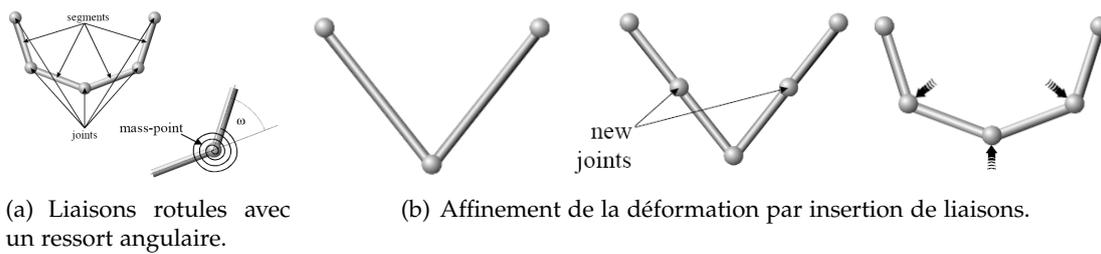
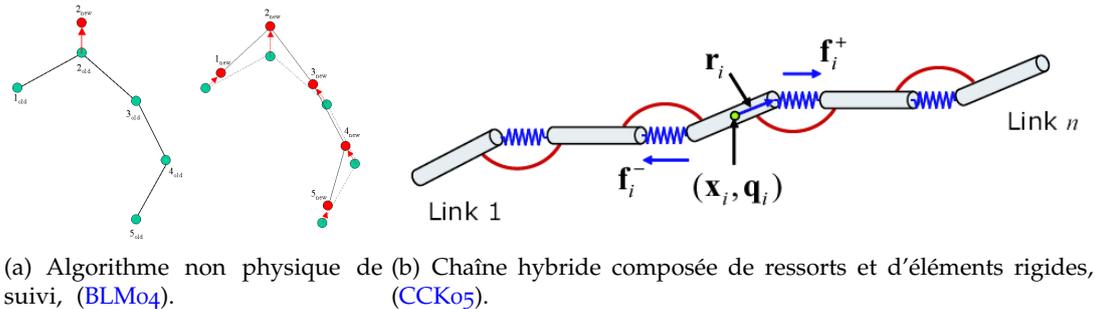


Fig. I.16 – Modélisation multi-résolution itérative de chaîne, (HD00).



**Autres chaînes articulées** Quelques approches atypiques ont tenté d'exploiter une modélisation de type chaîne, sans pour autant utiliser l'algorithme classique correspondant.

Hergenrother *et al.* (HD00) présentent un algorithme temps-réel pour la simulation de câbles, en utilisant la cinématique inverse. Puisque considéré comme inextensible, un câble est modélisé par une chaîne de segments connectés par des liaisons rotules, comme dans (Da99). A chacune de ces liaisons est ajouté un ressort angulaire, l'angle étant celui entre les deux segments. Fixant la position au début et à la fin du câble, l'algorithme calcule de manière itérative sa forme en minimisant l'énergie potentielle du système, constituée de la pesanteur des segments et de la déformation des ressorts. L'équilibre est d'abord résolu avec deux segments, puis leur nombre est doublé à chaque itération par insertion d'une nouvelle liaison au milieu des segments existants, en fonction du temps de calcul alloué pour un pas de simulation (figure I.16(b)). Cette méthode simple donne des résultats visuels intéressants, quoique limités, puisque le câble doit être maintenu à ses extrémités. Les temps de calcul sont également prohibitifs, de l'ordre du quart de seconde pour une résolution de 32 ou 64 éléments.

Brown *et al.* (BMLS01; BLM04) partent eux du principe que lorsqu'une liaison est déplacée, les voisins vont suivre la direction de la nouvelle position du voisin (figure I.17(a)). Cette méthode ne gère donc aucune raideur et ne peut convenir à une simulation qui a la vocation d'être mécaniquement valide.

Pour ne pas gérer de contraintes dures avec des coordonnées réduites ou des multiplicateurs de Lagrange, Choe *et al.* (CCK05; CK05) proposent une chaîne hybride composée de ressorts longitudinaux et angulaires, intercalés avec des éléments rigides (figure I.17(b)). Cet agencement permet d'améliorer le temps de simulation tout en permettant des déformations en torsion, parce que le système peut être résolu par une intégration implicite d'Euler, plus stable. Le temps de calcul d'un pas d'une mèche est de l'ordre de 2 ms pour un pas de temps de 10 ms, en omettant le traitement de la collision.

L'analyse traditionnelle de chaînes suppose que les liaisons mécaniques sont rigides et ne se déforment pas. Bien que cette hypothèse soit appropriée quand les déformations sont petites, les matériaux se déforment toujours dans une certaine mesure si une charge leur est appliquée. Comme le déplacement d'un objet déformable se fait essentiellement par les déformations de celui-ci, on ne peut évidemment pas les négliger. Malgré l'existence de nombreuses méthodes fondées sur des assemblages de corps rigides, seuls les modèles

établis à partir de la mécanique des objets déformables permettent d'obtenir des résultats précis et convaincants.

**Modèle de Cosserat** Si l'utilisation d'un système masse-ressort avec la théorie de Cosserat est exotique (GS06), de nombreux ouvrages modernes traitent du *modèle de Cosserat*, (Ant72, Rub00), ainsi que de mécanique (Sim85; LP95) et de biologie liée à la modélisation mécanique de l'ADN (MMK96).

En informatique graphique, Anjyo *et al.* (AUK92) ont été les précurseurs de l'utilisation de ce modèle. Ils modélisent chaque cheveu par une succession de poutres déformables à flexion constante. Au niveau des liaisons, le passage d'un segment à un autre est décrit à l'aide des coordonnées sphériques (angles d'azimut et de zénith, figure I.17). Se plaçant dans le domaine des petits déplacements qu'ils considèrent suffisants pour avoir un comportement réaliste de cheveux, ils calculent la déformation de chaque poutre à l'aide de cette équation :

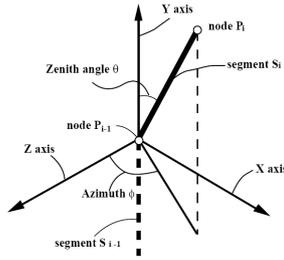
$$\frac{d^2y}{dx^2} = -\frac{E}{MI} \quad (\text{I.47})$$

la courbure  $k$  étant approximée par  $\frac{d^2y}{dx^2}$ . Le temps utilisé pour le calcul d'une image était à l'époque de l'ordre de 50s pour 20.000 cheveux de 20 segments. Néanmoins, le modèle ne prend en compte que la flexion pure dans un plan, ce qui limite la simulation à des cheveux lisses. L'hypothèse des petits déplacements entraîne également une élongation assez importante quand les cheveux tombent verticalement. Dalgegan *et al.* (DTKT93), Kurihara *et al.* (KiAT93) ont étendu cette méthode. Koh *et al.* (KH01) la reprennent pour modéliser les cheveux par des bandelettes. Enfin, Ward *et al.* (WLo3; WGL06) proposent une résolution implicite de cette dynamique.

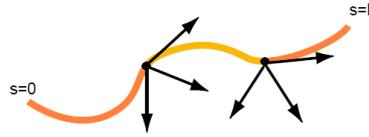
Le modèle de Cosserat s'exprime par la succession d'éléments déformables mais inextensibles, qui peuvent subir la flexion et la torsion. Issu du domaine de la robotique, Pai (Paio2) l'a introduit sous le nom de "Strand", dans le but de simuler le comportement d'un fil de chirurgie. Reprenant un procédé de complexité linéaire bien connu en cinématique, il calcule la configuration statique du fil à un instant donné à l'aide d'un algorithme de chaîne. Cela impose qu'une extrémité soit encastree, tandis que l'autre peut subir une force et un moment. Le calcul de l'équilibre se décompose en deux étapes : la première consiste, en partant de l'extrémité non fixe, à effectuer le bilan des forces élément par élément jusqu'à la base. Ce bilan permet de connaître la courbure et la torsion par unité de longueur, constantes pour chaque élément (figure I.18). A partir de ces données, la deuxième étape permet de reconstruire la tige, cette fois en partant du début. Cette modélisation de tige donne théoriquement des résultats précis pour une résolution élevée. Cependant un inconvénient majeur vient s'ajouter à la contrainte d'encastrement : appliquer des forces de contact, ou spécifier la position ou l'orientation de certains éléments intermédiaires est complexe. Pai recommande l'utilisation d'une méthode dite de "shooting" sans l'appliquer à son modèle. Grégoire *et al.* (GS06) l'ont testée et sont arrivés à la conclusion qu'en plus de son coût important, elle ne converge pas toujours. Enfin, aucune indication sur le temps de résolution du système n'est donnée.

Forts de ces observations, Bertails *et al.* reformulent le modèle de Cosserat par une approche de minimisation énergétique. Elle leur permet de tenir compte des forces externes comme les réponses à la collision, tout en étant robuste et relativement efficace, même si le temps de calcul limite l'interactivité à une centaine de mèches de cheveux. Les résultats visuels sont toutefois très satisfaisants : le fait de considérer la section du cheveu elliptique apporte une réponse convaincante et élégante à la formation des boucles (figure I.19). Notons que la construction de la mèche se fait en remarquant qu'un élément à courbure et à torsion constantes est un morceau d'hélice.

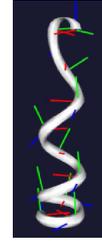
C'est cette remarque qui a permis à Bertails *et al.* d'aboutir au modèle de super-hélices (BAC<sup>+</sup>06). En considérant les courbures et torsions comme des degrés de liberté du mo-



**Fig. I.17** – Liaisons repérées par les coordonnées polaires, (AUK92; DTKT93; KiAT93; KH01; WL03)



**Fig. I.18** – Modèle de "Strand", constitué de 3 segments, (Paio2).



**Fig. I.19** – Influence de la section elliptique sous l'effet de la gravité, (BAQ<sup>+</sup>05).

dèle, ils appliquent le formalisme des équations de Lagrange aux tiges de Cosserat. L'ensemble des déformations est ainsi obtenu par la résolution d'un système linéaire. L'application de forces extérieures telles que la gravité ne se répartit cependant pas sur la bande de la matrice du système, donnant une résolution en complexité quadratique ( $o(n^2)$ ) par rapport au nombre  $n$  de segments. Même s'il est rapide pour simuler quelques éléments, il devient vite plus coûteux que tous les autres modèles, généralement de complexité linéaire. La nature des segments fait également qu'un grand nombre est nécessaire à une bonne gestion des contacts internes (vdHNGT03). Les auto-collisions sont donc difficilement simulées en temps interactif. Néanmoins, le modèle de super-hélices donne un comportement visuel convaincant, appuyé par de nombreuses validations visuelles, mettant notamment en évidence le flambage dynamique et le couplage flexion-torsion.

Les approches chaînées des modèles de Cosserat, bien qu'apportant des résultats tout à fait satisfaisants en terme de précision mécanique et de performance, gardent les défauts inhérents à une approche chaînée, à savoir la nécessité d'un point fixe à une extrémité et la difficulté de contraindre bilatéralement des points entre les extrémités de la tige. Certaines approches continues, que nous allons maintenant étudier, offrent de bien meilleures capacités de manipulation, pour une précision mécanique équivalente.

### c Approches continues

La méthode classique pour simuler les déformations et le mouvement d'un objet consiste d'abord à écrire les équations continues à partir de la mécanique des milieux continus, puis à en trouver une formulation approchée, de manière à obtenir des équations discrètes adaptées à la résolution numérique. Cela passe souvent par une approximation polynomiale d'autant plus précise que son degré est élevé.

**Différences finies** La méthode des *différences finies* consiste à approcher les termes dérivés des équations aux dérivées partielles (EDP) ou des équations différentielles ordinaires (EDO) par leur développement limité à un certain ordre. La précision sera d'autant plus importante que l'ordre est élevé. Il correspond au degré du polynôme d'approximation obtenu à l'aide de la formule de Taylor Young :

**Théorème I.1** Soient  $n \in \mathbb{N}^*$  et  $f$  une fonction  $n$  fois dérivable au voisinage  $]x - \epsilon, x + \epsilon[$  d'un point  $x$ , alors

$$\forall h / |h| < \epsilon, f(x+h) = \sum_{i=0}^n \frac{h^i}{i!} f^{(i)}(x) + \epsilon^n o(h) \quad (\text{I.48})$$

avec  $f^{(n)}$  la dérivée  $n^{\text{ème}}$  de  $f$  et  $o$  le résidu tel que  $\lim_{h \rightarrow 0} o(h) = 0$ .

Pour approximer la dérivée  $n^{\text{ème}}$  d'une fonction, il faut donc prendre le développement à l'ordre  $n$  et isoler le terme dérivé d'ordre  $n$ . D'autres types d'approximation existent également.

**Définition I.2** On appelle schéma numérique le type d'approximation choisi pour approximer le calcul d'une dérivée.

Appliquer un schéma numérique revient à substituer toutes les dérivées du système d'équation. Le résultat est un système matriciel, le plus souvent creux, puisque chaque terme dérivé est généralement approché par une expression qui dépend de ses voisins les plus proches de l'espace. Les différences finies constituent une méthode simple et directe pour discrétiser un problème (TPBF87; TW88). La précision et la stabilité sont cependant loin d'être garanties, d'autant que les objets 1D sont des problèmes *numériquement raides*. Malgré l'utilisation du schéma de Hou *et al.* (HKS98) qui propose une solution spécifique avec un ordre élevé, Bertails *et al.* (BAC+06) ne sont pas parvenu à obtenir une solution stable. Nous écartons donc directement l'utilisation d'un schéma à différences finies.

**Eléments finis classiques** La plus connue des méthodes de discrétisation d'équations est sans conteste la *méthode des éléments finis*. Issue de la nécessité de résoudre des problèmes structurels complexes en ingénierie civile et en aéronautique, elle a notamment pour origine les travaux de Hrennikoff (Hre41). Le développement des éléments finis a vraiment commencé au milieu des années 1950 à Berkeley (WR99), mais ce sont Strang et Fix (SF73) qui en ont posé les fondations mathématiquement rigoureuses. L'article d'Oden (Ode90) en donne un historique détaillé. La méthode des éléments finis est aujourd'hui utilisée dans la plupart des simulateurs industriels, et dans des domaines aussi variés que la biologie, la mécanique, l'électromagnétisme, l'acoustique, etc. De nombreux ouvrages traitent le sujet en détail, mais pour acquérir rapidement une idée claire et précise de la méthode, le lecteur est invité à consulter les cours en ligne de Champany, Felippa ou encore Garrigues (Gar07).

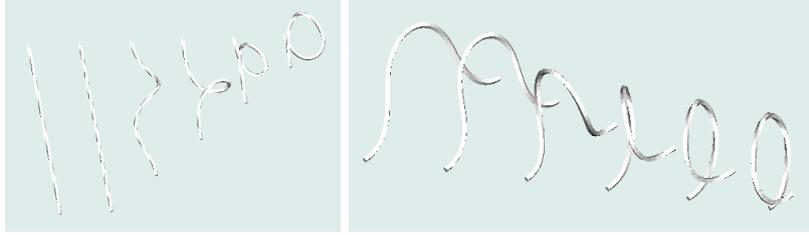
La méthode des éléments finis permet de résoudre de manière discrète et approchée, mais suffisamment fiable, une équation aux dérivées partielles. Le problème comporte également des conditions initiales permettant d'assurer l'existence et l'unicité d'une solution. Deux étapes sont nécessaires pour la trouver. Le problème doit d'abord être reformulé sous sa forme dite *faible*, ou *variationnelle*, souvent à partir du principe des travaux virtuels ou du principe de moindre action. Puis vient la discrétisation dans un domaine, espace de dimension finie, composée de sous-domaines appelés *éléments* ou *mailles*, l'ensemble de ces mailles constituant un maillage. Ces éléments peuvent être de formes diverses : segments dans le cas 1D, triangles, rectangles dans le cas 2D, tétraèdres dans le cas 3D... Sur chaque élément, la solution est définie par interpolation de différents points d'échantillonnage, appelés *noeuds*, généralement situés sur les sommets. Dans le cas d'un problème dynamique de mécanique, nous avons alors un système matriciel de taille conséquente mais linéaire dont la solution est constituée par la position ou le déplacement  $\mathbf{u}$  de ses  $n$  noeuds et donc approchée par morceaux :

$$\mathbb{M}\ddot{\mathbf{u}} + \mathbb{D}\dot{\mathbf{u}} + \mathbb{K}\mathbf{u} = \mathbf{F} \quad (\text{I.49})$$

où  $\mathbb{M}$  la matrice d'inertie liée à la masse,  $\mathbb{D}$  la matrice d'amortissement liée aux frottements visqueux,  $\mathbb{K}$  la matrice de raideur liée à l'élasticité du système, et enfin  $\mathbf{F}$  le vecteur des forces internes et externes du système. Si le problème est étudié statiquement, le système devient alors :

$$\mathbb{K}\mathbf{u} = \mathbf{F} \quad (\text{I.50})$$

Si les matrices  $\mathbb{M}$  et  $\mathbb{D}$  sont généralement constantes, la matrice de raideur  $\mathbb{K}$  ne l'est que si l'on se place dans l'hypothèse des petits déplacements. Les résultats sont alors peu convaincants dès que l'on s'éloigne de la configuration initiale. Dans le cas des grands déplacements, il est nécessaire de recalculer  $\mathbb{K}$  à chaque pas de temps, ce qui peut vite s'avérer très pénalisant dans le cas d'une matrice dense. Un autre problème lié à la taille du système est le coût de calcul prohibitif mis en jeu lors de la résolution du système,



**Fig. I.20** – Manipulations mettant en évidence le couplage flexion-torsion, (WH04).

dès lors qu'une fine discrétisation est nécessaire pour garantir la stabilité et la précision demandée. Même si les architectures des cartes graphiques actuelles permettent des gains de temps énormes par rapport aux processeurs centraux (GGHM05), la simulation non-linéaire d'objets volumiques par la méthode des éléments finis demeure encore éloignée du temps-réel. Néanmoins, une propriété intéressante des éléments 1D est la possibilité d'obtenir un système avec une matrice bande, en plaçant des noeuds sur la fibre moyenne.

Les éléments finis sont largement utilisés en mécanique et en robotique pour modéliser des objets 1D déformables des structures architecturales, des bras de robots ou des câbles. Ils le sont beaucoup moins en informatique graphique.

En mécanique et en robotique, la notion de *géométriquement exact* est souvent rattachée à la modélisation de poutre et de câbles. Simo en est à l'origine (SF89) : "Une fois la cinématique de base admise, il existe deux niveaux d'approximations dans la résolution numérique d'un problème de dynamique des structures. Le premier concerne la géométrie de la poutre et les équations d'équilibre qui régissent son évolution, le second est relatif à la résolution numérique proprement dite. Dans l'approche géométriquement exacte, acceptant les hypothèses cinématiques qui définissent les transformations admissibles, la géométrie de la poutre ainsi que ses équations d'équilibre sont traitées de manière exacte. Ainsi les seules approximations commises concernent le second niveau, c'est-à-dire la discrétisation spatiale et l'intégration numérique temporelle." Cette hypothèse revient à dire que la description des déformations ne contient pas d'approximation de type géométrique. Contrairement aux éléments finis classiques où la rotation est issue de la translation, elle est ici traitée indépendamment en considérant un milieu de type Cosserat. Parmi les contributeurs à cette théorie, on peut également citer Ibrahimbegovic (Ibr95), Vu-Quoc (VQDE96), Li (Lio0) et dernièrement Primault et Boyer (Pri03; BP05). Ces modélisations se démarquent par leur rigueur et leur précision, au détriment de la lisibilité pour des non-mécaniciens, et généralement de l'efficacité. Toutefois, le modèle de Primault et Boyer permet d'atteindre des pas de temps honorables, de l'ordre d'une dizaine de milli-secondes.

Wakamatsu *et al.* s'intéressent depuis près de 15 ans à la modélisation de câbles statiques (WHI95; Wak98; WMW<sup>+</sup>00; Hiroo; WH04) et dynamiques (WMHA97), en essayant de promouvoir leur méthode basée sur la géométrie différentielle. Celle-ci exprime les variations de déformations entre les repères successifs qui décrivent le câble. La flexion et la torsion sont exprimées sous forme de variations des angles d'Euler, qui définissent le passage d'un repère à un autre. Comme pour les approches à base de chaînes, cette formulation en grands déplacements nécessite un encastrement. La résolution du système est faite par la méthode de Ritz (Els61), formulation particulière des éléments finis. Le modèle a le mérite d'être validé par de nombreuses comparaisons entre manipulations réelles et virtuelles ; des comportements non linéaires complexes comme le couplage flexion-torsion sont également reproduits (figure I.20). La précision s'est améliorée depuis les premiers résultats, mais l'erreur est cependant encore de l'ordre de 6% dans l'espace (WH04). Son inconvénient majeur est le temps de calcul de plusieurs secondes par pas de simulation, ce qui le rend loin du temps interactif. D'autres publications de Wakamatsu *et al.* sont consacrées à la réalisation de noeuds (WMHA97; WAH06).

En informatique graphique, très peu de modèles sont construits à base d'éléments

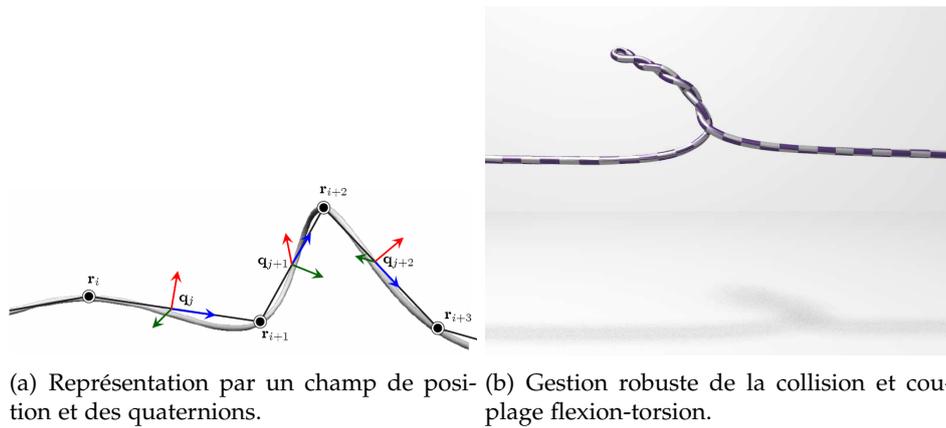


Fig. I.21 – Représentation d'une corde par un champ de position et des quaternions, (ST07)

finis. L'approche de Cotin *et al.* (CDL<sup>+</sup>05; LCDNo6a; LCDNo6b) utilise un modèle de poutre incrémental à éléments finis, pour modéliser l'évolution d'un cathéter dans l'appareil sanguin. Tiré du livre de Przemieniecki (Prz68), ce modèle tient compte de toutes les déformations en *petits déplacements* d'une poutre, à savoir la traction-compression, le cisaillement, la torsion et la flexion. La matrice élémentaire de raideur prend en compte la translation et la rotation d'un élément, elle est donc de dimension  $12 \times 12$ . Les auteurs adoptent une résolution statique (équation I.50). La linéarité des déformations ne pouvant suffire à représenter fidèlement le comportement d'un cathéter, les auteurs proposent d'actualiser la matrice de raideur à chaque pas de la simulation, en fonction de la nouvelle configuration. L'ajout d'un amortissement permet le retour de la poutre à sa configuration initiale si les efforts extérieurs sont relâchés ; sans amortissement elle reviendrait seulement à la configuration du pas précédent. Entre 100 et 200 éléments sont nécessaires à la simulation d'un cathéter, soit une fourchette de 1200 à 2400 équations à résoudre. Bien qu'un tel système bande puisse être résolu en temps-réel (de l'ordre de quelques millisecondes), l'ajout de contraintes augmente considérablement le temps de calcul. Pour améliorer les performances, les auteurs proposent une méthode pour subdiviser le problème en sous-problèmes d'un ou plusieurs éléments, qui pourront être résolus sur des processeurs différents puis rassemblés, le tout en temps linéaire. Cependant, la méthode impose une condition initiale à une extrémité, telle qu'un encastrement.

Construit à partir du modèle de Cosserat, le modèle de Spillmann (ST07) s'inspire des travaux de Grégoire *et al.* (GS06) pour passer d'une formulation discrète à continue. Chaque noeud est constitué d'un champ de position  $\mathbf{r}$  (3 degrés de liberté), couplé à un champ de rotation, représenté par un quaternion  $\mathbf{quat}$  (4 degrés de libertés), comme illustré sur la figure I.21.

Attaché au milieu de chaque élément, ce quaternion est équivalent à une matrice de rotation, donc à un repère  $(\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3)$  dont Spillmann *et al.* établissent les relations avec le quaternion. Le couplage des deux champs s'effectue dans un premier temps par l'ajout de deux contraintes. L'une porte sur le vecteur tangent du champ de position et le vecteur tangent  $\mathbf{d}_3$  du repère associé à chaque élément, qui doivent rester confondus (contrainte de dimension 3) :

$$\frac{\mathbf{r}'}{\|\mathbf{r}'\|} - \mathbf{d}_3 = \mathbf{0} \quad (\text{I.51})$$

L'autre impose que la norme du quaternion reste unitaire (contrainte de dimension 1) :

$$\|\mathbf{quat}\|^2 - 1 = 0 \quad (\text{I.52})$$

En effet, l'ensemble des orientations ou rotations de l'espace correspond au sous-espace des quaternions unitaires. Les 7 degrés de liberté et les contraintes de dimension 3 et 1

nous donnent des blocs de dimension  $11 \times 11$  à résoudre, ce qui fait vite augmenter la taille du système. Les auteurs ont d'abord essayé de traiter le système ainsi formé, mais le résultat s'est avéré inadapté au temps réel. Pour palier le problème, deux simplifications similaires à celles de Grégoire *et al.* sont proposées sur les contraintes : la première (I.51) est remplacée par une formation énergétique, ce qui permet de découpler la résolution en position et en quaternions. La deuxième (I.52) est remplacée par la normalisation du quaternion après chaque résolution du système. Ces contraintes n'apparaissent plus comme des équations supplémentaires mais agissent comme des forces sur le système. Le désavantage flagrant est malheureusement la perte de précision et de stabilité, puisqu'elles violent l'équilibre des forces. Les manipulations figurant dans la vidéo accompagnant la publication ne permettent pas d'éprouver la stabilité du modèle, les déformations paraissant très molles. Le modèle résultant est toutefois de complexité linéaire en temps et produit des animations visuellement convaincantes.

**Modèle spline** L'idée d'animer mécaniquement des courbes géométriques pour de l'animation physique n'est pas récente. Dans leur papier précurseur sur les déformations élastiques (TPBF87), Terzopoulos *et al.* ont donné une première formulation énergétique  $V$  d'une courbe, basée sur ses déformations *géométriques* en élongation  $\epsilon$ , en flexion  $k$  et en torsion  $\tau$ , pondérées par les résistances à la déformation, respectivement  $\alpha$ ,  $\beta$  et  $\gamma$  :

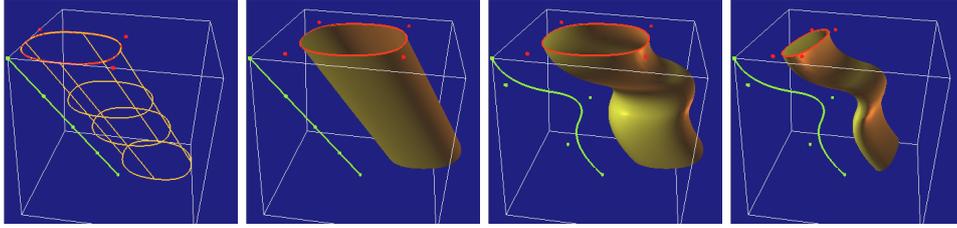
$$V(r) = \int_0^L \alpha(\epsilon - \epsilon^0)^2 + \beta(k - k^0)^2 + \gamma(\tau - \tau^0)^2 du \quad (\text{I.53})$$

la résolution se faisant avec le principe fondamental de la dynamique :

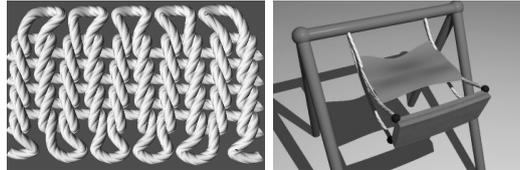
$$\frac{\partial}{\partial t} \left( \mu \frac{\partial \mathbf{r}}{\partial t} \right) + \frac{\partial V(r)}{\partial \mathbf{r}} = \mathbf{F} \quad (\text{I.54})$$

où  $\mu$  représente la densité linéique de la courbe,  $\mathbf{F}$  les forces extérieures et  $V(r)$  une fonctionnelle qui donne l'énergie potentielle de déformation de la courbe. Même si cette formulation énergétique de la courbe n'est pas valide mécaniquement, de par sa formulation erronée de la torsion, elle permet d'animer une courbe en réponse à des forces extérieures.

Les premières utilisations de mécanique sur des splines ont pour but de fournir un outil de modélisation intuitif *temps-réel* à base de B-Splines uniformes (TC90; BW90; CW92) ou non (NURBS) (TQ94; QT96) (voir figure I.22), car elles sont fortement présentes dans les outils de modelage commerciaux. Thingvold et Cohen (TC90) ont d'abord proposé des B-Splines déformables dont les points de contrôle sont des masses reliées par des ressorts en translation et en rotation. De plus, ils utilisent les propriétés de raffinement des B-Splines pour augmenter la discrétisation en plaçant des nouveaux points de masse. Bloor et Wilson (BW90), Celniker et Welch (CW92), ainsi que de Terzopoulos et Qin (TQ94; QT96) reprennent la formulation énergétique continue des déformations en élongation et en flexion (équation I.53). Ces paramètres sont intuitifs (QT96) : augmenter  $\alpha$  a tendance à réduire la longueur et augmenter  $\beta$  à rendre l'objet plus rigide. La flexion est calculée à partir d'une simple approximation de la courbure et la torsion n'est pas prise en compte. Comme le soulignent les auteurs, les effets exacts des valeurs utilisées sur la spline restent à déterminer. Pour contraindre les splines dans l'espace, des contraintes linéaires (CW92) et non linéaires (TQ94) sont utilisées. Ces dernières sont modélisées par les *multiplieurs de Lagrange* et associées à la *méthode de stabilisation de Baumgarte* (Bau72), nécessaire dans le cadre dynamique. Enfin, les splines sont déformées via l'action de ressorts. Les déformations restent cependant limitées à une validation visuelle et n'ont pas pour but de simuler un comportement réaliste d'objets 1D, d'autant que les splines constituent ici essentiellement des surfaces. Pour des raisons d'efficacité, les modèles antérieurs à celui de Qin et Terzopoulos (TQ94) ne résolvent que des équilibres statiques avec la masse et l'amortissement concentrés sur les points de contrôle. Leur apport majeur réside dans l'utilisation des *équations de Lagrange* (équations II.43) pour résoudre l'équilibre



**Fig. I.22** – Un cylindre généralisé sculpté en appliquant des ressorts sur deux NURBS (rouges et vertes), (TQ94).



(a) Tricot, (RNG99) (b) Cordes et dossier de balançoire modélisées à partir de splines 1D et 2D, (R60)

**Fig. I.23** – Modèle spline à base de ressorts en élongation.

dynamique de la spline. Dans ce formalisme, tout objet est défini par un ensemble de coordonnées généralisées qui déterminent entièrement la configuration du modèle. Une spline étant entièrement définie par ses points de contrôle, les auteurs proposent d'assimiler les coordonnées généralisées aux coordonnées des points de contrôle.

L'utilisation des équations de Lagrange et des multiplicateurs de Lagrange a ensuite été reprise dans les nombreux travaux de Rémion *et al.*, essentiellement pour simuler le comportement de mailles de tricot (figure I.23(a)). Leur formulation claire et élégante d'une spline aboutit à la résolution d'un système matriciel linéaire (RNG99; Nou99). Des forces externes peuvent être également appliquées en tout point de la spline. Même si l'ensemble des propriétés matérielles du modèle, telles que la masse et le frottement visqueux, sont définies de manière continue comme des fonctions paramétriques, ce n'est pas le cas des déformations. En effet, de manière analogue à (WW90), l'énergie potentielle ne figure pas dans les équations mais est remplacée par l'action de ressorts non linéaires, fixés au niveau des points de contrôle. Nourrit (Nou99) montre que sous certaines conditions, on peut décomposer le problème en trois sous-systèmes. En tenant compte de leur structure bande, la complexité de leur résolution non contrainte est de complexité linéaire  $o(n)$  par rapport aux  $n$  points de contrôle, dans chaque direction de l'espace. Cette optimisation permet au modèle de spline d'atteindre des meilleures performances, mais elle empêche l'utilisation de contraintes couplées, par exemple celles de distances. Pour tenir compte du comportement de la spline en flexion, Lenoir *et al.* (LMGC02) rajoutent des ressorts au milieu des segments de spline (figure I.24(a)). La collision est traitée simplement à l'aide de sphères et de pénalités (figure I.24(b)). Notons que Rémion *et al.* étendent leur méthode aux objets splines 2D et 3D (RNG00, RNN00).

La première modélisation d'une déformation interne de spline mécaniquement rigoureuse est donnée par Nocent *et al.* (NR01; Noco1). Elle concerne l'élongation, en grands déplacements et en grandes déformations. Sa comparaison avec un système de ressorts corrobore le lien entre la vision microscopique de l'élasticité et la vision macroscopique du ressort. Il est d'ailleurs possible d'établir tout un parallèle, ce qui justifie d'une certaine façon l'utilisation si massive de ressorts pour la simulation d'objets déformables. L'étude montre qu'un grand nombre de ressorts est nécessaire pour approcher l'équilibre donné par la modélisation continue I.25, sans compter la gêne occasionnée par les oscillations parasites induites lors des mouvements.

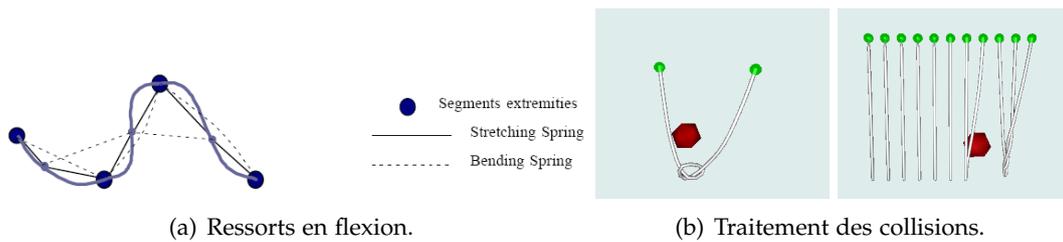


Fig. I.24 – Modèle spline à base de ressorts en élévation et en flexion, (LMGC02).

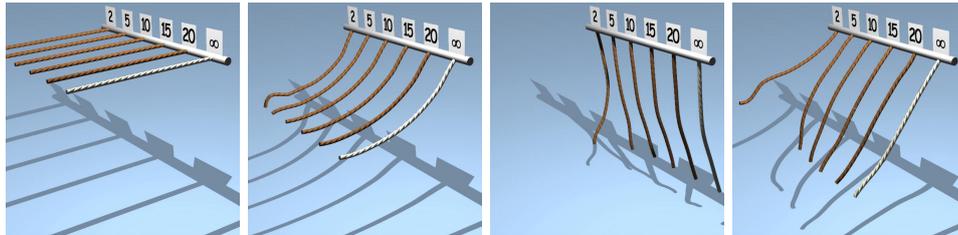


Fig. I.25 – Comparaison entre la modélisation continue et la modélisation à l'aide de ressorts, (NR01).

C'est pourquoi Lenoir *et al.* ont ensuite concentré leur efforts sur l'optimisation de l'énergie continue d'élongation par des techniques de précalculs, ainsi que sur la modélisation d'une énergie continue de flexion en petits déplacements, pour aboutir à une simulation temps-réel très rapide et visuellement convaincante (Leno4). Si les splines possèdent un pouvoir de représentation supérieur aux autres discrétisations, le nombre de degrés de liberté nécessaire par unité de longueur varie selon les déformations locales. En effet, plus de points de contrôle sont nécessaires aux endroits de forte courbure. Si leur nombre n'est pas suffisant, un noeud peut ne pas se serrer entièrement (figure I.26(b)). Si Phillips *et al.* ont tenté une première approche peu concluante (PLK02), Lenoir *et al.* ont proposé une technique automatique d'insertion et de suppression de points sur des B-Splines. L'ajout d'un noeud préserve à la fois la géométrie et l'équilibre mécanique (LGMCo5) : il existe en effet des techniques d'insertion exacte (Sch81; Mey91), quelle que soit la configuration de la spline. La suppression est elle beaucoup moins évidente, d'autant qu'il n'existe aucun algorithme qui permette de le faire de manière exacte sur une courbe gauche. En pratique, elle est effectuée sur une portion quasiment droite de spline. Lenoir *et al.* mettent également en évidence une propriété fort intéressante : l'ajout de 4 noeuds identiques pour un point donné d'une B-Spline permet de la séparer en deux parties mécaniquement indépendantes (figure I.26(b)).

**Dynamique modale** En mécanique, le but de l'analyse modale est de décomposer dans le domaine fréquentiel les formes naturelles d'un objet lors de vibrations. D'une manière expérimentale, on applique des vibrations à une structure, en mesurant l'amplitude des

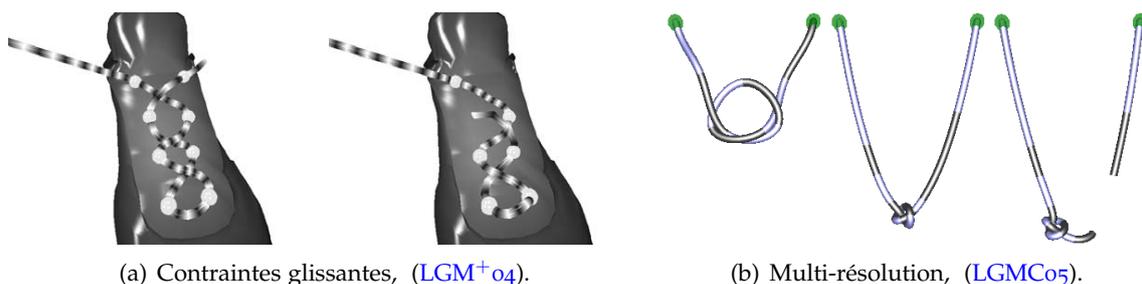
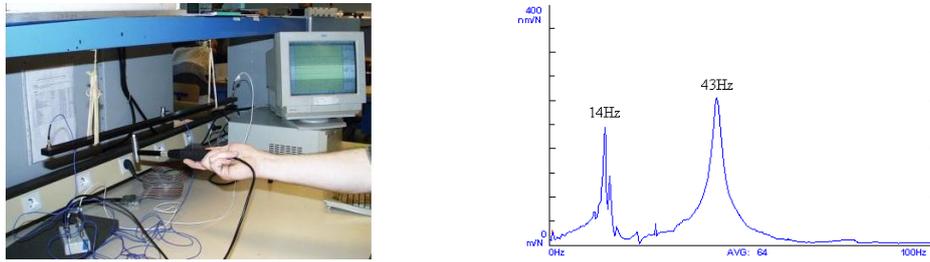


Fig. I.26 – Techniques avancées du modèle spline.

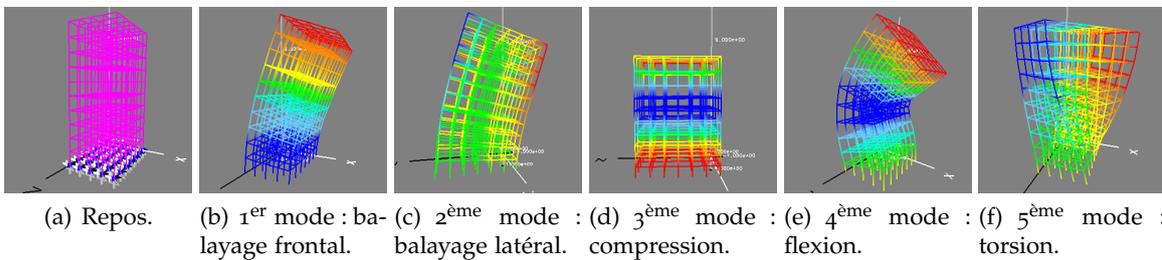
déformations résultantes (figure I.27(a)). Le signal temporel est ensuite converti en signal fréquentiel sur lequel les pics correspondent aux modes propres (figure I.27(b)).



(a) Marteau d'impact et trois accéléromètres (b) Spectre de l'analyse modale. Les deux pics correspondants pour déterminer fréquences propres et dépendent à des modes propres du matériau. formées modales d'une poutre.

**Fig. I.27** – Analyse expérimentale d'une poutre.

Pour obtenir le système matriciel linéaire, une discrétisation de type éléments finis est souvent utilisée. Une décomposition en valeurs propres est alors faite à partir des équations I.49 et I.50 et de quelques hypothèses sur les matrices. Les valeurs propres et les vecteurs propres ont pour interprétation physique les fréquences vibratoires et les formes correspondantes. Généralement, les modes désirés sont les plus petites fréquences parce que les formes correspondantes sont celles qui ont le plus d'impact dans les déformations. Les mouvements à haute fréquence sont de plus généralement sources d'instabilité numérique. Pour améliorer la résistance d'une structure aux tremblements de terre, une analyse modale est souvent effectuée. La figure I.28 montre les cinq premiers modes d'une modélisation de type éléments finis d'un immeuble excité à sa base. La base de valeurs propres



**Fig. I.28** – Différents modes propres d'un immeuble modélisé par éléments finis. ©Gnu FDL

permet d'écrire autant d'équations que de valeurs propres. Ces équations correspondent chacune au mouvement du mode propre. En ne gardant que les modes à basse fréquence, on obtient un système diagonal, donc rapide à résoudre dans le cas des petits déplacements, c'est-à-dire quand  $\mathbb{K}$  est constant. Dans le cas contraire, la diagonalisation n'est plus valable au pas de simulation suivant. La rapidité de résolution d'un tel système a vite suscité un vif intérêt des chercheurs pour la simulation temps-réel, notamment chez Pentland (PW89). Ce type d'approche, qui consiste à superposer les déformations modales de l'objet simulé, est appelé *dynamique modale*.

A l'aide d'une décomposition modale et d'un processus stochastique, Stam (Sta97) a proposé un comportement réaliste d'arbres balayés par le vent ou d'algues soumis à des courants marins. L'utilisation des dix premiers modes permet, pour des petits arbres (moins d'une cinquantaine de branches), d'avoir une simulation visuellement quasi-identique à une utilisant tous les modes. Son algorithme a cependant l'inconvénient majeur qu'il recalcule les valeurs propres du système entre deux états successifs. Sa complexité est donc cubique par rapport au nombre de branches.

Barzel (Bar97) a également repris cette approche dans le cadre d'un film d'animation, Toy Story. Un comportement visuel plutôt que physique est également recherché, allant

même jusqu'à la caricature. Les déformations d'éléments 1D déformables tels que des cordes, des cordons et des ressorts sont ainsi reproduites. La méthode étant construite à partir du comportement vibratoire d'une corde maintenue à une extrémité ou des deux côtés, elle ne reste plus applicable dans d'autres situations.

Quant à l'idée de varier le nombre de modes selon la déformation, elle apparaît d'autant plus difficile que l'impact lié à l'ajout ou au retrait d'un mode vibratoire sur le comportement et la stabilité d'un tel système est imprévisible (BW92).

## 4 Résolution numérique

Outre la discrétisation spatiale, une discrétisation temporelle peut être nécessaire pour animer des objets lors d'une simulation. Dans certains cas de simulation, seule la configuration de repos, est intéressante, comme dans le cas d'une coiffure (BAQ<sup>+</sup>05). L'inertie du système peut être également négligeable, à cause d'une masse faible ou d'un mouvement lent, comme pour des câbles fins maintenus à leurs extrémités (WHI95), et l'on peut considérer les déplacements comme une succession d'états d'équilibres statiques. On parle alors de *simulation quasi-statique*. Dans les autres cas, le régime transitoire entre les équilibres est recherché, comme dans le cas d'une chevelure en mouvement (BAC<sup>+</sup>06; WL03) ou d'un arbre balayé par le vent (GCF01). L'objet peut aussi tout simplement ne pas passer par des équilibres statiques, parce qu'il évolue constamment (pendule oscillant, objet en chute libre). On parle dans ce cas de *simulation dynamique*.

Dans la suite de cette section, nous allons étudier les techniques de résolution pour ces deux types d'équilibre. Pour comprendre facilement les enjeux d'un tel choix, nous allons appliquer l'équilibre quasi-statique puis dynamique à l'étude d'un système masse-ressort à une dimension.

Le système est composé de masses  $m$  mises bout-à-bout, de position  $x_i$  et de vitesse  $v_i$ , reliées par des ressorts. Les vecteurs d'état  $\mathbf{x}$  et  $\mathbf{v} = \dot{\mathbf{x}}$  regroupent ces données. Les forces potentielles des ressorts agissant sur les masses sont notées  $P_i$ , tandis que les forces externes, comme la gravité, sont notées  $F_i$ . L'exposant indique le temps, qui débute à  $t_0$ . Par exemple,  $x_i^n = x_i(t_0 + n\Delta t)$ . L'écart entre deux positions  $\mathbf{x}^{n+1}$  et  $\mathbf{x}^n$  est noté  $\Delta^{n+1}\mathbf{x} = \mathbf{x}^{n+1} - \mathbf{x}^n$ . L'écart entre deux vitesses s'exprime de manière analogue :  $\Delta^{n+1}\mathbf{v} = \mathbf{v}^{n+1} - \mathbf{v}^n$ .

### a Résolution quasi-statique

**Définition I.3** Une simulation est dite quasi-statique si tous les états intermédiaires du système au cours du mouvement sont des états définis, proches d'états d'équilibres.

Cela implique que le déséquilibre des variables d'état, responsable de la transformation, soit infiniment petit. Pour qu'une transformation soit quasi-statique, il faut donc qu'elle soit très lente, pour que l'on puisse considérer qu'elle est constituée d'une succession d'états d'équilibres statiques.

**Equilibre statique d'un système masse-ressort 1D** Des lois de Newton, on peut déduire l'énoncé suivant : un système mécanique est en équilibre statique dans un référentiel galiléen, si l'effet des efforts extérieurs qui s'appliquent sur lui est nul (somme des forces extérieures nulle et somme des moments extérieurs nulle).

L'équilibre statique au niveau d'une masse s'écrit donc :

$$0 = P_i + F_i \quad (\text{I.55})$$

tandis que l'équilibre statique du système de masses-ressort s'exprime de façon analogue sous forme vectorielle :

$$\mathbf{0} = \mathbf{P} + \mathbf{F} \quad (\text{I.56})$$

Trouver l'équilibre statique de l'ensemble des masses consiste à résoudre ce système d'équations pour trouver la position de repos des masses. Si l'on déplace l'une des masses par petits pas successifs, trouver l'équilibre quasi-statique au cours du mouvement consiste à résoudre successivement ce système à chaque étape.

**Optimisation** En mathématiques, l'*optimisation* se réfère à l'étude de problèmes dans lesquels on cherche à minimiser ou à maximiser une fonction en recherchant la solution dans un ensemble donné. La fonction n'étant souvent pas minimisable de manière analytique, on a recours à une résolution numérique itérative, jusqu'à ce qu'une précision donnée soit atteinte. L'enjeu est donc d'assurer non seulement la convergence vers la solution globale du problème et de ne pas s'arrêter à une solution locale, mais aussi de converger le plus rapidement possible, c'est-à-dire trouver la solution avec le moins d'itérations possibles. Dans notre recherche d'équilibre statique, l'équation I.56 est appelée *condition d'optimalité*. Les forces potentielles dérivant d'un potentiel énergétique défini comme positif, le problème peut se ramener à minimiser cette énergie.

De nombreuses méthodes existent, chacune étant plus ou moins adaptée à un problème donné (PFTV88a). Il n'existe donc pas de méthode parfaite !

Néanmoins, il est possible de les classer en deux grandes catégories : les unes qui demandent seulement l'évaluation de la fonction étudiée, et les autres qui demandent en plus celle de sa dérivée, appelée *gradient* pour une fonction vectorielle. Ces dernières convergent généralement mieux et plus rapidement mais ne compensent pas toujours les calculs supplémentaires nécessaires à la dérivée.

**Méthode de Newton** La *méthode de Newton*, ou *méthode de Newton-Raphson* est un algorithme simple et efficace pour trouver les approximations des racines d'une fonction vectorielle (ici le vecteur  $\mathbf{x}$  de la fonction  $\mathbf{P}$ ). L'algorithme consiste à linéariser la fonction et à prendre le point d'annulation de la linéarisation comme approximation du déplacement vers la solution recherchée :

$$\mathbf{P}^{n+1} = \mathbf{P}^n + \frac{\partial \mathbf{P}}{\partial \mathbf{x}} \Delta^{n+1} \mathbf{x} \quad (\text{I.57})$$

Pour plus de précision, on itère ensuite la procédure avec la solution trouvée :

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \Delta^{n+1} \mathbf{x} \quad (\text{I.58})$$

Dans les cas favorables, les approximations successives obtenues convergent avec une vitesse quadratique.

Comme cette méthode nécessite de construire explicitement la Hessienne, souvent trop lourde à calculer, les méthodes dites *quasi-Newton* prennent le parti de l'approximer. Actuellement, l'unanimité est faite autour de celle trouvée indépendamment par Broyden, Fletcher, Goldfarb et Shanno (Bro70).

**Méthodes utilisant le gradient** La *méthode du gradient*, ou méthode de la plus grande pente, consiste à suivre la pente négative du gradient de la fonction par pas proportionnels.

La méthode du gradient a une convergence initiale rapide mais la présence de vallées la fait osciller et rend sa convergence très lente quand nous nous rapprochons de la solution. L'avantage de cette méthode est la convergence certaine, même avec un vecteur initialement loin de la solution.

La *méthode du gradient conjugué* (HS52; She94) est la méthode la plus puissante et la plus fiable utilisée en optimisation multidimensionnelle sans contrainte. Elle repose sur le concept de direction conjuguées parce que les gradients successifs sont orthogonaux entre eux et aux directions précédentes.

Par exemple, Bertails *et al.* (BAQ<sup>+</sup>05) utilisent l'approche de descente rapide de gradient de Fletcher *et al.* (FP63) pour minimiser l'énergie d'un cheveu sous l'influence de la gravité.

Wakamatsu *et al.* ont d'abord utilisé la méthode de Ritz, (Els61) (WHI95; WMW<sup>+</sup>00; Hiroo), puis la méthode d'optimisation non-linéaire quasi-Newton BFGS.

Grégoire *et al.* (GS06; GS07), utilisent ces méthodes pour résoudre l'équilibre statique d'un câble. La méthode par défaut est celle de Newton, puis diverses variantes du gradient conjugué, en cas d'échec.

La méthode du *recuit simulé* (KGV83) est une métaheuristique inspirée d'un processus utilisé en métallurgie. Ce processus alterne des cycles de refroidissement lent et de réchauffage (recuit) qui tendent à minimiser l'énergie du matériau. Elle permet de ne pas trop vite enfermer la recherche de solution près d'un minimum local mais de tendre vers la solution globale du problème. L'idée est d'effectuer un mouvement selon une distribution de probabilité qui dépend de la qualité des différents voisins : les meilleurs voisins ont une probabilité plus élevée ; les moins bons ont une probabilité plus faible.

## b Résolution dynamique

Desbrun (DSB99) fournit une explication très claire sur l'intégration explicite et implicite d'un système masse-ressort 1D. Nous la reprenons ici.

**Equilibre dynamique d'un système masse-ressort 1D** L'équilibre dynamique au niveau d'une masse s'écrit alors :

$$m \frac{dv_i}{dt} = P_i + F_i \quad (\text{I.59})$$

tandis que l'équilibre dynamique du système de masses-ressort s'exprime de façon analogue sous forme vectorielle :

$$m \frac{d\mathbf{v}}{dt} = \mathbf{P} + \mathbf{F} \quad (\text{I.60})$$

Pour animer un tel système, il faut le discrétiser par rapport au temps. C'est le rôle des schémas d'intégration. Deux grandes formulations existent : celles qui se basent sur l'état suivant, dites *explicites*, et celles qui sont définies à partir de l'état suivant, dites *implicites*.

**Schéma d'intégration explicite** Le schéma d'Euler explicite peut être utilisé :

$$\begin{aligned} m \frac{\Delta^{n+1}\mathbf{v}}{\Delta t} &= \mathbf{F} + \mathbf{P}^n \\ \mathbf{x}^{n+1} &= \mathbf{x}^n + \mathbf{v}^{n+1}\Delta t \end{aligned} \quad (\text{I.61})$$

en considérant que les forces externes n'évoluent pas au cours du pas de temps :  $\mathbf{F} = \mathbf{F}^n = \mathbf{F}^{n+1}$ . Le système d'équations I.63, exprimé en fonction du déplacement  $\Delta^{n+1}\mathbf{x}$  entre les deux états  $n$  et  $n + 1$ , donne :

$$\frac{m}{\Delta t^2} \Delta^{n+1}\mathbf{x} = \mathbf{F} + \mathbf{P}^n + \frac{m}{\Delta t} \mathbf{v}^n \quad (\text{I.62})$$

Avec la méthode d'Euler explicite, les forces de l'état  $n$  affectent la vitesse à l'état  $n + 1$ . Des schémas d'ordre plus élevés, tels que Runge-Kutta à l'ordre 2 ou 4, sont meilleurs en terme de précision (Buto3).

S'il est facile à implémenter, ce schéma nécessite un pas de temps d'intégration  $\Delta t$  qui doit être inversement proportionnel à la racine de la raideur du système. Ce critère est connu sous le nom de *condition de Courant* (CFL67). Dans le cas contraire, le système divergera : supposer que les forces restent constantes sur un trop grand intervalle de temps induit souvent une variation trop brusque et un éloignement trop important de la solution

pour s'en rapprocher à nouveau. C'est le problème des *équations raides* : la stabilité ne peut être atteinte qu'à un pas de temps très faible pour des schémas explicites. La simulation d'objets 1D en est un. Dans le cadre de la simulation de mailles de tricot par des splines, Rémion *et al.* (RNG99; Nou99) utilisent une intégration de type Runge-Kutta à l'ordre 4. Pour préserver la stabilité du système, ils ont été amenés à choisir un pas de temps relativement petit (de l'ordre de  $10^{-4}$  seconde), principalement à cause des nombreux ressorts qui gèrent les déformations.

Un autre schéma d'Euler, dit implicite, est plus adapté dans ce cas. S'il a été utilisé dès les premiers travaux sur les objets déformables de Terzopoulos (TPBF87), ce n'est que dix ans plus tard que ses avantages ont été vraiment mis en valeur, notamment avec la simulation de vêtements par Baraff et Wikin (BW98) ou d'objets 3D quelconques à base de masses-ressorts par Desbrun (DSB99).

**Schéma d'intégration implicite** L'idée du *schéma d'Euler implicite* est de remplacer les forces de l'état  $n$  par celles de l'état  $n + 1$  :

$$\begin{aligned} m \frac{\Delta^{n+1} \mathbf{v}}{\Delta t} &= \mathbf{F} + \mathbf{P}^{n+1} \\ \mathbf{x}^{n+1} &= \mathbf{x}^n + \mathbf{v}^{n+1} \Delta t \end{aligned} \quad (\text{I.63})$$

Cette simple substitution améliore la stabilité de manière conséquente : les déplacements calculés sont maintenant cohérent avec les forces du même état. Les forces sont toujours constantes pendant le pas de temps, mais en théorie, quelque soit le pas de temps utilisé, la position de l'état  $n + 1$  du système sera toujours équilibré par rapport aux forces du même état.

Pour implémenter ce schéma, nous devons calculer les forces potentielles  $\mathbf{P}^{n+1}$  sans connaître les positions  $\mathbf{x}^{n+1}$  des masses à l'état suivant. Pour ce faire, la méthode de Newton-Raphson est souvent utilisée, avec un développement de Taylor Young à l'ordre 1 (équation I.57) :

$$\mathbf{P}^{n+1} = \mathbf{P}^n + \frac{\partial \mathbf{P}}{\partial \mathbf{x}} \Delta^{n+1} \mathbf{x}$$

En notant  $\mathbb{K} = -\frac{\partial \mathbf{P}}{\partial \mathbf{x}}$  la valeur négative du Jacobien de  $P$ , et en insérant l'équation I.57 dans les équation I.63, nous obtenons :

$$\left( \frac{m}{\Delta t^2} + \mathbb{K} \right) \Delta^{n+1} \mathbf{x} = \mathbf{F} + \mathbf{P}^n + \frac{m}{\Delta t} \mathbf{v}^n \quad (\text{I.64})$$

**Différence entre les deux schémas** Pour expliquer la différence entre le schéma explicite et le schéma implicite d'Euler, nous pouvons réécrire l'équation I.64 :

$$m \frac{\Delta^{n+1} \mathbf{v}}{\Delta t} = \left( \mathbb{I} + \frac{\Delta t^2}{m} \mathbb{K} \right)^{-1} \left( \mathbf{F} + \mathbf{P}^n + \Delta t \mathbb{K} \mathbf{v}^{n+1} \right) \quad (\text{I.65})$$

et la comparer au schéma I.61 :

$$m \frac{\Delta^{n+1} \mathbf{v}}{\Delta t} = \mathbf{F} + \mathbf{P}^n$$

Premièrement, un vecteur de forces supplémentaire fait son apparition dans le schéma implicite :

$$\tilde{\mathbf{P}}^n = \Delta t \mathbb{K} \mathbf{v}^n \quad (\text{I.66})$$

Il s'ajoute au vecteur de forces internes  $\mathbf{P}^n$  et il est homogène à une viscosité. Intuitivement, cela veut dire que le mouvement d'un point est influencé par le mouvement de ses voisins : *une particule suit le déplacement local*. Cette force de viscosité est proportionnelle au pas de

temps et à la raideur du système, sources d'instabilités. Elle augmente donc la stabilité de la résolution.

Deuxièmement, la force résultante est multipliée par une matrice  $W$  :

$$W = \mathbb{I} + \frac{\Delta t^2}{m} \mathbb{K} \quad (\text{I.67})$$

Selon Kass (Kas95), cette matrice  $W$  est un filtre dont la bande varie en fonction de la raideur et du pas de temps au carré. Dans le cas où leur produit est très petit, la bande est très étroite et tend vers un filtre de Dirac. L'intégration implicite est alors équivalente à une explicite. Dans le cas opposé où le produit est très grand, la bande est très large et tend vers un gain constant. Ce filtre impose alors une translation uniforme des points, comme pour un mouvement rigide. Donc, plus le système est raide, plus la bande du filtre est large, et plus le mouvement d'un point se répercute sur tous les autres.

L'intégration implicite offre des simulations stables, sans avoir à ajouter empiriquement des coefficients d'amortissement. Le pas de temps peut surtout être plus grand sans entraîner de divergence, ce qui réduit le temps de calcul pour un intervalle de temps donné. Néanmoins, il faut garder en mémoire que le gain en stabilité se fait au détriment de la précision, par l'introduction de viscosité et d'un filtrage.

## 5 Gestion des interactions

L'intérêt de la simulation d'un objet déformable virtuel est d'une part qu'il interagisse avec son environnement extérieur et de l'autre que l'on puisse le manipuler. D'abord, pour définir si un objet est en contact avec un autre, une phase de *détection de collision* est nécessaire. Dans le cas d'un objet déformable 1D, on distingue généralement les collisions avec les autres objets et les collisions avec lui-même, les *auto-collisions*. Ensuite, si tel est le cas, une phase de *réponse à la collision* permet d'adapter le comportement de l'objet à son environnement, à l'aide de *contraintes*. Ces contraintes permettent également à un utilisateur d'imposer ou d'altérer le comportement de l'objet. Enfin, on peut souhaiter lier deux objets de nature et de modélisation différentes. On parle alors de *couplage*. Nous allons dans cette section nous intéresser aux approches spécifiques aux objets déformables 1D, pour chacun de ces points. La collision est donc traitée de manière très succincte. Pour une étude plus détaillée, nous invitons le lecteur à consulter l'état de l'art de Jimenez (JTT01) et les thèses de Redon (Red02) et de Dequidt (Deq05).

### a Détection de collision

Deux corps sont détectés en collision si leur intersection géométrique est non vide. La détection de collision constitue généralement l'étape la plus difficile et la plus coûteuse lors du traitement des collisions entre objets, car elle doit chercher, parmi tous les objets constituant la scène, ceux qui sont susceptibles de s'intersecter à chaque instant. Pour avoir une vue d'ensemble des méthodes existantes, nous recommandons au lecteur l'état de l'art de Teschner *et al.* (TKH<sup>+</sup>05). Elles sont très nombreuses, aussi nous intéressons-nous ici aux méthodes adaptées aux objets déformables 1D.

Pour juger de la qualité d'un algorithme de détection de collision, on peut se baser sur les deux critères suivants : la *propriété de conservation* de la méthode, qui signifie qu'à chaque instant, toutes les collisions sont effectivement détectées (Red02), et l'efficacité de la méthode.

Il est possible de répartir les méthodes de détection de collisions en deux grandes catégories. Les *méthodes discrètes* constituent la grande majorité des méthodes. Elles échantillonnent la trajectoire des objets et ne détectent que des interpénétrations entre objets à certains instants. Deux objets se déplaçant peuvent donc se traverser entre deux pas de temps, sans que l'algorithme ne puisse le détecter. Les autres, moins nombreuses, sont les

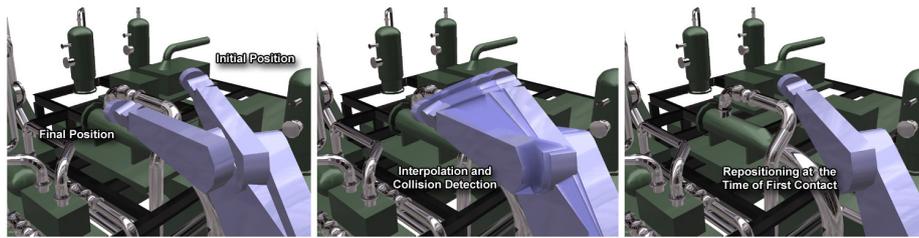


Fig. I.29 – Principe de la détection continue de collision, (RKLMO4)



(a) Répartition irrégulière, à partir de l'abscisse paramétrique. (b) Répartition équidistante des sphères, à l'aide de l'abscisse curviligne.

Fig. I.30 – Importance de l'utilisation de l'abscisse curviligne dans le calcul de répartition des sphères de collision, (Leno4).

*méthodes continues* (RKLMO4; ZRLKO7) : elles calculent l'instant de contact entre les objets pendant la détection de collisions (figure I.29). Elles sont cependant plus lourdes à mettre en oeuvre que les méthodes de détection discrète.

Le domaine d'utilisation des algorithmes de détection est souvent limité à certaines primitives géométriques pour les objets, pour réduire le temps et la complexité des calculs. C'est la méthode la plus fréquemment utilisée. Ainsi, un objet déformable 1D est souvent représenté par une série de sphères réparties selon la ligne moyenne (PLKO2; LMGC02; BLMO4; WBD<sup>+</sup>05).

Dans le cas des splines, Phillips *et al.* (PLKO2) proposent une détection de collision à base de sphères placées sur les points de contrôle. Si la spline s'étire, des points de contrôle doivent être intercalés pour combler les vides. Lenoir (Leno4) place les sphères contiguës, de rayon identique à celui de la section. Il fait remarquer que l'abscisse paramétrique ne peut pas être utilisée pour placer les sphères, car celle-ci est irrégulière. Il propose donc un algorithme pour le faire à partir de l'abscisse curviligne (figure I.30). Comme la courbe spline peut s'étirer ou se compresser, la répartition est recalculée à chaque pas de simulation.

Des cylindres peuvent également être employés (KIAT93; DTKT93; BKCN03). Les tests se ramènent à un calcul de distance entre segments et sont toujours à temps constant. L'objet 1D peut également être défini par l'équation de la ligne moyenne, mais le coût n'est plus constant et augmente très rapidement avec le degré de l'équation de la courbe, dans le cas d'auto-collisions. Enfin, des boîtes parallèles aux axes (ABB) (vdb97) ou des boîtes orientées (OBB) (GLM96; LSHLO2) constituent d'autres solutions.

Dans le cas d'un objet 1D replié sur lui-même comme un intestin, Raghupathi *et al.* (RGF<sup>+</sup>04) ont proposé une méthode stochastique de calcul de minimum de distance entre deux surfaces. Afin d'améliorer la réponse en collision, ils utilisent également un algorithme récursif pour chercher les voisins potentiellement en collision et fournir une réponse unique.

Afin de pouvoir simuler, en un temps raisonnable, un grand nombre d'objets en interaction, de nombreuses techniques permettent d'accélérer la détection en répartissant les objets en groupes. On distingue deux approches basées sur la description du milieu. Les *méthodes eulériennes* partitionnent l'espace (PCP01).

Les *méthodes lagrangiennes* approximent le volume de l'objet à travers une hiérarchie de primitives (BVH) (GLM96; LMO3; GLMO6). Plusieurs méthodes à base de sphères existent (Hub96; GNRZO2). Brown *et al.* (BMLS01; BLMO4), ainsi que Wang *et al.* (WBD<sup>+</sup>05; WVB05) utilisent la méthode hiérarchique de sphères englobantes de Guibas *et al.* (GNRZO2). Pour un objet qui évolue sur une dimension, elle est de complexité

$o(n^{4/3})$  dans le pire des cas. De plus, comme les liens entre sphères n'évoluent pas au cours des déformations, seules les positions et les rayons des sphères doivent être recalculés à chaque étape de simulation.

Une fois la collision détectée entre deux objets, il s'agit de modifier leur état pour empêcher ou corriger une interpénétration.

## b Réponse à la collision

Les réponses à la collision peuvent être réparties en deux groupes, les méthodes à pénalité et les méthodes à base de contraintes.

Les *méthodes à pénalités* constituent les réponses les plus simples et les plus utilisées. Elles consistent à insérer des ressorts qui vont tendre à rapprocher les points des objets devant être en contact. L'inconvénient de cette méthode est qu'elle est approximative, puisqu'on ne peut pas connaître à l'avance la raideur du ressort à ajouter au système. Si la raideur est trop faible, la contrainte n'est pas respectée, les objets se traverseront ; à l'inverse une raideur trop forte offre un meilleur respect de la contrainte mais peut introduire des problèmes raides lors de la résolution numérique. De plus, les méthodes à pénalité demandent un calcul robuste et précis de la profondeur de pénétration des points de collision, une irrégularité conduisant à des oscillations (HFS03; HTK+04). Cette formulation a cependant l'avantage de passer d'un système contraint à un système non contraint avec des forces supplémentaires. Le coût de cette méthode est donc faible. C'est pourquoi ce type de réponse à la collision est privilégié dans la simulation de cheveux (KiAT93; CCK05; Bero6).

Les *méthodes analytiques* imposent des contraintes unilatérales, de non pénétration, pour tous les points de contact, afin de calculer les forces de réponse. Robustes, elles procurent des résultats exacts. Les *contraintes unilatérales* sont définies par des inéquations mais il est plus facile et plus rapide de résoudre un système d'équations en ajoutant successivement au système libre les contraintes violées. Des schémas de résolution ont été proposés, formulant un *problème de complémentarité linéaire* (Bar94; Hado6) ou non linéaire (BS98), ou bien utilisant des *contraintes bilatérales*, les *multiplieurs de Lagrange* (PB88; Pla92). Contrairement aux méthodes à pénalité, les contraintes sont directement associées à la résolution du système, et des méthodes itératives doivent être employées. Pour un grand nombre de points de contacts, le problème est loin d'être trivial et peut s'avérer très coûteux. Cette approche est très peu utilisée pour des objets déformables 1D, mais elle se révèle adaptée au formalisme lagrangien, et en particulier les splines dynamiques (R60; RNG00). Cependant, l'ajout de  $c$  contraintes à une spline augmente considérablement la complexité du système (LMGC02), à savoir  $o(cn^2 + c^2n + c^3)$ , limitant l'interactivité à un nombre limité de points de contrôle et de contraintes.

La méthode n'est par contre pas adaptée aux approches chaînées : spécifier la position ou l'orientation de certains éléments intermédiaires est complexe. Pai (Pai02) recommande l'utilisation d'une méthode dite de "shooting" sans l'appliquer à son modèle. Grégoire *et al.* (GSo6) l'ont testée et sont arrivés à la conclusion qu'en plus de son coût important, elle ne converge pas toujours vers une solution. Cependant, Gayle *et al.* (GLMo6) proposent une méthode linéaire par rapport au nombre de degrés de liberté, qui tient compte également de l'approche multi-résolution de Redon (RGL05).

Dans la simulation d'un cathéter évoluant dans des vaisseaux, le nombre de points de contacts est important. Les méthodes mentionnées ci-dessus sont inadaptées à cause du glissement du cathéter le long des parois. Cotin *et al.* (CDL+05) proposent une méthode adaptée.

En fait, les contraintes sont surtout utilisées dans le cadre de la manipulation.

### c Manipulation

Les contraintes restreignent l'univers des configurations possibles d'un objet déformable en lui imposant des conditions à respecter. L'exemple le plus classique est la contrainte d'encastrement, qui permet de résoudre un problème statique, en lui imposant une condition aux limites.

Si la contrainte est formulée en fonction de la position de l'objet, on dit qu'elle est *holonome*. Si elle fait intervenir la vitesse, elle est dite *cinématique*.

De manière générale, un système contraint bilatéralement peut s'écrire sous la forme d'une fonction  $f$  à minimiser et d'une autre fonction  $g$  à annuler, dépendant toutes deux des degrés de liberté  $\mathbf{q}$  du système et de leurs dérivées  $\dot{\mathbf{q}}$  :

$$\begin{cases} \text{Minimiser } f(\mathbf{q}, \dot{\mathbf{q}}) \\ \text{sujette à } g(\mathbf{q}, \dot{\mathbf{q}}) = 0 \end{cases} \quad (\text{I.68})$$

Pour résoudre ce système, on procède par différentiation (Bar96) ou par la création d'une équation différentielle de contrainte (Bau72). Une nouvelle variable, généralement notée  $\lambda$  et appelée *multiplicateur de Lagrange*, est alors ajoutée au système.

Comme nous l'avons vu plus haut, les contraintes bilatérales s'ajoutent facilement à un système, sauf pour les modélisations à base de chaîne, comme le mentionnent Grégoire (GSo6) et Spillmann (STo7). Le modèle de spline dynamique est lui très adapté (Leno4) : contraintes en position, vitesse, accélération, couplage de points. Lenoir *et al.* ont également proposé des contraintes glissantes (LGM<sup>+</sup>o4) qui permettent de faire glisser une spline au travers d'un point, avec ou sans frottement. Elles leur permettent de modéliser des lacets sur une chaussure et des noeuds coulants. Ils proposent également un schéma qui baisse la complexité de résolution du système. Elle reste cependant quadratique.

Au chapitre III, nous détaillerons l'utilisation des multiplicateurs de Lagrange puis nous proposerons plusieurs nouveaux types de contraintes à partir de ceux-ci. Au chapitre IV, nous montrerons que l'on peut conserver la complexité linéaire du système non contraint par un agencement dynamique des multiplicateurs dans le système. Nous détaillerons l'algorithme mis au point.

### d Schémas de couplage

Si les objets peuvent entrer en contact, il peut être également nécessaire de les coupler, c'est-à-dire des les faire évoluer de manière solidaire, un câble ou un fil évoluant rarement seul dans l'espace : crochet attaché au bout d'un câble de remorquage, aiguille de fil de chirurgie, etc. Un modèle peut avoir toutes les qualités possibles, il n'aura pas grand intérêt si l'on ne peut l'utiliser que seul. Les interactions entre fluide et solides sont connues pour être un problème difficile de couplage qui fait l'objet d'intenses recherches de la communauté (BBBo7). Il existe également de nombreuses modélisations d'objets rigides et déformables, qui comme nous l'avons vu, peuvent être résolues avec des lois d'équilibre diverses et variées. Heureusement, les formulations de la mécanique du solide et des milieux continus sont équivalentes et peuvent assez facilement se ramener de l'une à l'autre : principe fondamental de la dynamique, principe des travaux virtuels, équations de Lagrange... Il existe deux types de couplages. Le *couplage faible*, ou *couplage lâche*, qui consiste à résoudre indépendamment et alternativement chacun des deux systèmes, en mettant à jour à chaque pas les conditions initiales. Prenons deux systèmes  $S_1$  et  $S_2$ . Le couplage lâche s'effectue en 4 étapes (BW97; FPF99) (figure I.31(a)) : la *prédiction* de  $S_2$  (par exemple à l'aide de la vitesse ou du gradient de force), la *résolution* de  $S_1$  en tenant compte de la prédiction de  $S_2$ , la *substitution* du nouvel état de  $S_1$  dans  $S_2$ , puis enfin la *résolution* de  $S_2$ . L'avantage est qu'il n'entraîne quasiment aucun surcoût et qu'il est précis géométriquement, mais il y a toujours un décalage temporel entre les deux systèmes. Le *couplage fort* consiste à rassembler les deux systèmes en un seul, pour les résoudre simultanément, en

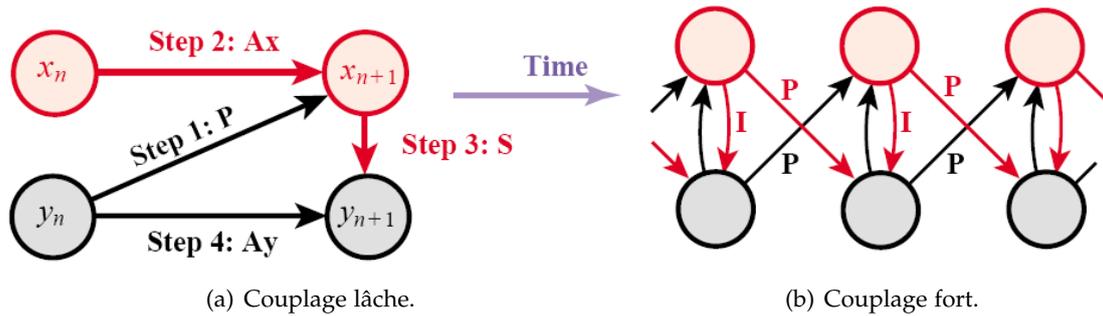


Fig. I.31 – Méthodes de couplage d'objets, (FPF99).

garantissant l'exactitude et la simultanéité des deux solutions (figure I.31(b)). Cela peut paraître à première vue la meilleure solution, mais le couplage faible est à lui préférer, ce pour plusieurs raisons. A moins que les deux systèmes soient linéaires, le temps de calcul pour résoudre le nouveau système est nettement plus important que la somme des temps pour chacun des systèmes pris séparément. Ensuite, pour des raisons évidentes de temps et de simplicité de conception, on ne peut pas développer des couplages forts pour toutes les modélisations existantes. Enfin, la tendance actuelle de développement des plateformes de simulation est que chaque nouveau modèle soit ajouté indépendamment des autres, ce qui va en l'encontre du couplage fort. Monaghan *et al.* (MDCA98) décrivent une méthode de couplage fort entre poutres 1D et solides 3D modélisés par éléments finis, Lenoir et Fonteneau (LFo4), entre splines dynamiques et objets rigides, à l'aide de multiplicateurs de Lagrange, Baraff (Bar96) et Faure (Fau99) entre des objets rigides. Rémion *et al.* (RNN01), Sifakis *et al.* (SSIF07) proposent l'utilisation d'un unique formalisme pour coupler facilement les différents modes de déformations.

Nous établirons au chapitre III un schéma de couplage lâche simple et efficace, qui convient à l'utilisation conjointe des équations et des multiplicateurs de Lagrange.

## BILAN

Après une brève introduction à la mécanique des milieux continus, nous avons vu que les méthodes générales de simulation d'objets virtuels ne conviennent pas à nos objectifs de précision et de performance. C'est donc tout naturellement que nous nous sommes orientés sur la théorie des poutres, partie de la mécanique dédiée aux objets unidimensionnels, pour comprendre ensuite les subtilités des modélisations existantes. Chacune possède des avantages et des inconvénients en terme de réalisme physique, de stabilité, de vitesse de résolution et d'aptitude à être contrainte.

Nous allons maintenant dresser un bilan comparatif des méthodes, afin de sélectionner celle la plus à même de remplir nos objectifs :

- elle doit être *physique*, c'est-à-dire pensée à partir de la mécanique des milieux continus, ou mieux encore, à partir d'une théorie dédiée aux éléments 1D (théorie des poutres, des tiges de Kirchhoff ou de Cosserat, etc.). Elle doit gérer la flexion et la torsion, et dans une moindre mesure la traction-compression. Sa validation doit se faire non seulement d'un point de vue visuel, mais aussi par des mesures. Nous éliminons d'emblée toutes les méthodes basées sur le positionnement empirique de ressorts, mais aussi les méthodes modales et les méthodes continues ne prenant en compte que les petits déplacements (AUK92);
- l'utilisateur doit pouvoir opérer une *manipulation intuitive et précise*, mais aussi *librement contrainte*, c'est-à-dire qu'aucune contrainte ne doit être imposée par le modèle, comme le maintien aux extrémités <sup>4</sup>. Elle doit par contre autoriser des contraintes

<sup>4</sup>sachant qu'en résolution statique une contrainte est obligatoire

dures en tout point de la ligne moyenne, afin de réaliser des assemblages de plusieurs éléments de splines et des couplages avec d'autres systèmes mécaniques.

- dans une optique de *temps interactif*, la méthode doit être rapide mais aussi de complexité linéaire par rapport au nombre de degrés de liberté ou d'éléments qui compose l'objet, pour pouvoir simuler de longs fils ou câbles. L'ajout de contraintes ne doit pas altérer la complexité de résolution. Le pouvoir de représentation des degrés de liberté et les aspects multi-résolution sont un plus.

Le tableau [I.1](#) dresse un récapitulatif des propriétés des modèles déformables 1D étudiés dans ce chapitre.

Modèle	Précision			Interaction		Performance		
	Discretisation	Déformations	Validité	Liberté	Contraintes	Complexité	Tps élém/pas	Multi-rés
(JL04)	ressorts	E,T,F	faible	oui	oui	$o(n)$	?	non
(WBD <sup>+</sup> 05)	ressorts	E,T,F	faible	oui	oui	$o(n)$	?	non
(GS07)	ressorts	E,T,F	vis., couplages	oui	oui	$o(n)$	10ms	non
(CCK05)	chaîne rig., ressorts	T,F	faible	oui	oui	$o(n)$	?	non
(Hado6)	chaîne rigide	T,F	vis., couplages	non	oui	$o(n)$	100ms	non
(GSR <sup>+</sup> 07)	chaîne rigide	Sans	faible	non	non	$o(n)$	1-5ms	oui
(Paio2)	tige Cosserat	T,F	visuelle	non	non	$o(n)$	?	non
(BAC <sup>+</sup> 06)	tige Cosserat	T,F	vis., couplages	non	non	$o(n^2)$	1ms	non
(BP05)	continue, FEM	E,C,T,F	très élevée	oui	oui	$o(n)$	10ms	non
(WH04)	continue, Ritz	E,T,F	bonne 6%	non	non	$o(n^2)$	1s	non
(CDL <sup>+</sup> 05)	continue, FEM	E,C,T,F	visuelle	non	oui	$o(n)$	10ms	non
(ST07)	continue, FEM	E,T,F	vis., couplages	oui	non	$o(n)$	0.05ms	non
(LGMCo5)	continue, spline	E,F (pts dépl)	visuelle	oui	oui	$o(n^2)$	0.5ms	oui

**Tab. I.1** – Comparatif des principaux modèles déformable 1D existants, sur les critères de validité physique, d'interaction et de performance.

A première vue, aucun de ces modèles ne semble convenir : tous pèchent soit en validité physique, soit en interaction, soit en performance. Le modèle de Grégoire *et al.* (GSo6; GSo7) est relativement lent et l'utilisation de ressorts ne garantit pas la stabilité en équilibre dynamique. Le modèle de Bertails *et al.* (BAC<sup>+</sup>06) souffre d'une complexité quadratique et présente de grosses difficultés à être contraint par les multiplicateurs de Lagrange, comme celui de Pai (Pai02). Les modèles de Boyer et Primault (BP05) et de Contin *et al.* nous semblent trop coûteux. Celui de Spillmann (ST07) est très séduisant, mais le découplage nécessaire du champ de position et d'orientation laisse planer le doute sur sa précision. L'absence de gestion des contraintes bilatérales rend la manipulation difficile.

Cependant, le modèle de spline (LGMC05) attire particulièrement notre attention. Il présente un avantage certain au niveau de son pouvoir de représentation : un degré de liberté d'une spline couvre plus de déformation qu'un élément à déformation constante ou nulle. Il possède en plus des propriétés de multi-résolution beaucoup plus précises et faciles à mettre en oeuvre que celles existantes pour la méthode des éléments finis. L'existence des contraintes glissantes est également un atout, dans l'optique de modéliser des sutures ou de faire glisser un câble dans une rondelle.

Dans les chapitres suivants, nous proposons une évolution du modèle de spline, conforme à nos attentes en termes de précision, de pouvoir d'interaction et performance. Celui qui suit s'intéresse à la validité mécanique.

## Chapitre II

# Déformations d'une spline matérielle

*La corde faite de trois brins est difficile  
à rompre.*

PROVERBE MALAIS

**M** AINTENANT que nous avons adopté la spline dynamique comme modèle d'objet déformable 1D, nous examinons ici tout le cheminement intellectuel qui, à partir de la définition d'une spline et des équations de Lagrange, aboutit à la proposition d'un modèle précis et stable. Tout au long du chapitre, nous nous attachons à être à la fois rigoureux et suffisamment clairs pour qu'un non mécanicien puisse comprendre et implémenter notre modèle. Pour ce faire, nous étudions d'abord les splines d'un point de vue géométrique. Nous proposons une méthode d'évaluation de la précision de l'interpolation spline, qui nous permet également de comparer la méthode spline à celle des éléments finis. Nous matérialisons ensuite la spline, en lui ajoutant une section. Nous lui appliquons alors de manière rigoureuse la théorie des poutres, pour proposer un modèle que nous appelons *Spline Dynamique Géométriquement Exact* (TGAB07), qui prend en compte les déformations élastiques et plastiques, sans approximer la géométrie. Nous lui appliquons enfin un schéma Euler-Implicite linéarisé, qui lui apporte stabilité (TGDM07). Notre contribution consiste ici à le rendre efficace, en formulant de manière optimale les différents termes du système d'équations à résoudre. Nous terminons par la validation du modèle, en évaluant d'une part la précision des déformations obtenues, et en reproduisant d'autre part des comportements de couplage entre déformations.

## A MODÉLISATION D'UNE SPLINE MATÉRIELLE

Si Nocent (Noc01) et Lenoir (Leno4) ont choisi de généraliser leur modélisation théorique à l'ensemble des splines, nous pensons au contraire que certaines sont plus adaptées que d'autres, et même que la précision mécanique est tributaire de la spline choisie. C'est pourquoi nous définirons une spline, puis nous énumérerons les propriétés géométriques utiles à la simulation mécanique, afin de de convenir du meilleur choix à adopter.

### 1 Spline géométrique

Avant l'arrivée des ordinateurs, les calculs numériques se déroulaient à la main. Bien que les fonctions définies par morceaux étaient déjà utilisées, les polynômes étaient privilégiés pour leur facilité d'utilisation.

Schoenberg fut le premier à introduire le terme *spline* pour l'approximation polynomiale par morceaux continûment dérivables, dans deux articles publiés en 1946 (Sch46a; Sch46b). Il s'inspira vraisemblablement du "conic lofting", technique utilisée par les anglais pour fabriquer le fuselage des avions, pendant la deuxième guerre mondiale. Elle servait à passer de maquettes de conception de petite échelle à la taille réelle, en reportant certains points clés sur des patrons (MN97). Les splines furent vraiment utilisées en tant que telles au début des années 1960 dans l'aéronautique, avec les travaux de Ferguson chez Boeing, puis introduites indépendamment dans l'industrie automobile par de Casterl jau chez Citroën, Bézier chez Renault, ainsi que Birkhoff et de Boor chez General Motors. Depuis l'arrivée des ordinateurs, ces courbes rencontrent un vif succès dans les communautés de conception assistée par ordinateur (CAO) et d'informatique graphique, pour leurs multiples propriétés qui facilitent la modélisation. Le premier ouvrage qui traite des splines est celui de Nielson et Walsh chez Pratt & Whitney Aircraft (NW67). De nombreux modèles de splines ont ensuite été proposés. Deux grandes familles permettant de les classer : les splines d'*interpolation*, pour lesquelles les points de contrôle sont situés sur la courbe, et les splines d'*approximation* dans le cas contraire. Tous ces modèles de splines diffèrent selon les propriétés géométriques données à l'objet modélisé et le degré d'influence de chacun des points de contrôle sur la forme de l'objet.

Dans cette section, nous définirons d'abord ce qu'est une spline puis présenterons les diverses propriétés utiles à la simulation mécanique d'un objet déformable 1D. Nous détaillerons ensuite les modèles splines existants. Enfin, nous matérialiserons ces courbes géométriques en leur rajoutant de la matière tout autour, par la donnée d'une section.

#### a Définitions et propriétés

Nous allons donner ici la définition d'une spline et décrire les propriétés essentielles pour la simulation physique, à partir de quelques références (Grig9; Leno4).

En informatique, on appelle *fonction spline* ou *spline*, toute fonction s'obtenant par combinaison linéaire de fonctions élémentaires. D'un point de vue mathématique, elle se définit de la manière suivante :

**Définition II.1** *Etant donné un ensemble de  $n$  points  $\mathbf{q}_i$ , appelés points de contrôle, et un ensemble de  $n$  fonctions  $b_i$  à valeurs réelles définies sur un intervalle fermé, appelées fonctions d'influence, une spline est définie comme la courbe d'équation paramétrique*

$$\mathbf{q}(u) = \sum_{i=0}^{n-1} b_i(u) \mathbf{q}_i \quad (\text{II.1})$$

On peut, sans perte de généralité, supposer cet intervalle égal à  $[0, 1]$ . On désigne par *segment* la portion de spline entre deux points de contrôle (NR01).

Les propriétés des splines dépendent de celles des fonctions d'influence. Elles permettent de les classer par famille, selon l'usage que l'on veut en faire.

### Normalité

**Propriété II.1** Une spline vérifie la condition de normalité si et seulement si la somme des fonctions d'influences associées est égale à 1 partout sur leur intervalle de définition :

$$\forall u \in [0, 1], \sum_{i=0}^{n-1} b_i(u) = 1 \quad (\text{II.2})$$

La conséquence directe de cette propriété est que tout point de la spline est le barycentre de ses points de contrôle, pondérés par les valeurs correspondantes à l'abscisse des fonctions d'influence. D'autre part, la somme des dérivées des fonctions d'influence est nulle partout sur l'intervalle de définition. Géométriquement, cela se traduit par une conservation de la forme par rotation ou changement d'échelle. Cette propriété est donc indispensable dans notre cas, puisqu'en l'absence de forces extérieures, la matière est invariante par déplacement dans l'espace.

### Positivité

**Propriété II.2** Une spline est dite positive si et seulement si toutes ses fonctions d'influence le sont :

$$\forall i \in \mathbb{N}/0 < i \leq n, \forall u \in [0, 1], b_i(u) \geq 0 \quad (\text{II.3})$$

Associée à la propriété de normalité, la positivité garantit que la courbe spline est entièrement comprise dans l'enveloppe convexe de ses points de contrôle. Ceci est très utile pour obtenir facilement des volumes englobants, par exemple dans le cadre de la détection de collision.

### Régularité

**Propriété II.3** Une spline est dite régulière si et seulement si

$$\forall i \in \mathbb{N}/0 < i \leq n, \quad \exists M_i \in [0, 1] / \begin{cases} \forall u < M_i \begin{cases} b'_i(u) \geq 0 \\ \forall j \in \mathbb{N}/i \leq j < n, b_j(u) \leq b_i(u) \end{cases} \\ \forall t > M_i \begin{cases} b'_i(u) \leq 0 \\ \forall j \in \mathbb{N}/0 \leq j < i, b_j(u) \leq b_i(u) \end{cases} \end{cases} \quad (\text{II.4})$$

Cela signifie que ses fonctions d'influence possèdent un maximum unique et se croisent toutes entre leurs maximums respectifs. On peut montrer que cette propriété assure un bon comportement vis-à-vis des oscillations parasites (Far88). Elle permet d'avoir une bonne idée de l'allure générale de la courbe. Elle est également cohérente avec le comportement de la matière : l'influence d'un atome est maximale sur ses voisins directs et décroît avec l'éloignement. Une spline non régulière ne peut donc être utilisée pour modéliser un objet réel.

### Localité

**Propriété II.4** Une spline est dite locale si ses fonctions d'influence vérifient :

$$\forall i \in \mathbb{N} / 0 < i \leq n, \exists (a_i, b_i) \in [0, 1]^2, a_i < b_i / \begin{cases} \forall u \leq a_i, b_i(u) = 0 \\ \forall u \geq b_i, b_i(u) = 0 \end{cases} \quad (\text{II.5})$$

La localité implique que les fonctions d'influence soient définies par morceaux, et par conséquent la spline aussi. La propriété de localité est essentielle pour espérer un modèle déformable 1D interactif à partir de spline. En effet, elle permet de diminuer considérablement le temps nécessaire à résoudre le système d'équation linéaire en donnant une structure bande à la matrice. Néanmoins, la continuité paramétrique de la spline n'est plus garantie par celle des fonctions d'influence, d'où la notion de continuité géométrique, détaillée dans la thèse de Grisoni (Gri99). Notons que les équations physiques qui régissent une spline matérielle nécessitent une continuité au minimum  $C^1$ . L'étude d'une poutre encastree en effort de flexion démontre à posteriori (section b) que l'on peut utiliser une spline de continuité  $C^1$  tout en obtenant des résultats très précis.

### Interpolation

**Propriété II.5** Une spline est dite d'interpolation si et seulement si elle vérifie :

$$\exists u_i \in [0, 1] / \begin{cases} b_i(u_i) = 1 \\ \forall j \neq i, b_j(u_i) = 0 \end{cases} \quad (\text{II.6})$$

Si la spline ne vérifie pas cette propriété, elle est dite d'approximation.

Les splines d'interpolation regroupent entre autres les splines cardinales, les splines de Catmull-Rom (CR74) et les splines TCB (KB84). Parmi les splines d'approximation les plus connues, on citera les splines de Bézier (Béz66b, Béz66a, Béz77), les B-splines (Rie73, FvDFH89), les  $\beta$ -splines (Bar81, BB83) et les NURBS (Pie91, PT97). Nous allons maintenant donner quelques éléments sur ces modèles, afin de déterminer celles qui correspondent le mieux à une représentation mécanique.

#### b Splines d'interpolation

**Polynômes de Hermite** Les polynômes de Hermite sont couramment utilisés en analyse. Ils permettent d'obtenir une continuité  $C^n$  à partir d'une base de  $2n + 2$  polynômes de degré  $2n + 1$ , à l'aide de conditions jusqu'au dérivées  $n^{\text{ièmes}}$ . Dans le cas d'une continuité  $C^1$ , on cherche donc 4 polynômes de degré 3 notés  $H_0, H_1, H_2$  et  $H_3$  avec les conditions suivantes :

	q(0)	q(1)	q'(0)	q'(1)
$H_0(t)$	1	0	0	0
$H_1(t)$	0	1	0	0
$H_2(t)$	0	0	1	0
$H_3(t)$	0	0	0	1

Ceci nous donne un système de quatre équations linéaires à quatre inconnues, déterminant les quatre polynômes de Hermite sur  $[0, 1]$  :

$$\begin{aligned} H_0(u) &= 1 - 3u^2 + 2u^3 \\ H_1(u) &= 3u^2 - 2u^3 \\ H_2(u) &= u - 2u^2 + u^3 \\ H_3(u) &= -u^2 + u^3 \end{aligned}$$

Ces polynômes sont couramment utilisés comme fonctions d'interpolation dans la méthode des éléments finis, pour lier points et orientations des éléments du maillage (Prio3; BPO5). Nous verrons plus loin que l'*interpolation de Hermite* n'est pas adaptée à la simulation mécanique de splines. En effet, elle nécessite de considérer comme degrés de liberté distincts les déplacements et leur dérivée par rapport à l'abscisse de la spline. De plus,  $2(m - 1)$  points de contrôle (position et orientation) sont nécessaires pour simuler  $m$  segments, alors que les méthodes se fondant uniquement sur les positions n'en nécessitent que  $m + 3$ . Son surcoût et sa complexité nous réorientent donc sur d'autres méthodes. A partir de ces polynômes, on peut néanmoins construire de nombreux types de splines, en spécifiant les tangentes en fonction des positions de contrôle de la spline. On peut citer les splines cardinales, les splines de Catmull-Rom, celles de Kochanek-Bartels ou encore les splines monotones cubiques (FC80).

**Splines cardinales** Une *spline cardinale* (CR74) est une spline cubique d'interpolation très fréquemment utilisée, telle que la tangente en chaque point de contrôle  $\mathbf{q}_i$  est calculée en utilisant le point précédent et le point suivant. Elle est égale à

$$\tau (\mathbf{q}_{i+1} - \mathbf{q}_{i-1}) \quad (\text{II.7})$$

$\tau$  étant un scalaire désignant la *tension*. On peut facilement en déduire les fonctions d'interpolation :

$$\begin{aligned} b_0(u) &= \tau (1 - 3u^2 + 2u^3) \\ b_1(u) &= 1 + (\tau - 3)u^2 + (2 - \tau)u^3 \\ b_2(u) &= \tau u + (3 - 2\tau)u^2 + (\tau - 2)u^3 \\ b_3(u) &= \tau (-u^2 + u^3) \end{aligned}$$

La tension  $\tau$  est un paramètre purement géométrique qui contrôle l'inflexion de la courbe au niveau des points de contrôle. La valeur de la tension  $\tau$  généralement utilisée est positive, une valeur négative provoquant un bouclage. Pour une valeur nulle, on retrouve une interpolation linéaire avec des segments de droite. Une valeur souvent prise est  $\frac{1}{2}$ , à laquelle on associe le nom de *spline de Catmull-Rom*. Les splines cardinales ont une continuité  $C^1$ , la propriété de localité mais pas de positivité. Une description simple et efficace est donnée par Twigg (Twio3).

**Splines de Kochanek-Bartels** Les *KB-Splines* (KB84) comportent deux paramètres en plus de la *tension*  $t$ , qui spécifie la longueur du vecteur tangent : un paramètre de *continuité*  $c$ , qui impose un changement d'amplitude et de direction du vecteur tangent, et un paramètre de *bias*  $b$ , qui définit la direction du vecteur tangent au niveau des points de contrôle. Si  $t = c = b = 0$ , on retrouve les splines cardinales de tension  $\tau$  égale à l'unité. Ces paramètres ne sont pas physiques mais géométriques. Ils correspondent à des effets que l'on veut donner dans le suivi de la trajectoire d'un objet, par exemple une caméra.

### c Splines d'approximation

**Splines de Bézier** La famille des courbes de Bézier (Béz66b, Béz66a, Béz77) a été la première utilisée en infographie.

**Définition II.2** Une spline de Bézier est définie de la manière suivante :

$$\forall u \in [0, 1], \mathbf{q}(u) = \sum_{i=0}^{n-1} B_i^{n-1}(u) \mathbf{q}_i \quad (\text{II.8})$$

$B_i^n$  désignant les polynômes de Bernstein de degré  $n$  :

$$B_i^n = \frac{n!}{(n-i)!i!} u^i (1-u)^{n-i} \quad (\text{II.9})$$

Cette spline d'approximation vérifie les propriétés de normalité, de régularité et de continuité  $C^\infty$ . Elle ne possède par contre pas la propriété de localité et le degré des fonctions d'influence est égal au nombre de points de contrôle moins un. Cela rend donc son temps de calcul prohibitif pour du temps réel. Pour pallier ce problème, les courbes de Bézier par morceaux ont été proposées, mais elles sont de continuité  $C^0$ , ce qui ne les rend pas plus utilisables pour de la simulation physique.

**B-Spline** Le modèle de courbes *B-Splines* (Cox72; dB72) est celui qui a fait l'objet du plus grand nombre d'études théoriques et de mise en oeuvre pratique.

**Définition II.3** Soit  $T = t_0 \dots t_i \dots t_{m-1}$  un vecteur nodal ou vecteur de noeuds de taille  $T$ , séquence de valeurs réelles appelées noeuds, classée par ordre croissant ; une B-Spline est une spline de degré  $p = m - n$ , dont les  $n$  fonctions d'influence  $b_i = N_{i,p}$  sont définies par :

$$\begin{cases} N_{i,0}(t) = \begin{cases} 1 & \text{si } t_i \leq t \leq t_{i+1} \\ 0 & \text{sinon} \end{cases} \\ \forall p \in \mathbb{N}^*, N_{i,p}(t) = \frac{t-t_i}{t_{i+p}-t_i} N_{i,p-1}(t) + \frac{t_{i+p+1}-t}{t_{i+p+1}-t_{i+1}} N_{i+1,p-1}(t) \end{cases} \quad (\text{II.10})$$

Si l'une des fractions a un dénominateur nul, elle est considérée comme nulle.

La dérivée des fonctions d'influence peut être évaluée également par une relation de récurrence, à partir des fonctions d'influence (Far (Far88))

$$\forall p \in \mathbb{N}^*, N'_{i,p}(t) = \frac{p}{t_{i+p}-t_i} N_{i,p-1}(t) - \frac{p}{t_{i+p+1}-t_{i+1}} N_{i+1,p-1}(t) \quad (\text{II.11})$$

L'utilisation conjointe des équations II.10 et II.11 permet de déduire des relations semblables pour les dérivées successives.

La *multiplicité*  $m$  d'un noeud correspond au nombre de fois où la valeur du noeud se retrouve dans le vecteur nodal. Il existe un processus dit d'*insertion de noeud* qui subdivise de manière exacte une B-Spline, pour y rajouter des points de contrôle. L'*algorithme d'Oslo* (Mey91) permet de réaliser cette opération en un temps linéaire. L'*algorithme de Böhm* (Boe80) existe également mais il est moins efficace si plusieurs points sont insérés en même temps. En un noeud  $t_i$  de multiplicité  $m_i$ , une B-Spline dont les fonctions d'influence sont de degré  $p$  présente une continuité de degré  $p - m_i$ . Ajouter un noeud de multiplicité supérieure ou égale à  $p + 1$  au vecteur nodal introduit donc une discontinuité dans la représentation géométrique. Lenoir (LGMCo5) utilise cette propriété pour modéliser la rupture de fils. Enlever un noeud est également possible, mais seulement quand deux segments de courbes adjacentes sont  $C_r$  avec  $r > p - m$  (EH95; Til92). Ce n'est donc généralement pas le cas.

Les B-Splines possèdent également les propriétés de positivité, de régularité et de localité.

Enfin, le processus de Schoenberg permet de trouver la B-Spline passant par un nombre donné de points. Des techniques très efficaces de résolution directe permettent d'aboutir à ce résultat (BBB87; PT97).

Quand les noeuds sont équidistants, c'est-à-dire que la multiplicité de tous les noeuds est égale à 1 et que la distance entre noeuds est constante, les B-splines sont dites *uniformes*.

Les fonctions d'influence peuvent alors facilement être précalculées. Pour une *B-Spline uniforme cubique* (BBB87) dont les noeuds sont distants de 1, elles sont égales à :

$$b_0(u) = \frac{1}{6} (1 - 3u + 3u^2 - u^3) \quad (\text{II.12})$$

$$b_1(u) = \frac{1}{6} (4 - 6u^2 + 3u^3) \quad (\text{II.13})$$

$$b_2(u) = \frac{1}{6} (1 + 3u + 3u^2 - 3u^3) \quad (\text{II.14})$$

$$b_3(u) = \frac{1}{6} u^3 \quad (\text{II.15})$$

Dans le cas où le vecteur de noeuds est quelconque, les B-splines sont dites *non uniformes*. Si les fonctions d'influence sont en plus rationnelles, on les appelle en anglais NURBS. C'est leur forme la plus utilisée.

#### d Splines précédemment utilisées pour la mécanique

D'un point de vue mécanique, les splines ont déjà fait l'objet de quelques travaux. Les splines de Catmull-Rom sont utilisées pour animer des splines par Nourrit (Nou99), pour leur simplicité de manipulation lors de la manipulation de trajectoire, sans toutefois établir des critères de choix de splines pour la modélisation mécanique. Nocent (NR01) s'abstrait complètement du problème du choix de spline en considérant que la mécanique s'applique à toute spline, puisque sa définition (II.1) est entièrement indépendante de la nature de ses fonctions de mélange et que la mécanique utilisée ne se base sur aucune propriété particulière.

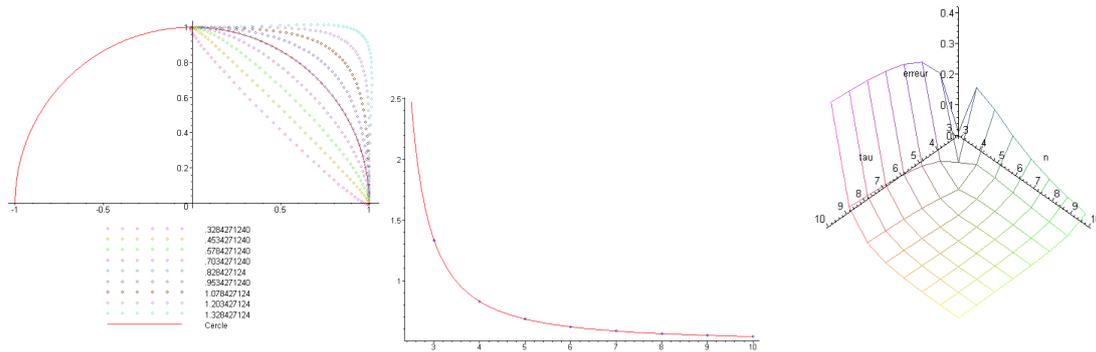
Lenoir a également choisi les splines de Catmull-Rom pour leur propriété d'interpolation, ainsi que les splines uniformes cubiques pour leur meilleure continuité (LMGC02), puis les NURBS (LGM<sup>+</sup>04). C'est avec les techniques d'insertion de noeuds que le choix des B-Splines apparaît vraiment justifié (LGMCo5). En effet, un nombre limité de noeuds ne permet pas d'atteindre la courbure nécessaire pour modéliser certaines configurations. L'insertion de noeuds supplémentaires pallie ce problème et n'est rendue possible de manière exacte qu'avec les B-Splines. Lenoir utilise également cette technique pour séparer la courbe spline en deux parties distinctes, modélisant ainsi un comportement de rupture. Par contre, il n'existe malheureusement pas de méthode pour supprimer de manière exacte un point de contrôle, quelle que soit la configuration de la spline. Deux types de splines nous semblent potentiellement adaptées :

- les *splines cardinales*, pour leur propriété d'interpolation et leur simplicité d'utilisation.
- les *B-Splines*, pour leurs propriétés de positivité et d'insertion de noeuds sans modifier la géométrie (LGMCo5), ce qui nous permettra de modéliser le phénomène de rupture. Ce sont des splines d'approximation mais des algorithmes existent pour interpoler des séries de points avec une B-Spline.

Nous allons maintenant évaluer la pertinence et les limitations de la représentation par spline dans le cadre mécanique.

#### e Choix du modèle spline pour la mécanique

Les splines ont été pensées comme des outils géométriques. Il serait naïf de penser que n'importe quelle configuration mécanique peut être atteinte avec n'importe quel jeu de paramètres et de points de contrôle, et que le type de spline choisi n'a pas d'influence. Nous proposons ici une méthode pour, à partir d'un nombre de points de contrôle, déterminer les paramètres optimaux d'une spline et évaluer la précision mécanique résultante à une configuration donnée. Pour ce faire, nous allons considérer une poutre fermée, homogène



(a) Effet de la variation de  $\tau$  sur la spline. (b) Valeur idéale de  $\tau$  en fonction de  $n$ . (c) Ecart en fonction de  $\tau$  et  $n$ .

et isotrope, à section constante, qui n'est soumise à aucune force extérieure. Le principe de Saint Venant (section I.1, page 23) permet d'établir que pour chaque section, la force et le moment sont égaux de part et d'autre. Comme la poutre est fermée, on peut en déduire que la tension et le moment flexion sont constantes tout le long de la ligne moyenne. La courbure étant proportionnelle au moment de flexion (équation I.34), elle l'est également. La ligne moyenne d'une poutre fermée, en l'absence de forces et de moments extérieurs, prend donc la forme d'un cercle. Cette configuration doit également pouvoir être reproduite avec la spline choisie, ou le cas échéant, s'en approcher le plus possible.

L'idée est de déterminer, pour un nombre donné de segments de même longueur, les paramètres nécessaires à l'obtention d'un cercle, si la spline en comporte, et la précision obtenue.

**Définition II.4** Nous appelons spline circulaire de densité  $n$ , la spline dont les  $n$  segments bouclés et les paramètres associés donnent la meilleure approximation d'un cercle.

Remarquons que tout point de la spline ainsi déterminée doit tendre vers son cercle osculateur. Le nombre de segments  $n$  est lié à leur longueur  $l_s$  par le rayon  $R$  du cercle :

$$n = \frac{2\pi R}{l_s} \quad (\text{II.16})$$

Le cercle ne peut être obtenu de manière exacte qu'avec des NURBS, mais d'autres splines comme les cardinales ou les B-Splines permettent d'obtenir des résultats très proches. Cette définition peut être étendue à une longueur quelconque de segment. Le nombre de segments devient alors une quantité paramétrique  $n(u)$  :

$$n(u) = \frac{2\pi}{k(u)\|\mathbf{r}'(u)\|} = \frac{2\pi\|\mathbf{r}'(u)\|}{\|\mathbf{r}'(u) \times \mathbf{r}''(u)\|} \quad (\text{II.17})$$

Plus  $n(u)$  est important, moins le pouvoir de représentation de la spline est exploité, mais plus la spline circulaire se rapproche d'un cercle. Nous allons maintenant appliquer ce critère sur les deux types de splines que nous avons retenus : la spline cardinale et la B-Spline.

**Spline cardinale** La spline cardinale est une spline d'interpolation cubique, avec un paramètre de tension, définissant la norme de la tangente au niveau des points de contrôle. C'est au milieu des segments que la valeur du paramètre a le plus d'influence et que la spline s'éloigne le plus du cercle (figure II.1(a)). Déterminons la valeur de tension  $\tau$  en fonction de  $n(u)$ , qui donnera la spline circulaire.

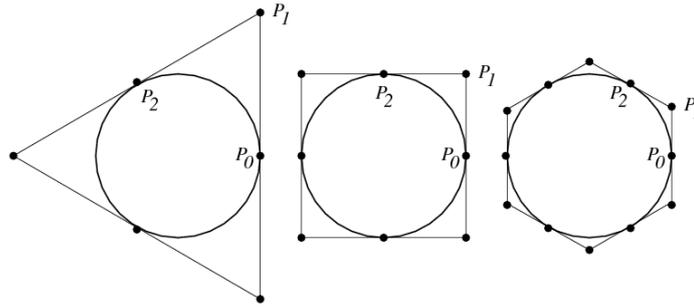


Fig. II.1 – Cercles à partir de NURBS, (BS96).

Nous considérons ici que le premier segment commence au point de coordonnées  $(R, 0)$  et nous nous restreignons à son étude. En effet, les segments suivants seront identiques par rotation le long du périmètre du cercle. Les coordonnées des points de contrôle correspondants sont les suivantes :

$$\left(R\cos\frac{2\pi}{n}, -R\sin\frac{2\pi}{n}\right), (R, 0), \left(R\cos\frac{2\pi}{n}, R\sin\frac{2\pi}{n}\right), \left(R\cos\frac{4\pi}{n}, R\sin\frac{4\pi}{n}\right) \quad (\text{II.18})$$

Au milieu du segment, la position de la spline doit être égale à  $(R\cos\frac{\pi}{n}, R\sin\frac{\pi}{n})$ . A partir de la définition II.1 de la spline, nous obtenons ensuite deux équations linéairement dépendantes de la tension  $\tau$ . La résolution de l'une d'entre elles nous donne la valeur de  $\tau$  en fonction de  $n(u)$  :

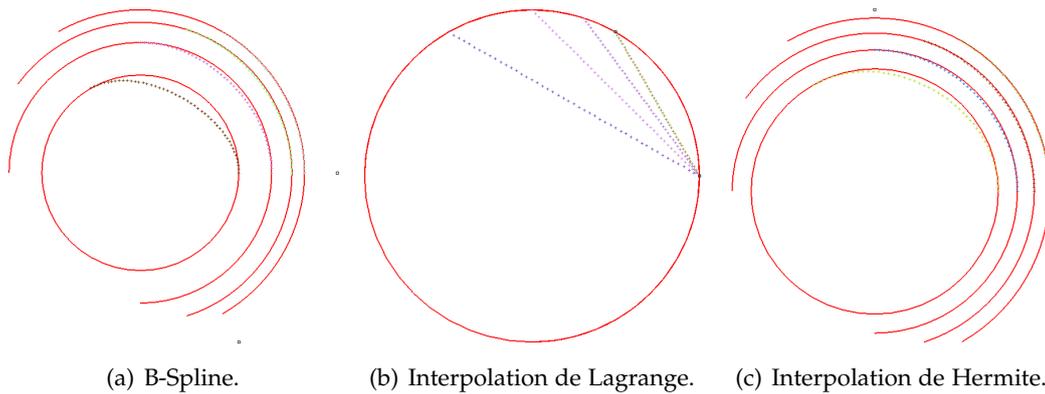
$$\tau_n = 4 \frac{1 + \cos\frac{2\pi}{n(u)} - 2\cos\frac{\pi}{n(u)}}{\cos\frac{4\pi}{n(u)} - 1} \quad (\text{II.19})$$

Notons qu'elle ne dépend que du nombre de segments  $n(u)$  mais pas du rayon  $R$  du cercle. La figure II.1(b) représente les valeurs de  $\tau_n$  sur  $[2.5, 14]$ . Quand  $n$  tend vers l'infini, la courbe spline tend vers une droite et  $\tau_n$  tend vers 0.5, valeur classiquement utilisée pour les splines cardinales. Cette valeur est donc celle qui donnera les meilleurs résultats pour une configuration proche d'un état rectiligne. Si l'on veut modéliser un objet initialement courbe, la tension choisie sera alors localement fonction de la densité  $n(u)$  (équation II.17). La figure II.1(c) donne l'écart entre les splines cardinales circulaires et le cercle, selon la densité de segments  $n(u)$  et la valeur de  $\tau$  choisie : plus le rayon du courbure sera différent de celui correspondant à la tension  $\tau$  de la spline, plus l'erreur sera importante. Enfin, la diagonale correspond aux valeurs déterminées par l'équation II.19. L'erreur y est très faible, le cercle parfait ne pouvant de toute façon pas être obtenu. Le tableau suivant donne pour les premières valeurs entières de  $n(u)$  les valeurs approchées de  $\tau_n$ , ainsi que l'erreur moyenne entre la spline cardinale et le cercle unitaire.

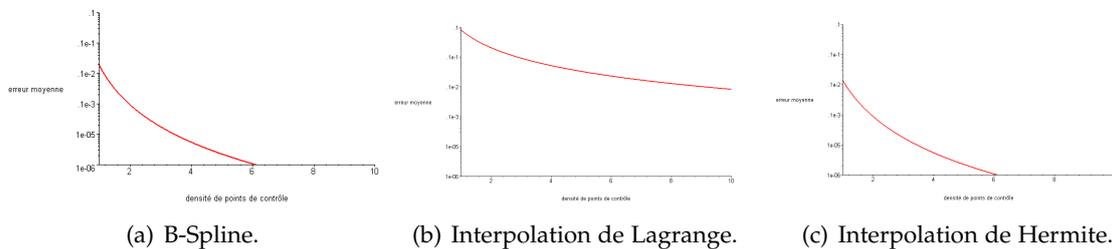
$n$	3	4	5	6	7	8	9	10
$\tau(10^{-3} \text{ près})$	1.333	0.828	0.683	0.619	0.584	0.563	0.549	0.539
$e(10^{-5} \text{ près})$	0.02257	0.07789	0.00312	0.00152	0.00083	0.00049	0.00031	0.00020

Pour une spline cardinale, il n'y a donc pas de valeur qui garantisse la précision mécanique pour n'importe quelle déformation. Néanmoins, un choix judicieux en début de simulation limite l'erreur. Il existe cependant un type de spline qui permet d'obtenir un cercle de manière parfaite : la B-Spline non uniforme rationnelle cubique (NURBS).

**NURBS** La méthode pour tracer un cercle avec une NURBS a déjà été étudiée de nombreuses fois (Ver75; Bli87; BS96). La figure II.1 donne des cercles obtenus à partir de NURBS. Néanmoins, ces cercles sont obtenus pour des poids différents : ils ne sont pas invariants si l'on effectue une permutation circulaire des points de contrôle et ne correspondent donc pas à un comportement physique. Nous allons donc nous restreindre aux B-Splines uniformes cubiques.



**Fig. II.2** – Influence de la densité de points de contrôle sur l'approximation du cercle pour le premier intervalle d'interpolation, selon la méthode choisie.



**Fig. II.3** – Ecart relatif moyen entre la spline et le cercle selon la densité de points de contrôle et la méthode choisie.

**B-Spline uniforme cubique** L'étude des B-Splines uniformes cubiques circulaires est bien plus simple que celles des splines cardinales circulaires, puisque leur fonctions d'influence ne dépendent pas d'un paramètre autre que l'abscisse. Les points de contrôles sont calculés de manière similaire, si ce n'est qu'ils sont positionnés sur le cercle qui permet d'obtenir l'approximation d'un cercle unitaire. La figure II.2(a) donne les B-Splines uniformes cubiques circulaire de densité 3,4,5,6 et 10. Notons que pour une densité de 2, la B-Spline circulaire est un diamètre du cercle.

L'erreur moyenne  $e$  entre les B-Splines uniformes cubiques et le cercle unitaire correspondant est donné figure II.3(a) et dans le tableau suivant pour les valeurs de  $n$  inférieures à 10 :

$n$	3	4	5	6	7	8	9	10
$e(10^{-5} \text{ près})$	0.07800	0.01683	0.00557	0.00236	0.00117	0.00065	0.00039	0.00025

La précision obtenue est ici légèrement moindre que pour une spline cardinale dont le paramètre  $\tau$  est optimisé, mais permet d'obtenir une précision par rapport à la densité  $n$  utilisée. Dans le cas d'un objet rigide initialement courbe, peu amené à se déformer, nous utiliserons donc une spline cardinale. Dans les autres cas, nous utiliserons des B-Splines, en adaptant le nombre de points de contrôle par unité de longueur<sup>1</sup>, pour obtenir la précision désirée.

**Interpolations utilisées en méthode des éléments finis** Les interpolations de *Lagrange* à l'ordre 1 et d'*Hermite* sont généralement utilisées en méthode des éléments finis. Nous allons rapidement évaluer leur précision, en vue de les comparer avec les splines cardinales et les B-Splines.

L'interpolation linéaire de Lagrange est la plus simple qui soit. Elle lie deux points par

<sup>1</sup>  $\frac{n(u)}{2\pi}$  correspond au nombre de points de contrôle par unité de longueur.

les polynômes suivants :

$$\begin{cases} b_0(u) = 1 - u \\ b_1(u) = u \end{cases} \quad (\text{II.20})$$

Dans le cas de deux points sur un cercle, elle revient à approximer l'arc correspondant par la corde. Le tableau suivant donne l'erreur moyenne  $e$  entre la corde et son arc, en fonction de  $n(u)$  :

$n$	3	4	5	6	7	8	9	10
$e(10^{-3} \text{ près})$	0.331	0.194	0.127	0.089	0.066	0.051	0.040	0.033

L'interpolation non-linéaire de Hermite (section 58) permet d'obtenir de bien meilleurs résultats (Gou77). L'erreur moyenne entre l'interpolation et le cercle est même légèrement meilleur que pour une B-Spline :

$n$	3	4	5	6	7	8	9	10
$e(10^{-5} \text{ près})$	0.02477	0.00810	0.00337	0.00164	0.00089	0.00052	0.00033	0.00022

Nous estimons cependant que le coût nettement inférieur d'une B-Spline en terme de points de contrôle compense largement sa précision moindre par rapport à une interpolation d'Hermite ( $n + 2$  points au lieu de  $n$  points et  $n$  tangentes de contrôle).

## 2 Matérialisation de la spline

Jusqu'à présent, nous avons vu les aspects géométriques d'une spline dans l'espace. Mais l'on peut aussi considérer cette courbe comme la ligne moyenne d'une poutre. Pour la compléter, une section doit donc lui être associée en chacun de ses points. Elle peut être de forme quelconque et sa taille varier de façon lente et progressive. La poutre possède également une masse dont il faut tenir compte. Enfin, pour pouvoir représenter la spline par un maillage de triangles, il est nécessaire de définir un repère local le long de la courbe. Cette section a pour objectif de préciser ces différents aspects, qui conditionnent la matérialisation de la spline.

### a Section de la spline matérielle

Pour modéliser la section autour de la spline, nous admettons l'hypothèse de Bernoulli : au cours de la déformation, toute section reste orthogonale à la fibre moyenne. Elle reste néanmoins libre de tourner autour de la fibre moyenne, d'un angle  $\theta$  que nous appelons *roulis*. Son origine est déterminée par l'orientation du premier vecteur de base de la section. Nous proposons de faire varier la forme et les dimensions de la section le long de la spline, mais de façon lente et progressive, puisque c'est une des conditions pour appliquer la théorie des poutres (Cou80c). Nous imposons également à toute section de posséder deux axes de symétrie, définis par les vecteurs de base  $n_1$  et  $n_2$  de la section. L'ellipse, le rectangle, le losange ont cette propriété. A chacun de ces vecteurs de base, nous associons deux *rayons principaux de section*  $R_1$  et  $R_2$ , qui achèvent de définir la section. Si la section est en plus invariante par rotation, un seul rayon  $R$  est nécessaire. C'est le cas d'une section circulaire ou carrée. On peut également considérer une section creuse, comme un tuyau ou une gaine en caoutchouc, dans lequel un matériau différent peut être inséré, par exemple un fil de cuivre, pour obtenir un câble électrique. Plusieurs sous-couches supplémentaires peuvent également être envisagées. Dans ce cas, autant de rayons supplémentaires sont nécessaires, selon la forme de la section. On désignera par  $R_1^c$  et  $R_2^c$  les rayons de sections de la  $c^{\text{ème}}$  couche et  $R^c$  s'il y a invariance par rotation de la section. La superposition de matériaux impose néanmoins qu'aucun glissement ne se fasse entre eux, pour pouvoir appliquer les résultats obtenus avec la théorie des poutres.

La géométrie de la section n'intervient dans les lois d'équilibre et de déformations que dans le calcul de son aire, de ses moments d'inertie par rapport aux axes principaux, et

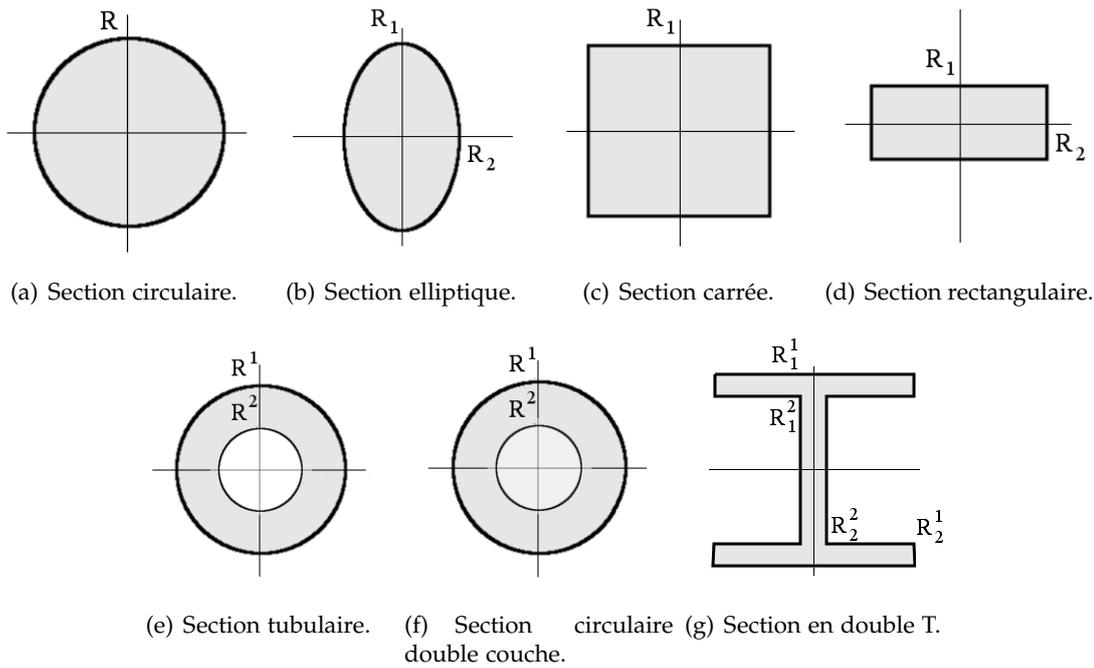


Fig. II.4 – Quelques types de section utilisables avec une spline matérielle.

de son module de rigidité à la torsion. Ces valeurs sont données pour certaines formes de section en annexe D, ainsi que dans certains ouvrages de référence (Howo1). La figure II.4 donne quelques sections définies par les rayons principaux.

## b Equation de la spline matérielle

Ces considérations sur la section nous amènent à définir une spline matérielle :

**Définition II.5** Une spline matérielle est une poutre, définie par un champ de splines  $\mathbf{p}$ , composé lui-même de deux champs :

- le champ de degrés de liberté  $\mathbf{q}$  qui contient le champ de position  $\mathbf{r} = (x, y, z)$  de la fibre moyenne, ainsi que le roulis  $\theta$  de la section.
- le champ géométrique de section  $\mathbf{s}$ , composé des rayons  $R_1^k$  et  $R_2^k$  qui définissent la géométrie de la section.

Le champ  $q$  permet de décrire les déformations de la ligne moyenne et l'orientation de la section :

$$\mathbf{q}(u, t) = (\mathbf{r}, \theta) = (x, y, z, \theta) = \sum_{i=0}^{n-1} b_i(u) \mathbf{q}_i(t) \quad (\text{II.21})$$

où les  $b_i$  désignent les  $i^{\text{èmes}}$  fonctions de base des points de contrôle  $\mathbf{q}_i$ ;  $u$  est le paramètre compris entre 0 et  $\ell$ , la longueur de la fibre moyenne. Il varie également au cours du temps  $t$  mais afin d'alléger les écritures, nous considérons le temps  $t$  de manière implicite :

$$\mathbf{q}(u) = (\mathbf{r}, \theta) = (x, y, z, \theta) = \sum_{i=0}^{n-1} b_i(u) \mathbf{q}_i \quad (\text{II.22})$$

Les dérivées  $j^{\text{èmes}}$  de  $\mathbf{q}$  par rapport à  $u$  sont données par :

$$\mathbf{q}^{(j)}(u) = \sum_{i=0}^{n-1} b_i^{(j)}(u) \mathbf{q}_i. \quad (\text{II.23})$$

La dérivée première de  $\mathbf{q}$  par rapport à  $u$  sera notée  $\mathbf{q}'$ , la dérivée seconde  $\mathbf{q}''$ , et ainsi de suite. Le champ de position  $\mathbf{r}$  et le roulis  $\theta$  adoptent les mêmes notations. Pour calculer

les valeurs du roulis  $\theta$  et des rayons de section  $R_1^k$  et  $R_2^k$  entre les points de contrôle, nous choisissons d'utiliser les mêmes fonctions d'interpolation que pour le champ de position de la spline.

L'abscisse curviligne est notée  $s$ . Les déplacements élémentaires  $ds$  et  $du$  le long de la ligne moyenne sont liés par

$$ds = \left\| \frac{dr}{du} \right\| du \quad (\text{II.24})$$

En intégrant cette relation, nous obtenons l'abscisse curviligne en fonction de l'abscisse paramétrique :

$$s = \int_0^u \left\| \frac{dr}{du} \right\| du \quad (\text{II.25})$$

Le champ géométrique de section  $\mathbf{s}$  est calculé de manière similaire à celui du champ  $\mathbf{q}$  :

$$\mathbf{s}(u) = \sum_{i=0}^{n-1} b_i(u) \mathbf{s}_i \quad (\text{II.26})$$

Par souci de simplicité, nous choisissons d'utiliser les mêmes fonctions de base pour toutes les splines du champ  $\mathbf{p}$ , mais l'on peut très bien utiliser des fonctions d'ordre différent, voire des modèles de spline différents.

### c Répartition de la masse

Les précédentes études portant sur les splines matérielles considèrent toutes des sections circulaires à rayon constant. La répartition massique n'y dépend ni du temps ni de l'abscisse paramétrique. Une masse linéique  $\mu$  peut donc être utilisée pour évaluer la masse le long de la spline. Nourrit (Nou99) l'a définie en fonction de l'abscisse paramétrique<sup>2</sup>  $u$ , ce qui n'est pas correct puisque l'abscisse curviligne  $s$  n'est pas proportionnellement liée à  $u$ . Cela conduit à une répartition massique non homogène entre les points de contrôle. Forts de cette remarque, Nocent (NR01) puis Lenoir (LMGCo2; Leno4) ont corrigé ce problème en multipliant  $\mu(u)$  par  $\|\mathbf{r}'\|$ . Remarquons que le vrai problème se situe dans l'intégration par rapport à l'abscisse paramétrique, au lieu de l'abscisse curviligne. Le passage en déplacement élémentaire paramétrique se fait tout naturellement avec la relation II.24. Multiplier la masse linéique par la norme de la dérivée paramétrique du champ de position revient toutefois à utiliser cette relation, mais devrait également être appliquée aux intégrales des déformations. Nous considérons la spline homogène sur un segment  $j$ , de masse volumique  $\rho_j$ . Pour obtenir la densité linéique  $\mu_j(u)$ , il suffit de multiplier la masse volumique  $\rho_j$  par l'aire  $A$  de la section :

$$\mu_j(u) = \rho_j A(u) \quad (\text{II.27})$$

### d Repère local, courbure et torsion

Nous avons défini la spline matérielle à l'aide du champ de splines  $\mathbf{p}$ , lui-même composé notamment du champ de section  $\mathbf{s}$ , dont les composantes dépendent des axes principaux de la section.

D'un point de vue mécanique, si la section est circulaire ou tubulaire, il n'est pas nécessaire de lui associer une base, puisque les propriétés géométriques de la section sont alors invariantes autour de la fibre moyenne. Le fait de ne pas utiliser un repère local dans le calcul des déformations rend les résultats plus précis. En effet, les repères locaux sont généralement calculés par propagation, c'est-à-dire en fonction du précédent, et introduisent donc des erreurs en terme de précision.

<sup>2</sup>Nourrit a noté l'abscisse paramétrique  $s$ , ce qui peut prêter à confusion avec la notation usuelle de l'abscisse curviligne...

Si la section n'est pas circulaire, des axes locaux sont nécessaires pour évaluer l'orientation de la section dans le plan par rapport à la fibre moyenne. Néanmoins, le premier vecteur de base étant déterminé par le champ de position  $r$ , l'imprécision est bien moindre que dans les méthodes basées uniquement sur des repères (WH04).

Indépendamment de la mécanique, nous avons besoin de visualiser l'objet de manière volumique pour pouvoir interagir avec lui. Un repère est donc nécessaire afin de construire les polygones autour de la fibre moyenne.

Le repère de Frenet est bien connu de la géométrie différentielle (Bis75; Klo86; Vel92). A partir d'une équation paramétrique comme une spline, il permet de définir de manière intuitive un repère  $\{\mathbf{t}, \mathbf{n}, \mathbf{b}\}$  associé à tout point de la courbe.

La direction de la tangente est déterminée par la dérivée  $\mathbf{r}'$  du champ de position. Le vecteur *tangent* est égal à la normalisation de  $\mathbf{r}'$  :

$$\mathbf{t} = \frac{\mathbf{r}'}{\|\mathbf{r}'\|} \quad (\text{II.28})$$

**Définition II.6** Le vecteur normal de courbure  $\mathbf{n}$ , ou vecteur normal principal, est défini par :

$$\mathbf{n} = \frac{\mathbf{t}'}{k} \quad (\text{II.29})$$

où  $k$  est un scalaire, appelé courbure, tel que  $\|\mathbf{n}\| = 1$ . Les courbes droites ont une courbure nulle.

Après quelques calculs, la courbure s'exprime en fonction de  $\mathbf{r}$  et de sa dérivée première  $\mathbf{r}'$  :

$$k = \frac{\|\mathbf{r}' \times \mathbf{r}''\|}{\|\mathbf{r}'\|^3} \quad (\text{II.30})$$

où  $\times$  désigne le produit vectoriel. Comme la norme du vecteur tangent n'est jamais nul, la courbure est toujours définie.

**Définition II.7** Le vecteur de courbure binormal  $\mathbf{b}$ , normal à  $\mathbf{t}$  et à  $\mathbf{n}$ , est défini par :

$$\mathbf{b} = \frac{\mathbf{n}' + k\mathbf{t}}{\tau} \quad (\text{II.31})$$

où  $\tau$  est un scalaire, appelé torsion, tel que  $\|\mathbf{b}\| = 1$ . Les courbes planes ont une torsion nulle.

La torsion  $\tau$  n'est pas matérielle mais géométrique. En effet, l'orientation de la section n'intervient pas dans sa formulation :

$$\tau = \frac{\mathbf{r}' \times \mathbf{r}'' \cdot \mathbf{r}'''}{\|\mathbf{r}' \times \mathbf{r}''\|^2} \quad (\text{II.32})$$

Comme l'a montré Chouaïeb (CM04), la torsion matérielle se décompose en fait en la somme de deux parties scalaires : la torsion géométrique  $\tau$ , ou torsion de Frenet, et le taux de roulis  $\theta'$  autour de la fibre moyenne. La torsion géométrique est responsable du couplage entre le flexion et la torsion, tandis que le roulis décrit la rotation du matériau autour de la fibre moyenne, comme le décrivent Boyer et Primault. (BP05; Prio3). Cette définition de la torsion est également exprimée par Grégoire *et al.* (GSo6) :

$$\tau + \theta' \quad (\text{II.33})$$

La torsion géométrique est vue comme la variation d'angle de la normale au plan osculateur (i.e le plan qui contient la courbure). Ni elle, ni le vecteur binormal  $\mathbf{b}$  ne sont donc définis quand la courbe reste localement dans le même plan. Le dénominateur de sa formulation est alors nul. Le vecteur normal change également de sens à chaque fois que la courbure s'annule.

Nous devons donc définir autrement le repère local associé à la fibre moyenne, que nous noterons  $(\mathbf{t}, \mathbf{n}_1, \mathbf{n}_2)$

Bishop (Bis75) propose une méthode alternative basée sur le *transport parallèle* plutôt que sur des dérivées locales. Il fait le constat suivant : comme  $\mathbf{t}$  est unique, n'importe quelle base  $(\mathbf{n}_1, \mathbf{n}_2)$  peut être choisie dans le plan de la section. L'idée est d'exprimer les dérivées de  $\mathbf{n}_1$  et  $\mathbf{n}_2$  uniquement en fonction de  $\mathbf{t}$ , pour ne plus avoir à se soucier de la courbure. Essentiellement solution d'une équation différentielle, ce repère est tributaire des conditions initiales (HM95).

Pour éviter le changement brusque de sens du repère de Frenet, Shani et Ballard (SB84) proposent une méthode de repère qui minimise la rotation. Le repère correspondant est donné comme solution d'équations différentielles par Klok (Klo86), mais celles-ci sont généralement difficiles à résoudre explicitement. Un repère de départ est également nécessaire. Minimiser la torsion dans la propagation du repère revient à effectuer une rotation entre deux repères successifs selon l'axe orthogonal au plan de courbure. Cela permet au premier vecteur normal  $\mathbf{n}_1$  d'avoir même direction que le vecteur normal  $\mathbf{n}$ . Pour obtenir le repère matériel, il n'y a ensuite plus qu'à effectuer une rotation autour de l'axe.

Beaucoup utilisée en robotique et en mécanique (Paio2), la *formule de Rodrigues* est définie à partir d'une rotation entre deux repères successifs  $\{\mathbf{t}^i, \mathbf{n}_1^i, \mathbf{n}_2^i\}$  et  $\{\mathbf{t}^{i+1}, \mathbf{n}_1^{i+1}, \mathbf{n}_2^{i+1}\}$ , d'angle  $\alpha = (\mathbf{t}_i, \mathbf{t}_{i+1})$  et d'axe  $\mathbf{V} = \mathbf{t}_i \times \mathbf{t}_{i+1}$ . Elle donne la matrice de rotation  $R(\alpha, \mathbf{V})$  correspondante :

$$R(\alpha, \mathbf{V}) = \cos\alpha I_3 + (1 - \cos\alpha) \mathbf{V} \otimes \mathbf{V} + \sin\alpha \hat{\mathbf{V}} \quad (\text{II.34})$$

avec  $\otimes$  désignant le produit tensoriel,  $I_3$  la matrice identité de rang 3, et  $\hat{\mathbf{V}}$  la matrice antisymétrique telle que  $\hat{\mathbf{V}} = \mathbf{V} \times$ .

Boyer (BP05) donne une expression plus pratique et efficace, sans avoir à calculer l'angle  $\alpha$ , en fonction des tangentes successives  $\mathbf{t}_i$  et  $\mathbf{t}_{i+1}$ . En remarquant que  $\cos\alpha = \mathbf{t}_{i+1} \cdot \mathbf{t}_i$  et que  $\sin\alpha \mathbf{V} = \mathbf{t}_{i+1}$ , la formule de Rodrigues devient :

$$R(\alpha, \mathbf{V}) = (\mathbf{t}_i \cdot \mathbf{t}_{i+1}) I_3 + \frac{(\mathbf{t}_i \times \mathbf{t}_{i+1}) \otimes (\mathbf{t}_i \times \mathbf{t}_{i+1})}{1 + \mathbf{t}_i \cdot \mathbf{t}_{i+1}} + \widehat{(\mathbf{t}_i \times \mathbf{t}_{i+1})} \quad (\text{II.35})$$

Cette formulation n'est pas définie pour un demi-tour entre deux repères, ce qui est facilement évitable dans le cas d'une spline cubique, mais demeure surtout relativement coûteuse.

Les quaternions constituent aujourd'hui un outil indispensable pour décrire l'orientation d'un vecteur ou d'un objet. Leur coût est faible et ils sont particulièrement adaptés à la résolution de problèmes d'orientation et de rotation en informatique graphique et en animation. La même rotation d'angle  $\alpha$  et d'axe  $\mathbf{V}$  peut être exprimée à l'aide d'un quaternion **quat**

$$\mathbf{quat} = \left( \cos\left(\frac{\alpha}{2}\right), \sin\left(\frac{\alpha}{2}\right) \mathbf{V} \right) \quad (\text{II.36})$$

Le passage du repère  $\{\mathbf{t}^i, \mathbf{n}_1^i, \mathbf{n}_2^i\}$  au suivant  $\{\mathbf{t}^{i+1}, \mathbf{n}_1^{i+1}, \mathbf{n}_2^{i+1}\}$  est toujours défini et se fait alors par simple multiplication dans l'algèbre d'Hamilton ou par utilisation de la matrice de rotation correspondante de rang 4.

Hanson et Ma (Han94) proposent une formulation élégante de repère à base de quaternions, qui englobe le repère de Frenet et le repère à base de transport parallèle. Nous vous recommandons par ailleurs la lecture de (HM95) pour une vue détaillée de ces notions.

Cependant, dans l'optique temps-réel, nous recherchons la méthode la moins coûteuse qui remplisse le critère de minimisation de rotation. Bloomenthal (Blo90) mentionne la *méthode de Kenneth Sloan* pour propager les repères. Elle définit les vecteurs normaux du repère  $i + 1$  à partir de son vecteur tangent et du premier vecteur normal précédent :

$$\begin{cases} \mathbf{n}_2^{i+1} = \mathbf{t}^{i+1} \times \mathbf{n}_1^i \\ \mathbf{n}_1^{i+1} = \mathbf{n}_2^{i+1} \times \mathbf{t}^{i+1} \end{cases} \quad (\text{II.37})$$

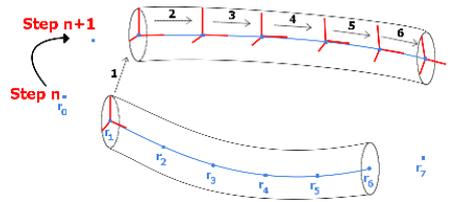


Fig. II.5

Nous utilisons donc cette méthode qui minimise la torsion du repère. Il ne nous reste plus qu'à appliquer une rotation d'axe  $\mathbf{t}^i$  et d'angle  $\theta^i$  à chaque repère, pour obtenir l'orientation correcte de la section. Bloomenthal fait remarquer que le repère suivant est non défini pour  $\mathbf{t}_{i+1} = \mathbf{n}_1^i$ . Cela n'est pas critique dans le sens où cette égalité correspond à un quart de tour (rotation de  $90^\circ$ ) entre deux repères. Cette situation est cependant improbable dans notre contexte puisqu'un tel écart entre deux repères conduirait à une grossière approximation mécanique : nous utiliserons donc une résolution suffisamment fine pour éviter ce cas. Notons également que cette méthode a été utilisée dans le contexte temps-réel par Grisoni et al. (RGF<sup>+</sup>04; FLA<sup>+</sup>05). Pour calculer le repère de référence, soit nous utilisons le vecteur de courbure s'il est défini, soit nous déterminons un vecteur de base quelconque orthogonal au vecteur tangent, puis nous en obtenons un second par produit vectoriel. Pour mettre à jour le vecteur de référence entre deux pas de temps, nous préférons cependant utiliser la formule de Rodrigues ou une rotation par quaternion (figure II.5).

Une fois les repères définis le long de la spline, nous pouvons nous intéresser à l'habillage de la spline, afin de la matérialiser dans l'espace.

### e Enveloppe géométrique de la spline matérielle

Pour générer une enveloppe à partir d'une courbe dans l'espace, il existe de nombreuses méthodes plus ou moins sophistiquées : cylindres généralisés, surfaces implicites, surfaces à convolution... Une étude approfondie a déjà été effectuée par Lenoir (Leno4), dont le choix s'est porté sur les cylindres généralisés. Nous invitons donc le lecteur à s'y référer pour se forger une opinion sur les autres méthodes.

**Définition II.8** *Un cylindre généralisé est la surface générée par le déplacement d'une section plane orientée, le long d'un chemin (Bin71; Bin87; Blo90).*

Dans le cas d'une spline matérielle, le cylindre généralisé se rapporte au contour de la section  $S$  générée par le champ  $\mathbf{s}$  et le chemin à la courbe spline décrite par le champ de position  $\mathbf{r}$ .

La surface générée est continue si la courbe est au moins de continuité  $C^1$ , ce qui est garanti pour des splines cardinales et des B-Splines cubiques.

Des problèmes d'interpénétration peuvent apparaître dans les zones de forte courbure, quand le rayon de courbure local devient plus petit que le rayon du cylindre. Ce problème ne sera pas rencontré si l'on se place dans le cadre des hypothèses de la théorie des poutres : le rayon de courbure doit être au moins 5 fois plus grand que le rayon de la section.

La technique dite du *mélange géométrique*, ou *Vertex Blending* permet de tirer parti de la rapidité des cartes graphiques (Blo02; GM03). Étant données deux matrices de transformation  $\mathbb{M}_1$  et  $\mathbb{M}_2$  associées à deux repères, elle consiste à interpoler ces transformations à l'aide d'une fonction  $\nu$  :

$$u \in [0, 1], \mathbb{M}(u) = \nu(u)\mathbb{M}_1 + (1 - \nu(u))\mathbb{M}_2 \quad (\text{II.38})$$

Il est cependant très difficile de trouver une fonction  $\nu$  qui permette d'obtenir une surface qui suive la courbe spline, quelle que soit la déformation de celle-ci.

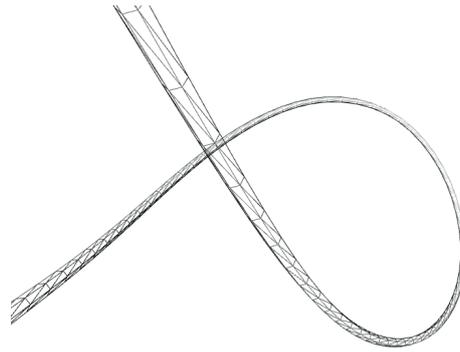


Fig. II.6 – Maillage triangulaire d'une spline matérielle.

Pour créer l'enveloppe de la spline, nous discrétisons le périmètre de la section en un nombre constant de points, positionnés de manière identique dans chaque repère local. Un maillage à base de triangles peut donc être généré, comme illustré par la figure II.6. Un nombre fixe de points appartenant au contour de la section est d'abord défini par son repère local  $(\mathbf{G}, \mathbf{n}_1, \mathbf{n}_2)$ . Ces points sont reliés entre eux, ainsi qu'à leur homologue précédent et suivant. Deux paires de voisins successifs ne sont généralement pas coplanaires. Il faut donc achever la formation du maillage par la subdivision de ces quadrilatères en deux triangles.

Maintenant que nous avons défini entièrement la géométrie de notre objet 1D, nous allons appliquer les lois de la mécanique pour simuler ses déformations. Nous allons dans un premier temps nous intéresser aux déformations réversibles, dites élastiques.

## B MODÈLE MÉCANIQUE SPLINE À DÉFORMATIONS RÉVERSIBLES

Dans le cas de la simulation d'objets 1D, la théorie des poutres fournit les bases mécaniques nécessaires pour modéliser leurs déformations. Dans cette section, nous proposons son application rigoureuse aux splines matérielles, pour obtenir un modèle géométriquement exact. Nous poserons d'abord le contexte dans lequel nous faisons évoluer la spline matérielle. Nous introduirons ensuite la loi d'équilibre la plus adaptée à la formulation d'une spline matérielle, à savoir les équations de Lagrange. Enfin, nous appliquerons les lois de constitution de la théorie des poutres et les équations de Lagrange aux splines matérielles, afin d'obtenir un premier système matriciel.

### 1 Hypothèses simplificatrices de départ

Au vu de la diversité des modèles de poutres que nous avons présentés à la section c, il convient d'abord de bien définir les hypothèses de départ, afin d'être sûrs que les simplifications effectuées par la suite sont valables. Nous souhaitons un modèle le plus général et précis possible, mais aussi des temps de calculs qui permettent le temps réel. Nous proposons un compromis qui satisfait ces contraintes, au travers de plusieurs hypothèses :

- le comportement élastique des objets que nous souhaitons simuler, tels que les fils de chirurgie, les câbles électriques, les cheveux, etc. est matériellement linéaire. Nous nous placerons donc dans le cadre des *petites déformations*. Cette hypothèse nous permet de sommer les contributions des différentes énergies déformations exprimées indépendamment les unes des autres.
- des objets tels que des câbles se déforment considérablement, contrairement à des poutres de bâtiment. Considérer que la fibre moyenne ne s'écarte pas beaucoup de sa configuration initiale conduirait vite à des résultats aberrants. Nous travaillerons donc en *grands déplacements*, c'est-à-dire sans faire d'approximation géométrique, notamment au niveau de la courbure.

- nous nous intéressons aux déformations de la fibre moyenne ; la forme de la section courante nous importe peu. Nous pouvons donc appliquer l'hypothèse de Kirchhoff ; cela nous amène à négliger le *cisaillement*.
- dans le cas d'objets tels que le fil de chirurgie, la propriété d'*élongation* intervient dans la manipulation. Nous choisissons donc d'en tenir compte ; la *flexion* et la *torsion* intervenant de manière évidente, la spline sera donc affectée par trois déformations.
- la section étant relativement fine et les déplacements peu rapides, nous choisissons de négliger les inerties de rotation dues à la flexion, mais pas en torsion. Cela nous situe quelque part entre le *modèle de Rayleigh* qui tient compte de l'énergie de rotation, et le *modèle d'Euler-Bernoulli* qui la néglige. Ce dernier présente en effet l'inconvénient d'imposer une résolution statique du roulis et nécessite par conséquent de contraindre un point de la fibre moyenne. Un fil de chirurgie pouvant être amené à évoluer seul dans l'espace ou à glisser au travers de points de suture, nous conservons l'énergie de rotation en torsion.

Maintenant que nous avons matérialisé la spline et exprimé les énergies potentielles internes de déformations en fonction de la géométrie de la spline, il ne nous manque plus que la loi de d'équilibre pour lier déformations et déplacements. Comme nous l'avons vu au chapitre dernier, les équations de Lagrange sont parfaitement adaptées à la formulation spline. Nous allons donc décrire les équations de Lagrange et leurs relations avec les splines, puis établir ensuite le système numérique à résoudre.

## 2 Equations de Lagrange

**Principe II.1** *Le principe de moindre action, énoncé par de Maupertuis en 1747, établit que la nature recherche toujours le chemin qui minimise les efforts entre deux états.*

C'est en 1788 que Lagrange propose une formulation mathématique, qui est encore en vigueur de nos jours. Il définit entièrement un système par un ensemble de paramètres indépendants, appelés *coordonnées généralisées*  $\mathbf{q}_i$ . Le vecteur d'état  $\mathbf{q}$  regroupe l'ensemble de ces  $n$  paramètres. De ces paramètres dépend un terme énergétique, le lagrangien  $\mathcal{L}$ , différence de l'énergie cinétique  $T$  et de l'énergie potentielle  $U$  du système.

$$\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) = T(\mathbf{q}, \dot{\mathbf{q}}) - U(\mathbf{q}, \dot{\mathbf{q}}) \quad (\text{II.39})$$

où  $\dot{\mathbf{q}}$  désigne la dérivée temporelle de  $\mathbf{q}$  par rapport au temps. L'action  $\mathcal{S}$  étant définie comme l'intégrale du Lagrangien entre deux états  $A$  et  $B$ , le principe de moindre action s'énonce de la manière suivante<sup>3</sup> :

$$\delta \mathcal{S}(\mathbf{q}, \dot{\mathbf{q}}) = \delta \int_{t_A}^{t_B} \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) dt = 0 \quad (\text{II.40})$$

Lagrange a démontré que ce principe est équivalent à un système d'équations différentielles dépendant des coordonnées généralisées :

$$\forall i \in \{1, \dots, n\}, \frac{\partial \mathcal{L}}{\partial \mathbf{q}_i} - \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}_i} = 0 \quad (\text{II.41})$$

Ce système est appelé *équations de Lagrange*, ou *équations d'Euler-Lagrange*.

Comme les auteurs précédents de travaux portant sur les splines matérielles (Nou99; Noco1; Leno4), nous adoptons la *représentation lagrangienne* : cela nous permet d'associer les coordonnées généralisées au champ de degrés de liberté de la spline. L'énergie cinétique ne dépend alors que de  $\dot{\mathbf{q}}$ , et l'énergie potentielle que de  $\mathbf{q}$ . Le lagrangien devient donc :

$$\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) = T(\dot{\mathbf{q}}) - U(\mathbf{q}) \quad (\text{II.42})$$

<sup>3</sup>On parle ici de calcul variationnel puisqu'on ne se base pas sur l'action elle-même mais sur ses variations.

Les équations de Lagrange se réduisent à :

$$\forall i \in \{1, \dots, n\}, \frac{d}{dt} \left( \frac{\partial T}{\partial \dot{\mathbf{q}}_i} \right) + \frac{\partial U}{\partial \mathbf{q}_i} = 0 \quad (\text{II.43})$$

Cette formulation n'est valable que dans le cas où les efforts extérieurs  $\mathbf{F}_i$  sont nuls. Dans le cas contraire, le système devient :

$$\forall i \in \{1, \dots, n\}, \frac{d}{dt} \left( \frac{\partial T}{\partial \dot{\mathbf{q}}_i} \right) = \mathbf{F}^i - \frac{\partial U}{\partial \mathbf{q}_i} \quad (\text{II.44})$$

Ce système d'équations constitue la *loi d'équilibre* d'une spline matérielle. Dans les sous-sections suivantes, nous allons successivement exprimer le terme correspondant à l'énergie potentielle de déformation, puis celui à l'énergie cinétique, par rapport aux degrés de libertés du système. Nous aurons alors obtenu un premier système que l'on pourra résoudre numériquement.

### 3 Forces et moments généralisés de déformation d'une spline matérielle

En remaniant les densités d'énergie potentielles de déformation établies dans la section 2 pour la traction-compression, la torsion et la flexion, nous allons exprimer le terme  $\frac{\partial U}{\partial \mathbf{q}_i}$  en fonction du champ de degrés de liberté  $\mathbf{q}$  de la spline matérielle.

Nous choisissons de définir un vecteur de déformation  $\varepsilon$  :

$$\varepsilon = \begin{pmatrix} \epsilon \\ \tau + \theta'k \end{pmatrix} \quad (\text{II.45})$$

L'état initial est lui noté  $\varepsilon_0$ . Nous construisons également la *matrice de Hooke*  $\mathbb{H}$ , homogène à une raideur :

$$\mathbb{H} = \begin{pmatrix} EA(u) & 0 & 0 \\ 0 & GJ(u) & 0 \\ 0 & 0 & EI(u) \end{pmatrix} \quad (\text{II.46})$$

A partir des équations des densités d'énergie de déformation I.28, I.36 et I.45, nous pouvons maintenant exprimer la densité d'énergie sous forme de matrices :

$$W = (\varepsilon - \varepsilon^0)^t \mathbb{H} (\varepsilon - \varepsilon^0) \quad (\text{II.47})$$

L'énergie potentielle de déformation de la spline matérielle s'obtient en intégrant la densité sur toute sa longueur  $L$  :

$$U = \frac{1}{2} \int_0^L (\varepsilon - \varepsilon^0)^t \mathbb{H} (\varepsilon - \varepsilon^0) ds \quad (\text{II.48})$$

Le terme correspondant à l'énergie potentielle de déformation dans les équations de Lagrange II.43 s'obtient en dérivant  $U$  par rapport aux coordonnées généralisées  $\mathbf{q}_i$ . L'intégrale et la dérivée partielle dépendant de paramètres différents, elle peuvent être commutées :

$$\mathbf{P}^i = - \frac{\partial U}{\partial \mathbf{q}_i} = - \frac{1}{2} \int_0^L \frac{\partial (\varepsilon - \varepsilon^0)^t \mathbb{H} (\varepsilon - \varepsilon^0)}{\partial \mathbf{q}_i} ds \quad (\text{II.49})$$

où  $\mathbf{P}$  est un vecteur non homogène composé de forces et de moments généralisés : la force généralisée de traction-compression  $\mathbf{P}_s$ , le couple généralisé de torsion  $\mathbf{P}_t$  et le moment généralisé de flexion  $\mathbf{P}_b$ .

Pour résoudre les équations de Lagrange, nous allons devoir exprimer ces forces et moments généralisés en fonction du champ de degrés de liberté  $\mathbf{q}$  - composé du champ de position  $\mathbf{r}$  et du roulis  $\theta$  - et de ses dérivées  $\mathbf{q}', \mathbf{q}'', \mathbf{q}'''$ , ainsi que des fonctions de base  $b_i$  de

la spline et de ses dérivées  $b'_i, b''_i, b'''_i$ . Cela nous permet d'évaluer les intégrales par exemple à l'aide de sommes de Riemann (PFTV88b). Nous nous plaçons dans le cadre des grands déplacements, nous ne faisons donc aucune approximation au niveau de la géométrie. Les expressions des forces et moments généralisés sont par conséquent assez compliquées. Nous avons dans un premier temps effectué les calculs à la main, puis les avons vérifiées à l'aide du logiciel mathématique *Maple*. Pour plus de clarté, nous ne donnons pas les détails de ces calculs, mais uniquement les résultats. De plus, nous considérons séparément les dérivations par rapport au champ de position  $\mathbf{r}$  et au roulis  $\theta$ . Leurs dérivées par rapport aux coordonnées généralisées  $\mathbf{r}_i$  et  $\theta_i$  sont égales à :

$$\frac{\partial \mathbf{r}^{(j)}}{\partial \mathbf{r}_i} = b_i^{(j)} \quad (\text{II.50})$$

$$\frac{\partial \theta^{(j)}}{\partial \theta_i} = b_i^{(j)} \quad (\text{II.51})$$

où  $b_i^{(j)}$  est la dérivée  $j^{\text{ème}}$  de  $b_i$  par rapport à  $u$ . Enfin, nous introduisons les variables suivantes pour réduire de manière sensible la taille des expressions :

$$\mathbf{C} = \mathbf{r}' \times \mathbf{r}'' \quad (\text{II.52})$$

$$\mathbf{P}^i = \frac{\partial \mathbf{C}}{\partial \mathbf{r}_i} \quad (\text{II.53})$$

$$\mathbf{R}^i = \mathbf{C} b_i''' - \mathbf{P}^i \times \mathbf{r}''' \quad (\text{II.54})$$

$$\mathbf{G}^i = \mathbf{C} \times \mathbf{P}^i \quad (\text{II.55})$$

Pour calculer de manière optimisée les forces et moments généralisés, nous calculons une seule fois ces expressions au début de chaque pas de temps.

### a Force généralisée de traction-compression

En petites déformations, la déformation en traction-compression est définie par

$$\epsilon_s = 1 - \|\mathbf{r}'\| \quad (\text{II.56})$$

Après dérivation, la force généralisée de traction-compression  $P_s^i$  devient :

$$P_s^i(\mathbf{r}) = -E \int_0^L A(u) \left( 1 - \frac{\|\mathbf{r}'_0\|}{\|\mathbf{r}'\|} \right) \mathbf{r}' b'_i ds \quad (\text{II.57})$$

Puisque la traction-compression ne dépend pas de  $\theta$ , la composante selon  $\theta$  de la force généralisée est nulle :

$$P_s^i(\theta) = 0 \quad (\text{II.58})$$

Noçant (NR01) et (Leno4) se sont placés dans le cadre des grandes déformations, c'est pourquoi ils aboutissent à une expression différente de la nôtre.

Pour tenir compte de la déformation transverse associée à la traction-compression, nous proposons de mettre à jour les rayons à la fin de chaque pas de temps, à l'aide de l'équation I.29 :

$$R^{n+1} = R^n \left( 1 - \nu \frac{\Delta L}{L} \right) \quad (\text{II.59})$$

La variation temporelle des rayons de section étant lente et progressive, nous choisissons de ne pas en tenir compte dans la loi d'équilibre, et donc de considérer nulles les dérivées par rapport au temps.

### b Moment généralisé de torsion

Comme nous l'avons expliqué précédemment (voir section d), la torsion s'exprime à partir de la torsion géométrique  $\tau$  et du taux de roulis  $\theta'$  :

$$\epsilon_t = \theta' + \tau \quad (\text{II.60})$$

avec la torsion géométrique  $\tau$  ayant pour expression :

$$\tau = \frac{\mathbf{r}' \times \mathbf{r}'' \cdot \mathbf{r}'''}{\|\mathbf{r}' \times \mathbf{r}''\|^2} = \frac{\mathbf{C} \cdot \mathbf{r}'''}{\|\mathbf{C}\|^2} \quad (\text{II.61})$$

La contribution géométrique au moment généralisé de torsion  $P_t^i$  s'exprime :

$$P_t^i(\mathbf{r}) = -\frac{G}{2} \int_0^L J(u) \epsilon_t F^i ds \quad (\text{II.62})$$

La contribution du roulis s'écrit elle :

$$P_t^i(\theta) = -\frac{G}{2} \int_0^L J(u) \epsilon_t \frac{b_i'}{\|\mathbf{r}'\|} ds \quad (\text{II.63})$$

### c Moment généralisé de flexion

La flexion est fonction de la courbure de Frenet  $k$  :

$$\epsilon_b = k = \frac{\|\mathbf{r}' \times \mathbf{r}''\|}{\|\mathbf{r}'\|^3} = \frac{\|\mathbf{C}\|}{\|\mathbf{r}'\|^3} \quad (\text{II.64})$$

Dans le cas d'une section circulaire, l'impact de la courbure sur la flexion ne dépend pas de son orientation dans la base locale de la section. Le moment généralisé de flexion  $P_b^i$  est alors égal à :

$$P_b^i(\mathbf{r}) = -\frac{E}{4} \int_0^L I(u) \left( \frac{G^i}{\|\mathbf{r}'\|^6} - 3 \frac{k^2 b_i' \mathbf{r}'}{\|\mathbf{r}'\|^2} \right) ds \quad (\text{II.65})$$

Dans le cas contraire, il faut décomposer le moment généralisé de flexion local  $p_b^i$  dans la base de la section. En posant :

$$p_b^i = \frac{G^i}{\|\mathbf{r}'\|^6} - 3 \frac{k^2 b_i' \mathbf{r}'}{\|\mathbf{r}'\|^2} \quad (\text{II.66})$$

le moment généralisé de flexion pour une section non circulaire devient :

$$P_b^i(\mathbf{r}) = -\frac{E}{4} \int_0^L \left( I_1(u) \cdot p_{b1}^i + I_2(u) \cdot p_{b2}^i \right) ds \quad (\text{II.67})$$

$I_1(u)$  et  $I_2(u)$  désignant les moments d'inerties par rapport aux axes  $n_1$  et  $n_2$ , et  $p_{b1}^i$  et  $p_{b2}^i$  les projetés de  $p_b^i$  sur les axes  $n_1$  et  $n_2$ .

Comme l'énergie de flexion  $U_b$  ne dépend pas de  $\theta$ , le moment généralisé de flexion selon  $\theta$  est nul :

$$P_b^i(\theta) = 0 \quad (\text{II.68})$$

Lenoir (Leno4) a approximé la courbure  $k$  par la norme de la dérivée seconde du champ de courbure,  $\|\mathbf{r}'\|$ , en se plaçant dans le cadre des petits déplacements. Cela introduit une erreur trop importante dès que l'on s'écarte de la configuration initiale. D'un point de vue visuel et haptique, les résultats semblent plausibles.

Si nous considérons un *équilibre quasi-statique*, toutes les forces et tous les moments généralisés que nous venons de voir suffisent à résoudre numériquement les équations de Lagrange (II.43) ; les termes cinétiques, dépendant du temps, sont nuls. Le système à résoudre se présente donc sous la forme :

$$\mathbf{F} + \mathbf{P} = \mathbf{0} \quad (\text{II.69})$$

Ce résultat est équivalent au principe fondamentale de la statique. Pour résoudre l'*équilibre dynamique*, nous devons calculer en plus le terme cinétique des équations de Lagrange.

#### 4 Inertie d'une spline matérielle

Le terme de gauche des équations de Lagrange correspond à l'énergie induite par le mouvement du système. Comme la configuration de la spline matérielle est déterminée par un champ de position et un champ de rotation, son énergie cinétique est la somme d'une énergie de translation et d'une énergie de rotation. L'énergie de translation correspond au déplacement des points de contrôle, et l'énergie de rotation est due au mouvement des sections autour de la fibre moyenne. Nous définissons la *matrice de densité d'inertie*  $\mathbb{J}$ , non homogène, qui varie le long de la spline :

$$\mathbb{J} = \begin{pmatrix} \mu(u) & 0 & 0 & 0 \\ 0 & \mu(u) & 0 & 0 \\ 0 & 0 & \mu(u) & 0 \\ 0 & 0 & 0 & I_o(u) \end{pmatrix} \quad (\text{II.70})$$

$\mu(u)$  correspondant à la masse linéique et  $I_o$  au moment polaire d'inertie. Nous rappelons que  $I_o$  est différent du module de rigidité à la torsion,  $J$ .

Nous proposons alors une formulation de l'énergie cinétique  $T$  :

$$T = \frac{1}{2} \int_0^L \frac{d\mathbf{q}^t}{dt} \mathbb{J} \frac{d\mathbf{q}}{dt} ds \quad (\text{II.71})$$

Dériver cette expression par rapport aux coordonnées généralisées  $\mathbf{q}_i$  nous donne le terme de gauche des équations de Lagrange :

$$\frac{d}{dt} \frac{\partial T}{\partial \dot{\mathbf{q}}_i} = \frac{1}{2} \int_0^L \frac{d}{dt} \frac{\partial \frac{d\mathbf{q}^t}{dt} \mathbb{J} \frac{d\mathbf{q}}{dt}}{\partial \dot{\mathbf{q}}_i} ds \quad (\text{II.72})$$

En remplaçant le champ  $\mathbf{q}$  par son expression en fonction des degrés de liberté (équation II.22), nous obtenons d'une manière similaire à celle décrite par Rémion *et al.* (RNG99) :

$$\frac{d}{dt} \frac{\partial T}{\partial \dot{\mathbf{q}}_i} = \sum_{j=1}^n \int_0^L \mathbb{J}(b_i(u)b_j(u)) ds \frac{d^2 \mathbf{q}_j}{dt^2} \quad (\text{II.73})$$

En considérant la matrice  $\mathbb{M}$  et le vecteur d'accélération  $\mathbf{A}$ , de composantes respectives  $\mathbb{M}_{i,j} = \int_0^L \mathbb{J}(b_i(u)b_j(u)) ds$  et  $\mathbf{A}_j = \frac{d^2 \mathbf{q}_j}{dt^2}$ , cette équation se réécrit plus simplement :

$$\frac{d}{dt} \frac{\partial T}{\partial \dot{\mathbf{q}}_i} = \sum_{j=1}^n \mathbb{M}_{i,j} \mathbf{A}_j \quad (\text{II.74})$$

Ce terme s'exprime plus simplement sous la forme d'un produit matrice-vecteur :

$$\mathbb{M}\mathbf{A} \quad (\text{II.75})$$

La forme dynamique du système est donc de forme matricielle :

$$\mathbb{M}\mathbf{A} = \mathbf{F} + \mathbf{P} \quad (\text{II.76})$$

Les quatre sous-systèmes correspondant au champ de position  $\mathbf{r} = (x, y, z)$  et au roulis  $\theta$  peuvent être résolus indépendamment. Comme les splines que nous avons choisies ont la propriété de localité, la matrice  $\mathbb{M}$  est bande de largeur  $2l - 1$ , où  $l$  est la localité de la spline. Les splines cardinales et les B-splines cubiques ont une localité égale à 4.

Nous pouvons maintenant simuler une spline matérielle à déformations réversibles. Ajouter la plasticité au modèle ne nécessite que quelques considérations supplémentaires, que nous allons préciser dans la section suivante.

## 5 Résolution efficace du système d'équations

Le système dynamique obtenu II.76 est très semblable à celui considéré par Nocent *et al.* (NR01) et par Lenoir *et al.* (LGMCo5). Pour trouver les inconnues du système, nous avons recours à une simple décomposition et résolution LU de type *bande* à chaque pas de temps. Nous avons dans un premier temps utilisé deux méthodes d'intégration, les mêmes utilisées par Lenoir *et al.* (LMGCo2) : la *méthode de Runge Kutta* à l'ordre 4, rapide mais instable une rigidité de flexion un tant soit peu élevée ; et le *schéma Euler-Implicite avec une version itérative de la méthode de Broyden*, proposé par Hilde *et al.* (HMC01). La méthode de Broyden consiste à évaluer numériquement la matrice hessienne  $\mathbb{K}$ . Hilde *et al.* ont proposé de le faire de manière itérative, en subdivisant le pas de temps jusqu'à ce que le résultat soit acceptable. Cependant, une raideur élevée augmente considérablement le nombre d'itérations nécessaires et le temps de calcul pour un même pas, pour parfois finalement diverger. Nous allons donc appliquer un *schéma implicite linéarisé* (BW98; DSB99), (voir section b, page 46), mais cette fois en calculant analytiquement la matrice de rigidité.

### a Schéma implicite linéarisé

A partir de l'équation II.76, nous posons le schéma Euler-Implicite correspondant :

$$\begin{cases} \mathbb{M}\mathbf{A}^{n+1} = \mathbf{F} + \mathbf{P}^{n+1} \\ \mathbf{X}^{n+1} = \mathbf{X}^n + \mathbf{V}^{n+1}\Delta t \end{cases} \quad (\text{II.77})$$

avec  $\Delta t$  le pas de temps entre deux états successifs  $n$  et  $n + 1$  de la simulation, et  $\mathbf{A}$ ,  $\mathbf{V}$  et  $\mathbf{X}$  les vecteurs d'accélération, de vitesse et de position qui rassemblent respectivement les degrés de liberté  $\ddot{\mathbf{q}}_i$ ,  $\dot{\mathbf{q}}_i$  et  $\mathbf{q}_i$  de chaque point de contrôle de la spline matérielle. La vitesse  $\mathbf{V}^{n+1}$  s'exprime de manière similaire à la position  $\mathbf{X}^{n+1}$  :

$$\mathbf{V}^{n+1} = \mathbf{V}^n + \mathbf{A}^{n+1}\Delta t \quad (\text{II.78})$$

Les différences de vitesse et de position (les déplacements) entre deux pas sont notés :

$$\begin{cases} \Delta^{n+1}\mathbf{V} = \mathbf{V}^{n+1} - \mathbf{V}^n \\ \Delta^{n+1}\mathbf{X} = \mathbf{X}^{n+1} - \mathbf{X}^n \end{cases} \quad (\text{II.79})$$

Notons que par rapport au schéma I.63,  $\mathbb{M}$  est une matrice non diagonale mais bande et n'est donc pas équivalente à un scalaire pour chaque degré de liberté : la masse est répartie sur toute la spline et non ponctuelle. Nous considérons que  $\mathbb{M}$  varie peu entre deux pas :  $\mathbb{M}^n = \mathbb{M}^{n+1}$ . Nous appliquons également la méthode de Newton-Raphson avec un développement limité à l'ordre 1 :

$$\mathbf{P}^{n+1} = \mathbf{P}^n + \frac{\partial \mathbf{P}^n}{\partial \mathbf{X}} \Delta^{n+1}\mathbf{X} \quad (\text{II.80})$$

Le terme  $\frac{\partial \mathbf{P}^n}{\partial \mathbf{X}}$  ayant la forme d'une matrice, nous posons :

$$\mathbb{K}^n = -\frac{\partial \mathbf{P}^n}{\partial \mathbf{X}} = \frac{\partial^2 U^n}{\partial \mathbf{X}^2} \quad (\text{II.81})$$

Après quelques calculs, nous obtenons un système similaire à celui du masse-ressort série (équation I.64) :

$$\left( \frac{\mathbb{M}}{\Delta t^2} + \mathbb{K}^n \right) \Delta^{n+1}\mathbf{X} = \frac{\mathbb{M}}{\Delta t} \mathbf{V}^n + \mathbf{F} + \mathbf{P}^n \quad (\text{II.82})$$

où  $\mathbf{V}^n = \frac{\mathbf{X}^n}{\Delta t}$ . Le système quasi-statique s'obtient très facilement :

$$\mathbb{K}^n \Delta \mathbf{X} = \mathbf{F} + \mathbf{P}^n \quad (\text{II.83})$$

Remarquons que si l'on fait tendre  $\Delta t$  vers l'infini, l'équation II.82 tend vers l'équation II.83. Ce constat sera le point de départ d'un algorithme exposé au chapitre IV, qui permet à la simulation d'alterner automatiquement entre l'équilibre quasi-statique et l'équilibre dynamique.

Cette nouvelle formulation a donc l'avantage d'être bien plus stable que la précédente, avec une résolution constante pour n'importe quelle raideur. Les coordonnées  $x, y, z$  et  $\theta$  étant liées, nous perdons la possibilité de découpler le système en 4 systèmes indépendants. Cela n'a guère d'importance, puisque le système est toujours de complexité linéaire. Cela nous permet d'utiliser des contraintes portant sur plusieurs coordonnées à la fois, comme des contraintes de distance entre deux points de la spline. Enfin, un autre avantage non négligeable est la résolution des contraintes en déplacements et non en accélérations : le coût résultant est bien moindre puisque direct. De plus nous pouvons les positionner dans le système de façon à garder la structure bande de la matrice. Contrairement à Lenoir *et al.* (LGM<sup>+</sup>04), la résolution du système contraint sera donc de complexité linéaire, sans passer par des étapes intermédiaires.

## b Matrice de raideur du système

L'expression de la matrice de raideur  $\mathbb{K}$  est la principale difficulté de ce schéma de résolution. Elle est en effet fort complexe et son expression développée prend plusieurs pages si l'on ne s'attache pas à la factoriser au maximum. Nous avons fourni un effort très important dans ce sens. Le calcul a été vérifié à l'aide d'un logiciel de calcul symbolique, *Maple*. Celui-ci est par contre incapable de fournir les expressions condensées sans lui indiquer les substitutions à faire. Nous rappelons celles effectuées pour le calcul des forces généralisées et nous en introduisons quelques-unes supplémentaires :

$$\mathbf{C} = \mathbf{r}' \times \mathbf{r}'' \quad (\text{II.84})$$

$$\mathbf{P}^i = \frac{\partial \mathbf{C}}{\partial \mathbf{r}_i} \quad (\text{II.85})$$

$$\mathbf{R}^i = \mathbf{C} b_i''' - \mathbf{P}^i \times \mathbf{r}''' \quad (\text{II.86})$$

$$\mathbf{G}^i = \mathbf{C} \times \mathbf{P}^i \quad (\text{II.87})$$

$$\mathbf{T}^i = \mathbf{R}^i - 2\tau \mathbf{G}^i \quad (\text{II.88})$$

$$\mathbf{F}^i = \frac{\mathbf{T}^i}{\|\mathbf{C}\|^2} - \frac{\theta' \mathbf{r}' b_i'}{\|\mathbf{r}'\|^2} \quad (\text{II.89})$$

$$\mathbf{B}^{i,j} = b_j'' b_i' - b_j' b_i'' \quad (\text{II.90})$$

Elles présentent un intérêt double. Le premier est de clarifier et compacter de manière très conséquente les expressions. Le second est de diminuer radicalement le temps de calcul pour rendre la formulation adaptée au temps-réel. Nous définissons également la liste  $C$  comme la permutation des éléments de l'ensemble  $\{x, y, z, \theta\}$  :

$$C = (\alpha, \beta, \gamma, \theta) \quad (\text{II.91})$$

En considérant deux éléments  $c_1$  et  $c_2$  de  $C$ , un élément  $\mathbb{K}^{i,j}$  de la matrice de raideur, situé à la  $i^{\text{ème}}$  ligne et la  $j^{\text{ème}}$  colonne s'exprime de la manière suivante :

$$K_{c_1, c_2}^{i,j}(\mathbf{r}, \theta) = -\frac{\partial P_a^i(\mathbf{q}_{c_1})}{\partial c_2^j} \quad (\text{II.92})$$

La matrice de raideur est symétrique. Ses éléments ont donc la propriété suivante :

$$K_{c_2, c_1}^{j,i}(\mathbf{r}, \theta) = K_{c_1, c_2}^{i,j}(\mathbf{r}, \theta) \quad (\text{II.93})$$

Nous ne donnons ici que les raideurs généralisées pour un état initial rectiligne, sans donner les étapes intermédiaires, afin de ne pas trop alourdir la lecture de ce chapitre. Pour les calculer, nous dérivons les forces généralisées obtenues précédemment (équations II.57, II.62, II.63 et II.65) par rapport aux degrés de liberté du système.

### Traction-Compression

$$K_{s,(\alpha,\alpha)}^{i,j}(\mathbf{r}) = E \int_0^L A(u) \left( 1 + \frac{\mathbf{r}'_\alpha{}^2 \|\mathbf{r}'_0\|}{\|\mathbf{r}'\|^3} - \frac{\|\mathbf{r}'_0\|}{\|\mathbf{r}'\|} \right) b'_i b'_j ds \quad (\text{II.94})$$

$$K_{s,(\alpha,\beta)}^{i,j}(\mathbf{r}) = E \int_0^L A(u) \frac{\mathbf{r}'_\alpha \mathbf{r}'_\beta \|\mathbf{r}'_0\|}{\|\mathbf{r}'\|^3} b'_i b'_j ds \quad (\text{II.95})$$

### Torsion

$$K_{t,(\alpha,\alpha)}^{i,j}(\mathbf{r}, \theta) = G \int_0^L J(u) \left( \mathbf{F}_\alpha^j \mathbf{F}_\alpha^i + \epsilon_t \left( -2 \frac{\mathbf{R}_\alpha^i \mathbf{G}_\alpha^j + \mathbf{R}_\alpha^j \mathbf{G}_\alpha^i}{\|\mathbf{C}\|^4} + 8\tau \frac{\mathbf{G}_\alpha^i \mathbf{G}_\alpha^j}{\|\mathbf{C}\|^4} - 2\tau \frac{\mathbf{P}_\beta^i \mathbf{P}_\beta^j + \mathbf{P}_\gamma^i \mathbf{P}_\gamma^j}{\|\mathbf{C}\|^2} \right) \right) ds \quad (\text{II.96})$$

$$K_{t,(\theta,\theta)}^{i,j}(\mathbf{r}, \theta) = G \int_0^L J(u) \frac{b'_i b'_j}{\|\mathbf{r}'\|^2} ds \quad (\text{II.97})$$

$$K_{t,(\alpha,\theta)}^{i,j}(\mathbf{r}, \theta) = G \int_0^L J(u) \left( \frac{b'_j}{\|\mathbf{r}'\|} \mathbf{F}_\alpha^i - \frac{\epsilon_t b'_i b'_j \mathbf{r}'_\alpha}{\|\mathbf{r}'\|^3} \right) ds \quad (\text{II.98})$$

$$K_{t,(\alpha,\beta)}^{i,j}(\mathbf{r}, \theta) = G \int_0^L J(u) \left( \mathbf{F}_\beta^j \mathbf{F}_\alpha^i + \epsilon_t \left( \frac{-\mathbf{P}_\gamma^j b_i'' + \mathbf{P}_\gamma^i b_j'' + \mathbf{B}^{i,j} r_\gamma'''}{\|\mathbf{C}\|^2} - 2 \frac{\mathbf{R}_\alpha^i \mathbf{G}_\beta^j + \mathbf{R}_\beta^j \mathbf{G}_\alpha^i}{\|\mathbf{C}\|^4} + 8\tau \frac{\mathbf{G}_\alpha^i \mathbf{G}_\beta^j}{\|\mathbf{C}\|^4} + 2\tau \frac{\mathbf{P}_\alpha^j \mathbf{P}_\beta^i + \mathbf{C}_\gamma \mathbf{B}^{i,j}}{\|\mathbf{C}\|^2} \right) \right) ds \quad (\text{II.99})$$

### Flexion

$$K_{b,(\alpha,\alpha)}^{i,j}(\mathbf{r}) = E \int_0^L I(u) \left( \frac{\mathbf{P}_\beta^i \mathbf{P}_\beta^j + \mathbf{P}_\gamma^i \mathbf{P}_\gamma^j}{\|\mathbf{r}'\|^6} - 6 \frac{((\mathbf{C} \times \mathbf{P}^i)_\alpha b'_j + \mathbf{G}_\alpha^j b'_i) \mathbf{r}'_\alpha}{\|\mathbf{r}'\|^8} + 3 \frac{k^2 b'_i b'_j}{\|\mathbf{r}'^2\|} \left( 8 \frac{\mathbf{r}'_\alpha{}^2}{\|\mathbf{r}'\|^2} - 1 \right) \right) ds \quad (\text{II.100})$$

$$K_{b,(\alpha,\beta)}^{i,j}(\mathbf{r}) = E \int_0^L I(u) \left( -\frac{\mathbf{P}_\alpha^j \mathbf{P}_\beta^i + \mathbf{C}_\gamma \mathbf{B}^{i,j}}{\|\mathbf{r}'\|^6} - 6 \frac{\mathbf{G}_\alpha^i \mathbf{r}'_\beta b'_j + \mathbf{G}_\beta^j \mathbf{r}'_\alpha b'_i}{\|\mathbf{r}'\|^8} + 24k^2 b'_i b'_j \frac{\mathbf{r}'_\alpha \mathbf{r}'_\beta}{\|\mathbf{r}'\|^4} \right) ds \quad (\text{II.101})$$

Etant donné que certains dénominateurs s'annulent dans les expressions d'une part en torsion, et d'autre part en flexion pour une configuration rectiligne d'un état initial déformé, il est nécessaire d'effectuer des prolongements par continuité. Le système étant maintenant résolu de manière efficace et stable, nous pouvons passer à l'ajout des déformations irréversibles.

## C MODÈLE MÉCANIQUE SPLINE À DÉFORMATIONS IRRÉVERSIBLES

Nous traitons les splines matérielles comme des objets à plastiques, qui peuvent également se rompre. Modéliser la plasticité soulève deux problèmes : le premier est de déterminer

à partir de quelle amplitude la déformation va devenir irréversible : ce seuil est la limite d'élasticité ; le second est de déterminer la relation entre la contrainte appliquée et les déformations résultante.

## 1 Limite d'élasticité

A la limite d'élasticité, les contraintes et les déformations sont encore proportionnelles. Nous choisissons de convertir les contraintes de limite d'élasticité en déformations, et de travailler avec celles-ci. En effet, les déformations sont calculées à chaque pas de simulation et il est alors peu coûteux de les comparer aux valeurs limites. Nous choisissons de considérer séparément les limites et la plasticité de chaque type de déformation. Ainsi, la spline matérielle peut par exemple rester élastique en élongation et en même temps se fléchir de manière irréversible. Nous n'avons donc pas utilisé de critère de limite d'élasticité, mais pris pour chaque type de déformation la valeur limite. Comme celles-ci sont difficiles à obtenir, surtout en flexion, nous avons procédé par extrapolation de certains matériaux connus.

## 2 Déformations plastiques

Pour simuler les déformations plastiques, nous nous restreignons à un comportement de *plasticité parfaite* : au-delà de la limite d'élasticité  $A$ , la contrainte appliquée sur le matériau n'augmente plus et tous les efforts supplémentaires sont directement convertis en déformations. Les contraintes et les déformations ne sont alors plus proportionnelles (figure I.2, page 12). Nous pourrions simuler la plasticité réelle, en utilisant une fonction dépendant de la déformation  $\varepsilon$  et de la déformation limite d'élasticité  $\varepsilon_A$ , qui donnerait la part des efforts à convertir en déformations. Mais en pratique, les matériaux homogènes et isotropes ont un comportement suffisamment proche de la plasticité parfaite pour accepter l'approximation. Par exemple, la courbe de contrainte-déformation du nylon, donnée par Shuttleworth (SVH99), justifie cette hypothèse.

### a Plasticité parfaite

En plasticité parfaite, la force  $\mathcal{F}$  correspondant à un type de déformation ne peut excéder un seuil positif  $\mathcal{F}_{A_+}$  ou passer sous un seuil négatif  $\mathcal{F}_{A_-}$ . En l'absence de déformation plastique préalable, ces seuils correspondent aux limites d'élasticité  $A_+$  et  $A_-$ . Le centre de la section que nous avons définie est centre de symétrie. Ces seuils sont donc diamétralement opposés en élongation et en torsion, en plus de la flexion :

$$\mathcal{F}_{A_+} = -\mathcal{F}_{A_-} \quad (\text{II.102})$$

Aux points  $A_+$  et  $A_-$ , le comportement est encore élastique. Si l'on connaît la force limite, on peut donc facilement déduire la déformation limite  $\varepsilon$ . Comme il est plus facile d'obtenir les déformations, nous préférons travailler avec les déformations limites d'élasticité  $\varepsilon_{A_+}$  and  $\varepsilon_{A_-}$ .

Au delà de ces seuils, les efforts appliqués sur la spline matérielle sont entièrement convertis en une déformation supplémentaire  $\Delta\varepsilon^p$ . Nous appelons cette quantité *décalage de plasticité*. Cette déformation s'ajoute de manière définitive à celle déjà accumulée : une fois tous les efforts relâchés, la configuration sera translatée de  $\Delta\varepsilon^p$ . La déformation  $\varepsilon$  d'une configuration s'exprime donc :

$$\varepsilon = \varepsilon^e - \varepsilon^0 - \Delta\varepsilon^p \quad (\text{II.103})$$

où  $\varepsilon^e$  désigne la déformation élastique et  $\varepsilon^0$  l'état initial. La formulation de l'énergie potentielle de déformation devient alors :

$$U = \frac{1}{2} \int_0^L (\varepsilon - \varepsilon^0 - \Delta\varepsilon^p)^t \mathbb{H} (\varepsilon - \varepsilon^0 - \Delta\varepsilon^p) ds \quad (\text{II.104})$$

Les résultats obtenus par dérivation par rapport aux degrés de liberté restent acquis :  $\Delta\varepsilon^p$  est vu comme une constante, au même titre que  $\varepsilon^0$ .

En pratique, à chaque fois que la déformation  $\varepsilon$  dépasse la limite d'élasticité  $\varepsilon_{A_+}$ , leur différence est ajoutée dans  $\Delta\varepsilon^p$ . Graphiquement, cela revient à translater la courbe d'élasticité selon l'axe des déformations (voir figure a). Ces observations se traduisent facilement

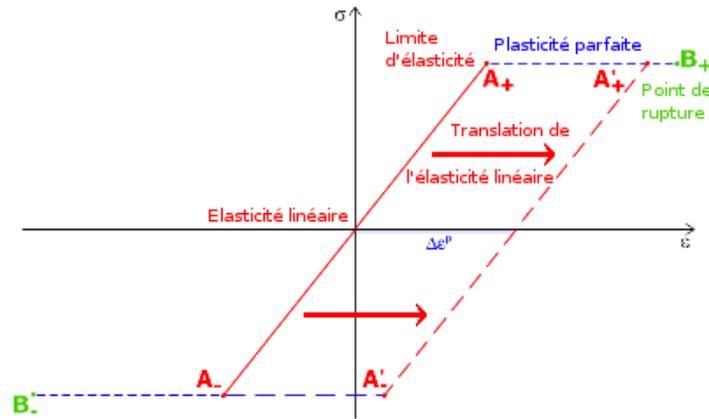


Fig. II.7 – Les domaines de déformation réversibles et irréversibles : élasticité linéaire, plasticité parfaite et point de rupture.  $\Delta\varepsilon^p$  correspond à la déformation conservée par le matériau au repos.

sous forme d'algorithme (algorithme 1).

---

#### Algorithme 1 : Algorithme de plasticité

---

**pour chaque** échantillon de la spline matérielle **faire**

    Calculer le vecteur initial  $\varepsilon^0$   
     $\Delta\varepsilon^p \leftarrow 0$

**tant que** simulation **faire**

**pour chaque** échantillon de la spline matérielle **faire**

$\varepsilon \leftarrow \varepsilon^e - \varepsilon^0 - \Delta\varepsilon^p$

$\varepsilon_+ \leftarrow \varepsilon - \varepsilon_{A_+}$

$\varepsilon_- \leftarrow \varepsilon - \varepsilon_{A_-}$

**si**  $\varepsilon_+ > 0$  **alors**

$\Delta\varepsilon^p \leftarrow \Delta\varepsilon^p + \varepsilon_+$

**si**  $\varepsilon_- < 0$  **alors**

$\Delta\varepsilon^p \leftarrow \Delta\varepsilon^p + \varepsilon_-$

Nous supposons qu'au début de la simulation le matériau n'a pas été endommagé préalablement : le décalage de plasticité  $\Delta\varepsilon^p$  est donc nul. A chaque pas de simulation, nous vérifions si la déformation  $\varepsilon$  a dépassé la limite  $\varepsilon_{A_+}$  ou  $\varepsilon_{A_-}$  selon le sens. Si tel est le cas, la différence égale à  $\varepsilon_+$  ou  $\varepsilon_-$  est ajoutée au décalage de plasticité  $\Delta\varepsilon^p$ . La déformation courante  $\varepsilon$  devient alors égale à la limite d'élasticité. Tout excès supplémentaire aux pas de temps suivants sera traité de la même façon. Notons que nous n'avons pas à mettre à jour les limites d'élasticité  $\varepsilon_{A_+}$  et  $\varepsilon_{A_-}$ , cela est fait implicitement. Notons également que le décalage de plasticité peut être à nouveau réduit à 0, en dépassant la limite d'élasticité opposée.

#### b Rupture

La rupture est détectée de manière similaire à la plasticité, c'est-à-dire par un dépassement de seuil, le point de rupture  $B$  (figure a). Celui-ci se définit également par rapport à une déformation. Quand la *déformation relative*  $\varepsilon^r = \varepsilon^e - \varepsilon^0$  est plus élevée que la déformation

positive  $\varepsilon_{B+}$ , ou plus petite que la négative  $\varepsilon_{B-}$ , le matériau se rompt. Pour modéliser ce comportement, nous reprenons l'idée d'utiliser une B-Spline matérielle et d'insérer 4 noeuds identiques à l'abscisse correspondante. Ce comportement donne un algorithme simple 2.

---

**Algorithme 2** : Algorithme de rupture
 

---

```

tant que simulation faire
  pour chaque échantillon de la spline matérielle faire
     $\varepsilon^r \leftarrow \varepsilon^e - \varepsilon^0$ 
    si  $\varepsilon^r > \varepsilon_{B+}$  ou  $\varepsilon^r < \varepsilon_{B-}$  alors
       $\lfloor$  Simuler la rupture : insertion de 4 noeuds à l'abscisse correspondante
  
```

---

Un objet peut toutefois se rompre, sans que la limite de rupture  $B$  ait été forcément atteinte. En effet, une force appliquée de manière répétée conduit également à la rupture, par fatigue. Nous choisissons cependant de ne pas la modéliser. Les éléments que nous sommes amenés à manipuler étant relativement déformables, ils y sont peu sujets. Tout du moins, la rupture par fatigue sur de tels matériaux est suffisamment lente pour ne pas risquer pas de se produire pendant la simulation! Nous allons maintenant combiner la plasticité et la rupture en un seul algorithme.

**c Plasticité, rupture**

La déformation plastique ne déplace évidemment pas les points de rupture dans le graphe de contrainte-déformation, sinon ils ne seraient jamais atteints. Les limites d'élasticité et de rupture sont indépendantes. Il est donc difficile de tirer parti de la modélisation commune de la plasticité et de la rupture. L'algorithme 3 regroupe l'algorithme de plasticité 1 et l'algorithme de rupture 2.

---

**Algorithme 3** : Algorithme de plasticité et de rupture
 

---

```

pour chaque échantillon de la spline matérielle faire
  Calculer le vecteur initial  $\varepsilon^0$ 
   $\Delta\varepsilon^p \leftarrow 0$ 
  tant que simulation faire
    pour chaque échantillon de la spline matérielle faire
       $\varepsilon^r \leftarrow \varepsilon^e - \varepsilon^0$ 
       $\varepsilon \leftarrow \varepsilon^r - \Delta\varepsilon^p$ 
       $\varepsilon_+ \leftarrow \varepsilon - \varepsilon_{A+}$ 
       $\varepsilon_- \leftarrow \varepsilon - \varepsilon_{A-}$ 
      si  $\varepsilon^r > \varepsilon_{B+}$  ou  $\varepsilon^r < \varepsilon_{B-}$  alors
         $\lfloor$  Simuler la rupture : insertion de 4 noeuds à l'abscisse correspondante
      si  $\varepsilon_+ > 0$  alors
         $\lfloor$   $\Delta\varepsilon^p \leftarrow \Delta\varepsilon^p + \varepsilon_+$ 
      si  $\varepsilon_- < 0$  alors
         $\lfloor$   $\Delta\varepsilon^p \leftarrow \Delta\varepsilon^p + \varepsilon_-$ 
    
```

---

Maintenant que les déformations réversibles et irréversibles sont modélisées et intégrées au système, avec une méthode de résolution efficace et stable, nous pouvons passer à l'étape de validation de notre modèle de spline matérielle.

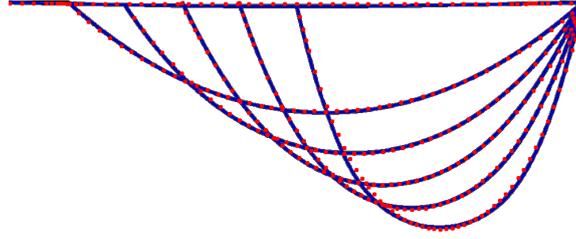


Fig. II.8 – Chaînette obtenue pour des écarts de 1m,0,9m,0,8m,0,7m et 0,6m. La forme théorique est donnée par les points.

## D VALIDATIONS MÉCANIQUES

Nous avons procédé à de nombreux tests afin de valider notre modèle de spline matérielle. Tout d’abord, nous avons vérifié la précision de chaque type de déformation. Nous avons validé le comportement non linéaire des déformations ainsi que les couplages, puis l’aspect dynamique.

### 1 Déformations

Nous avons reproduit des configurations classiques qui mettent à l’épreuve les différentes déformations de manière indépendante : la chaînette pour la traction, la poutre encastree soumise à une force à son extrémité pour la flexion et la rotation d’une poutre encastree à son extrémité pour la torsion.

#### a Validation de la traction : la chaînette

Notre première validation concerne la *chaînette*. C’est la forme que prend une chaîne suspendue à ses extrémités et soumise à la seule gravité. Son équation dans le plan a été donnée par Leibniz, Huygens et Johann Bernoulli en 1691 :

$$y = \frac{\mathcal{F}_h}{A\mu} \left( ch\left(\frac{A\mu x}{\mathcal{F}_h}\right) - 1 \right) \quad (\text{II.105})$$

où  $\mathcal{F}_h$  est la composante horizontale de la force  $\mathcal{F}$  qui s’exerce en chaque point de la chaîne,  $\mu$  la densité linéique et  $A$  l’aire de la section. Nous fournissons son calcul détaillé en annexe E. Nous y montrons également que la forme adoptée ne dépend que de l’effort de traction. Cela nous permet de valider deux points du modèle : sa précision en traction-compression et l’indépendance de la traction-compression avec les autres types de déformations, la flexion et la torsion. L’équation II.105 donnée plus haut ne permet pas à elle seule de tracer la chaînette pour des conditions imposées. Il faut d’abord déterminer la composante horizontale de la force de traction  $\mathcal{F}_h$ . Celle-ci est liée à la longueur  $L$  de la chaîne :

$$L = \frac{2\mathcal{F}_h}{A\mu} sh\left(\frac{A\mu L}{2\mathcal{F}_h}\right) \quad (\text{II.106})$$

Nous avons fait varier l’écart entre les extrémités d’une ficelle, en considérant un rayon circulaire égal à 3mm. Les résultats sont donnés figure II.8. La simulation donne des résultats très proches des calculs théoriques.

#### b Validation de la flexion : spline matérielle encastree soumise à une force à son extrémité

Notre deuxième validation est le problème classique de la déformation d’une poutre encastree soumise à une force ponctuelle à son extrémité libre, illustré par la figure II.9(a). Cette fois, seule la flexion intervient dans la configuration prise par la poutre. Une étude détaillée de ce problème se trouve dans le livre d’Howell (How01), dans le cadre des

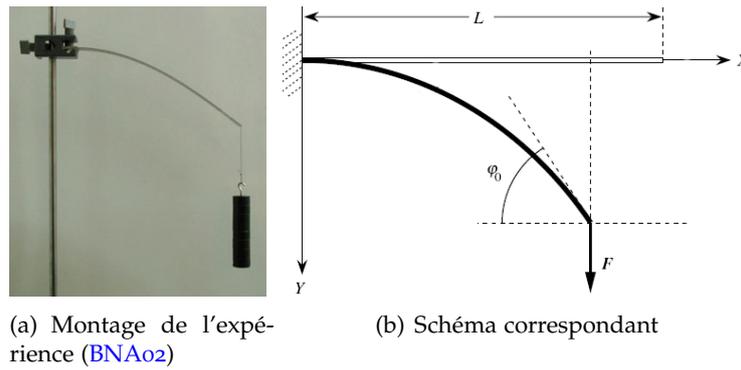


Fig. II.9 – Problème de la poutre encastrée.

petits et des grands déplacements. Beléndez *et al.* (BNA02) nous en livrent également une analyse complète. Pour plus de détails, le lecteur est invité à consulter ces références. Cette expérience est une bonne occasion de visualiser la différence entre petits et grands déplacements, puisque c'est essentiellement au niveau de la courbure que cela se traduit. La déformée est caractérisée d'une part par son équation paramétrique et d'autre part par l'angle  $\phi$  entre les extrémités de la configuration initiale et de la configuration étudiée (voir le schéma de la figure II.9(b)).

En petits déplacements, l'équation paramétrique de la poutre est, pour une force orientée vers le haut, la suivante :

$$y = \frac{Fx^2}{6EI} (3L - x) \quad (\text{II.107})$$

tandis que l'angle à son extrémité est donné par l'expression :

$$\phi = \frac{Fx}{2EI} (2L - x) \quad (\text{II.108})$$

L'angle  $\phi_0$ , à l'extrémité libre de la poutre est donc égal à

$$\phi_0 = \frac{FL^2}{2EI} \quad (\text{II.109})$$

En grands déplacements, l'équation paramétrique de la déformée est sous la forme d'intégrales elliptiques :

$$\begin{aligned} x &= \sqrt{\frac{EI}{2F}} \left( \sqrt{\sin \phi_0} - \sqrt{\sin \phi_0 - \sin \phi} \right) \\ y &= \sqrt{\frac{EI}{2F}} \int_0^\phi \frac{\sin \phi \, d\phi}{\sqrt{\sin \phi_0 - \sin \phi}} \end{aligned} \quad (\text{II.110})$$

Elle est liée d'une manière plus complexe à l'angle  $\phi_0$  de l'extrémité libre de la poutre. Pour obtenir sa valeur, il faut résoudre cette autre intégrale elliptique, qui exprime la longueur de la poutre :

$$L = \sqrt{\frac{EI}{2F}} \int_0^{\phi_0} \frac{d\phi}{\sqrt{\sin \phi_0 - \sin \phi}} \quad (\text{II.111})$$

Ces intégrales n'ont pas de solution type, mais il existe des schémas de résolution (How01; Mat81). Beléndez *et al.* (BNA02) résolvent les équations avec le logiciel *Mathematica*. Nous avons opté pour une résolution à l'aide de *Maple*. L'ensemble des paramètres que nous avons choisis est purement subjectif, une même déformée pouvant être obtenue avec des paramètres différents. C'est l'angle  $\phi_0$  résultant qui est discriminant.

Pour nos mesures, nous avons adopté les valeurs suivantes :

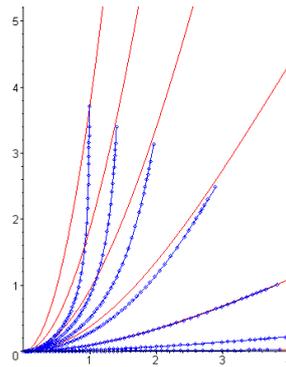
- un rayon  $R$  de 0,05m, ce qui donne un moment d'inertie axial  $I$  égal à  $4,90874m^4$  ;
- un module d'Young  $E$  égal à  $200GPa$  ;
- une longueur  $L$  de  $4m$

Force $F$ (N)	0	1e3	1e4	5e4	1e5	2e5	5e5	1e6	2e6
$\phi$ (rad) (pts dép.)	0	0,008148	0,08148	0,4074	0,8148	1,629	4,074	8,148	16,29
$y(L)$ (m) (pts dép.)	0	0,02173	0,2173	1,087	2,172	4,346	10,87	21,73	43,46
$\phi$ (rad) (gds dép.)	0	0,008148	0,08129	0,3854	0,6796	1,0263	1,3795	1,5122	
$y(\phi_0)$ (m) (gds dép.)	0	0,02059	0,2130	1,006	1,7229	2,483	3,135	3,397	

**Tab. II.1** – Forces appliquées à l’extrémité de la poutre encastrée, de longueur  $L = 4m$  et angles de déformée correspondants.

**Comparaison des petits et grands déplacements** Pour comparer les déformations en petits et grands déplacements, nous avons fait varier la force d’application  $F$  à l’extrémité, en prenant successivement les valeurs figurant dans le tableau II.1. Celui-ci donne également les angles  $\phi_0$  et les amplitudes maximales de déformée, correspondant aux deux niveaux d’approximation géométrique. Notons qu’en petits déplacements, les résultats deviennent vite aberrants : l’angle  $\phi$  dépasse allègrement la valeur limite  $\frac{\pi}{2}$  et la déformée la longueur de la poutre ! D’autre part, en grands déplacements, le temps de calcul nécessaire augmente exponentiellement quand  $\phi_0$  converge vers  $\frac{\pi}{2}$ .

La figure II.10 donne les courbes correspondantes. La différence d’amplitude est suffisamment flagrante pour nous convaincre de ne pas approximer la géométrie par des petits déplacements.

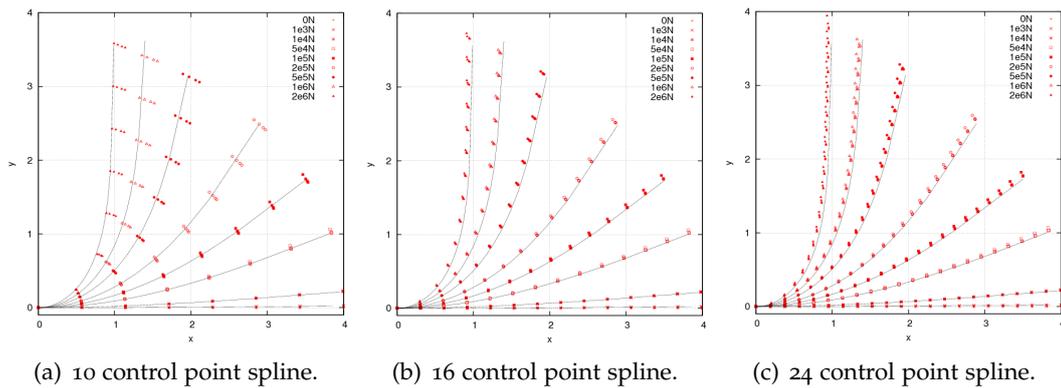


**Fig. II.10** – Petits déplacements (traits rouges) et grands déplacements (points bleus)

Notons enfin que la poutre s’allonge d’autant que l’angle de déformée augmente, puisque l’abscisse de l’extrémité reste constante, égale à la longueur de la poutre.

**Précision de la flexion** La précision obtenue en flexion dépend du niveau de discrétisation spatiale. Celle-ci intervient à deux niveaux : d’une part à celui des points de contrôle, d’autre part à celui des échantillons utilisés pour résoudre numériquement les intégrales des forces et des raideurs généralisées. La figure II.11 donne les résultats obtenus en faisant varier le nombre de points de contrôle et le nombre d’échantillons.

Deux constats s’offrent à nous. Le premier est que la déformée donnée par le modèle de spline matérielle converge vers la bonne solution. Le deuxième est qu’il le fait rapidement, et en fonction des deux niveaux de discrétisation. En effet, à partir d’une répartition régulière de 16 points de contrôle, les déformées ne présentent plus qu’une erreur très faible. Ceci est vraisemblablement dû à la corrélation entre le rayon de courbure local de la spline et la densité de points de contrôle par unité de longueur, que nous avons établie précédemment (voir section e, page 61). Les systèmes masse-ressorts (GS06), les chaînes articulées (Hado6; KGL07), les modèles à base d’éléments finis (ST07), ou ceux basés sur un modèle de Cosserat (Paio2) nécessitent un nombre de degrés de liberté d’un ordre parfois nettement supérieur (de l’ordre de la centaine, voir section e, page 87). Au niveau de la précision des intégrations numériques, 3 échantillons par segment semblent ici un nombre raisonnable pour 16 points de contrôle, ce qui revient à un échantillon pour une densité

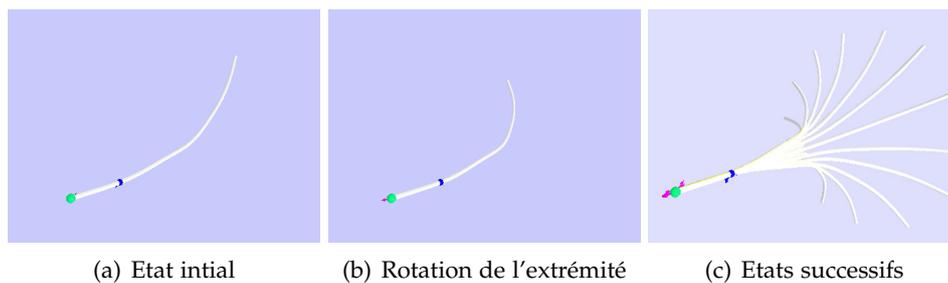


**Fig. II.11** – Déformée de la poutre encastrée, chargée à son extrémité par une force verticale. Les points correspondent à 3,5,10 et 20 échantillons par segment, tandis que les courbes correspondent aux équations théoriques.

de segments égale à  $\frac{2}{7}$ . Ce choix est équivalent à un nombre plus important d'échantillons mais un nombre plus faible de points de contrôle. En pratique, c'est le nombre total d'échantillons qui est vraiment discriminant. La convergence de la déformée ne repose que sur des considérations géométriques, pas matérielles, et permet donc de valider le cadre de déplacement choisi. La différence conséquente entre les petits déplacements et les grands déplacements nous conforte dans notre choix.

### c Validation de la torsion

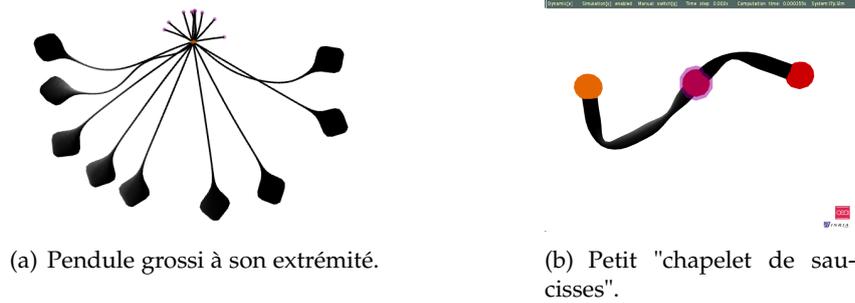
Pour valider le couplage entre le roulis et la torsion géométrique, nous effectuons la manipulation suivante : nous encastrons un câble courbe puis nous y faisons varier  $\theta$ . Avec une modélisation naïve, c'est-à-dire uniquement avec le roulis, toutes les sections pivoteraient mais la ligne moyenne resterait inchangée. La spline matérielle convertit bien le roulis en torsion géométrique, ce qui fait que le câble déformé effectue en rotation par rapport à l'axe d'encastrement. L'autre extrémité décrit bien un cercle (figure II.12).



**Fig. II.12** – Validation du couplage de la torsion matérielle et de la torsion géométrique.

### d Sections variables

Notre modèle permet de considérer un large éventail de types de section. Par manque de temps, nous n'avons implémenté et validé que les sections circulaires à rayon variable. Celles-ci procurent d'ailleurs des résultats visuels satisfaisants. Nous les illustrons à l'aide de deux manipulations (figure II.13). La première met en avant l'inertie induite par un poids à l'extrémité d'un pendule, tandis que la seconde montre la différence de rigidité de la spline matérielle selon l'épaisseur de la section, rapprochant son comportement de celui d'une chaîne rigide articulée.



(a) Pendule grossi à son extrémité.

(b) Petit "chapelet de saucisses".

**Fig. II.13** – Deux manipulations mettant en avant les possibilités d'une spline matérielle de section à rayon variable.

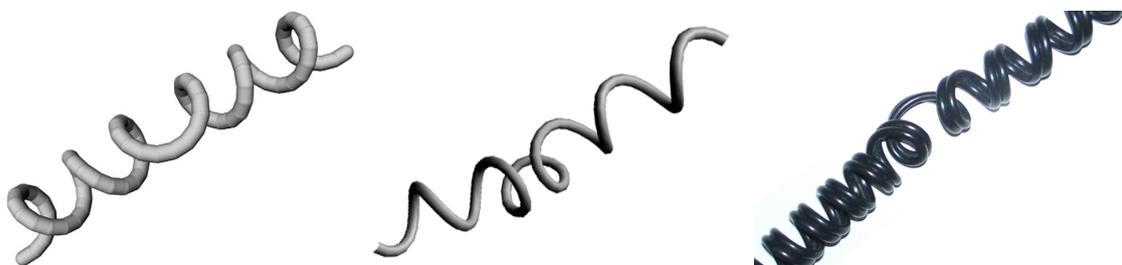
### e Etat initial déformé

La modélisation d'un ressort hélicoïdal valide des déformations initiales en flexion et en torsion. L'utilisation d'un schéma implicite amortit considérablement les oscillations du ressort; un pas de temps très petit, inférieur ou égal à la milliseconde, est de plus nécessaire pour que les oscillations ne soient pas filtrés numériquement (d'après le théorème de Shannon). Nous obtenons néanmoins un comportement semblable à celui d'un ressort (figure II.14).



**Fig. II.14** – Déformations successives d'un ressort soumis à plusieurs contraintes.

La simulation d'un ressort est également l'occasion de montrer une nouvelle fois le fort pouvoir de représentation du modèle spline. Celui-ci est ici pleinement exploité. La comparaison avec le modèle de Grégoire *et al.* (GS07) pour une configuration donnée d'hélice révèle un écart important en termes d'éléments nécessaires : 24 points de contrôle contre 80 particules. Une conséquence immédiate est bien sûr le temps de calcul moindre nécessaire pour simuler un pas. Nous avons constaté un facteur 20 à puissance égale entre les deux modèles.

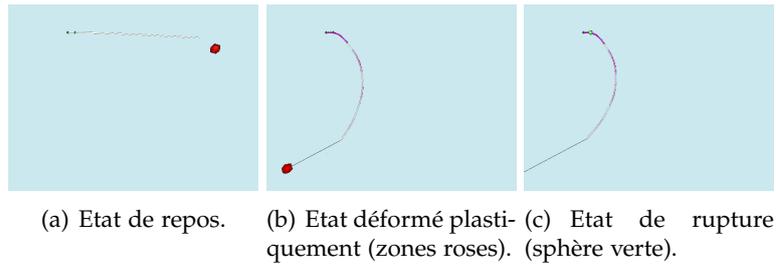


(a) Système masse-ressort de Grégoire *et al.*, composé de 80 éléments; temps de simulation : 32ms par pas (GS07) (b) Spline matérielle hélicoïdale de comparaison avec le réel. (c) Câble double de rasoir, pour comparaison avec le réel. temps de simulation : 0.8ms par pas.

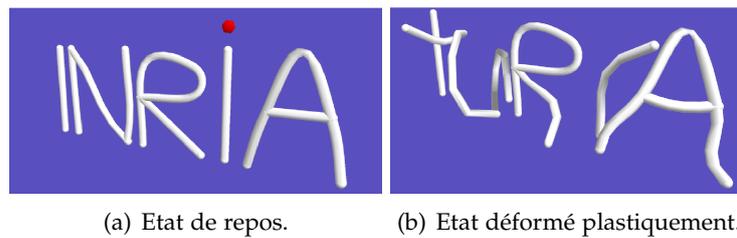
**Fig. II.15** – Pouvoir de représentation et état déformé d'une spline matérielle.

## f Plasticité et rupture

La figure II.16 illustre rapidement le phénomène de plasticité et de rupture. Un ressort effectue une tension sur l'extrémité d'un câble encastré. Au-delà d'une certaine déformation, des zones roses plus ou moins foncées apparaissent, indiquant l'amplitude des déformations ajoutées irréversiblement. Une sphère verte indique finalement le point de rupture de la spline, ce comportement n'étant pas simulé ici. Nous avons également modélisé un alphabet de lettres élastiques et plastiques, qui permettent de mettre en avant toutes les déformations II.17.



**Fig. II.16** – Etats successifs d'une spline matérielle encastrée, faisant intervenir les déformations plastiques et la détection de rupture.



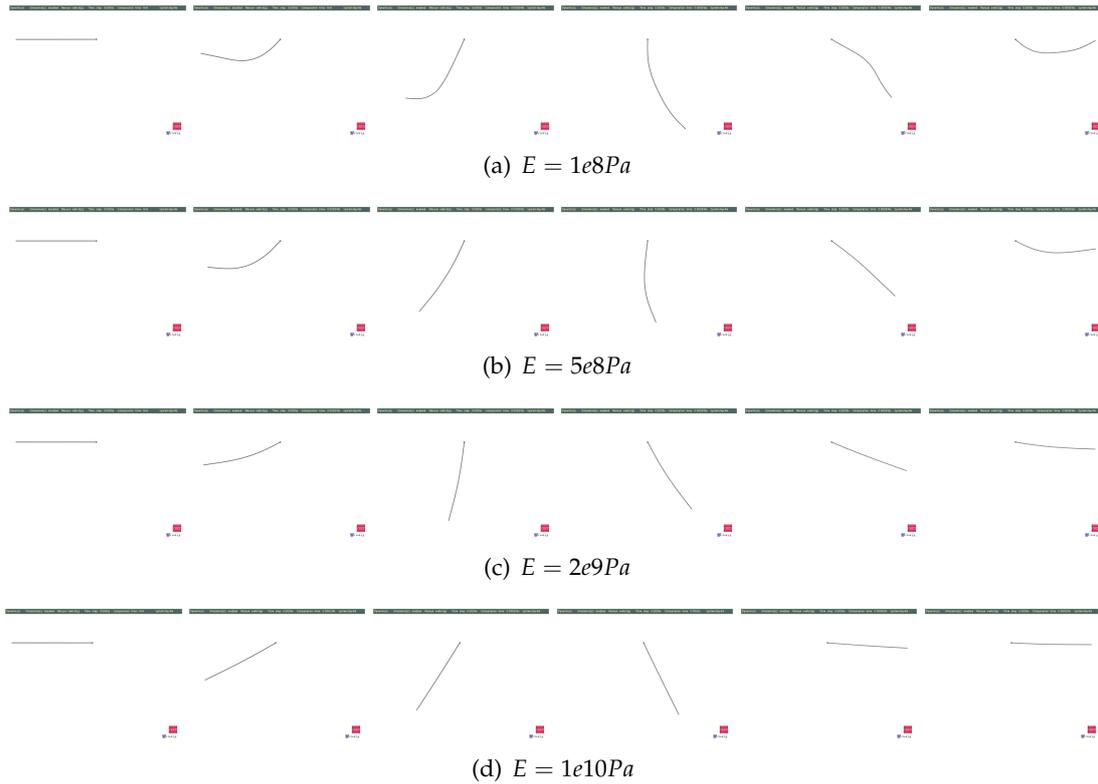
**Fig. II.17** – Lettres déformées plastiquement.

## 2 Dynamique

En vue d'illustrer l'aspect dynamique d'une spline matérielle, nous proposons ici de reproduire le mouvement d'un pendule pour différentes raideurs, en faisant varier le module d'Young. La figure II.18 montre des comportements de peu à très rigides de splines matérielles. A la manière de l'étude du pendule Redon *et al.* (RGL05), nous avons pris les images successives d'une oscillation, toutes séparées d'un intervalle de 80 ms.

## 3 Non-linéarités géométriques

Bertails *et al.* (BAC<sup>+</sup>06) ont mis en avant les instabilités en compression, flexion et torsion. Certaines ne peuvent être reproduites qu'avec des équations matérielles non-linéaires, c'est-à-dire dans le cadre des grandes déformations. En effet, il faut alors que les énergies de déformations soient dépendantes entre elles. Le flambage est un phénomène bien connu qui a fait l'objet de nombreuses études (Ant95; GHBV05; GvdHTN05). Par simple compression, nous ne pouvons a priori pas obtenir de flambage sur une poutre parfaitement rectiligne. Une force de direction différente de la ligne moyenne est nécessaire. Néanmoins, le schéma de résolution peut introduire des instabilités numériques qui vont artificiellement provoquer le flambage, avec une erreur plus ou moins importante. Nous avons effectué quelques tests avec les premiers schémas de résolution (Runge-Kutta à l'ordre 4 et Euler Implicite avec méthode de Broyden itérative).



**Fig. II.18** – Oscillation de pendules plus ou moins raides.

Une colonne verticale, de section circulaire, de densité  $\rho$ , de module d'Young  $E$  et de rayon  $r$  flambe sous son propre poids à partir d'une hauteur critique  $h_{crit}$ , définie par la relation suivante :

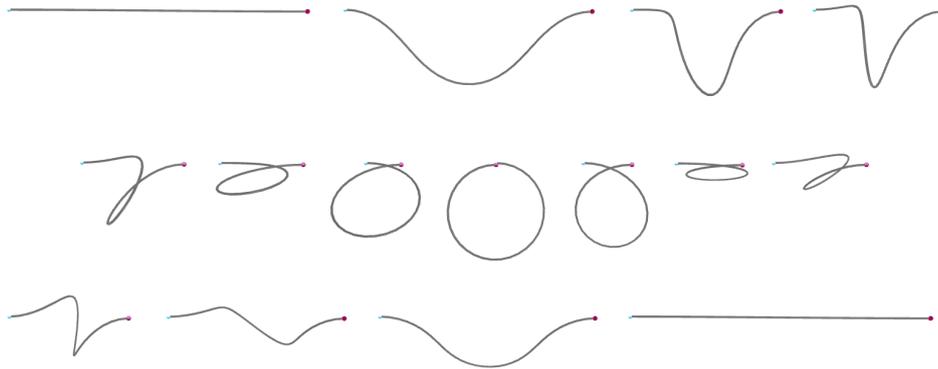
$$h_{crit} = \left( \frac{2.5Er^2}{\rho g} \right)^{1/3} \quad (\text{II.112})$$

La théorie donne  $h_{crit} = 1.335$  Nous avons fait varier la hauteur d'une colonne avec les paramètres suivants (TGAB07) :  $E = 5,6 \cdot 10^7 \text{ Pa}$ ,  $\rho = 2400 \text{ kg.m}^{-3}$  et  $r = 0.02 \text{ m}$ . Sans ajouter de force à l'équilibre initial, nous avons trouvé une hauteur critique de  $h_{crit} = 1.25 \text{ m}$ , soit avec une erreur de 6%. L'utilisation du schéma implicite d'Euler linéarisé, plus précis et plus stable, a réduit à néant les perturbations numériques, rendant impossible le flambage en compression.

Une autre manipulation classique pour mettre en évidence les instabilités en déformations est de rapprocher les deux extrémités d'une tige. La tige doit flamber au bout d'un moment. Puis les déformations de la tige ne peuvent plus se produire dans le même plan, la flexion se convertit partiellement en torsion, c'est la deuxième instabilité. En superposant les deux extrémités, un cercle est enfin obtenu. Les écarter nous ramène à la configuration initiale. La figure II.19 illustre les chemins aller et retour que nous avons obtenus, en introduisant une très légère perturbation orthogonale à la poutre<sup>4</sup> ; les déformations qui en résultent sont cependant conformes à la réalité.

Nous avons été incapables de reproduire le couplage flexion-torsion mis en avant par Boyer *et al.*, Goss *et al.* ou encore Wakamatsu *et al.* (BP05; GvdHTN05; WH04), pour des raisons d'instabilité. Il consiste à effectuer une rotation à l'une des deux extrémités d'une poutre doublement encastree. Celle-ci se tord d'abord, puis une partie de la torsion est convertie en flexion. Dans notre modèle, un état rectiligne ou quasi-rectiligne est mal défini en torsion géométrique, ce qui donne des réponses importantes à de petites perturbations dans ce voisinage, problème qu'ont également rencontré Grégoire *et al.* (GSo6). Néanmoins, dans le cas pratique, nous sommes peu confrontés à ce problème. L'utilisation

<sup>4</sup>Nous avons ici utilisé le schéma implicite d'Euler linéarisé.



**Fig. II.19** – *Reproduction des instabilités en compression, flexion et torsion.*

de quaternions permettrait le pallier, mais elle nous ferait perdre tous les avantages d'une formulation spline, à savoir la grande précision en flexion, l'ensemble des contraintes applicables et la multi-résolution.

## BILAN

Dans ce chapitre, nous avons repris une à une toutes les étapes du cheminement conduisant à notre proposition du modèle de spline matérielle. Dans un premier temps, nous avons étudié les différents types de splines, pour nous porter sur les splines cardinales, pour leur propriété d'interpolation, et les B-Splines pour les nombreux outils géométriques sous-jacents. Nous avons également évalué la précision des splines choisies pour modéliser les déformations, en essayant de reproduire une configuration circulaire. Dans le cas général, les B-Splines se trouvent être les plus adaptées, mais pour des objets rigides fort courbés, la spline cardinale peut leur être préférée. Nous avons ensuite matérialisé la spline en lui ajoutant une masse volumique et une section. Celle-ci n'est pas obligatoirement circulaire ou elliptique, mais elle doit posséder deux axes de symétrie orthogonaux. Dans un deuxième temps, nous avons combiné les équations de Lagrange et la théorie des poutres, dans le cadre des grands déplacements, pour obtenir des déformations élastiques (réversibles) précises. L'utilisation d'un schéma Euler-Implicite linéarisé permet d'allier stabilité et performance. Dans un troisième temps, nous avons considéré les déformations plastiques (irréversibles) et avons proposé un algorithme simple qui gère la plasticité parfaite, ainsi que la rupture. Dans un dernier temps, nous avons validé les déformations d'élongation, de flexion et de torsion. Nous avons enfin établi la précision du modèle de spline matérielle. Nous n'avons cependant pas encore traité les forces extérieures et les contraintes, qui permettent non seulement une manipulation aisée, mais aussi la reproduction de mécanismes plus ou moins complexes. C'est l'objet du chapitre qui suit.

## Chapitre III

# Interactions d'une spline matérielle

*Trop tirer rompt la corde.*

PROVERBE FRANÇAIS

**D**ANS le chapitre précédent, nous avons modélisé le comportement en déformation de notre spline matérielle. Nous n'avons cependant pas encore précisé comment interagir avec elle. L'interaction est d'ailleurs un point fort du modèle de spline matérielle et a déjà été étudiée par Lenoir *et al.* (Leno4), mais nous apportons de nombreux points nouveaux. L'interaction n'est pas limitée à la discrétisation et peut donc s'exercer en tout point de la spline. Elle peut se faire mécaniquement par l'action de forces ou géométriquement par l'ajout de multiplicateurs de Lagrange. L'idée de ce chapitre est de concilier les deux pour donner à l'utilisateur un contrôle intuitif et réaliste. Dans un premier temps, nous présentons rapidement comment appliquer des forces sur le système. Ensuite, comme le système d'équation est formulé de manière différente de celui de Lenoir *et al.*, nous reformulons et réorganisons toutes les contraintes existantes, en insistant sur leur évolution au cours de la simulation. Nous en proposons également de nouvelles, qui nous permettent ensuite de modéliser non seulement des structures complexes, telles que les faisceau de câbles, un brin d'ADN, ou un pont de corde, mais aussi des clips de fixation, ou un couplage avec un autre système mécanique. Nous réalisons enfin une étude portant sur la recherche de la meilleure représentation de la spline pour la détection de collision.

## A CONTRAINTES

Il existe deux façons d'agir sur un objet déformable. La première est de lui appliquer des forces. Pour les forces bien connues comme la gravitation ou celles découlant d'un potentiel, leur usage est facile. Si par contre on désire atteindre une configuration spécifique en ne connaissant pas la force à appliquer, la deuxième lui est préférable. L'ajout de multiplicateurs de Lagrange au système permet en effet de contraindre géométriquement un objet déformable. Nous allons dans cette section formuler ces deux types d'action extérieure à la spline matérielle. Quelques cas pratiques leur sont associés.

### 1 Application de forces

Nous avons auparavant formulé les systèmes dynamiques II.82 et statiques II.83, sans expliciter le vecteur généralisé des forces extérieures  $\mathbf{F}$ . C'est ce à quoi nous allons maintenant procéder dans cette section.

#### a Forces ponctuelles

Il est d'usage de distinguer les forces conservatives, celles dérivant d'un potentiel, des autres énergies.

Si une force  $\mathbf{f}$ , appliquée au point  $\mathbf{r}(u)$  de la spline matérielle, dérive d'un potentiel  $U$ , tous deux sont liés par la relation :

$$\mathbf{f} = - \overrightarrow{\text{grad}} U \quad (\text{III.1})$$

La force généralisée  $\mathbf{F}_i$  est définie de manière similaire par rapport aux coordonnées généralisées  $\mathbf{r}_i$  :

$$\mathbf{F}_i = - \frac{\partial U}{\partial \mathbf{r}_i} \quad (\text{III.2})$$

Cette force généralisée peut également s'exprimer en fonction de la force  $\mathbf{f}$  :

$$\mathbf{F}_i = - \frac{\partial U}{\partial \mathbf{r}} \frac{\partial \mathbf{r}}{\partial \mathbf{r}_i} = \mathbf{f} b_i(u) \quad (\text{III.3})$$

Une force  $f$  ne dérivant pas d'un potentiel sera directement donnée par ce dernier résultat.

#### b Forces uniformes

Si une densité de force  $\mathbf{f}(u)$  s'applique sur une portion  $\Delta L$  de la spline matérielle, il suffit de sommer les valeurs de  $\mathbf{f}(u)$  sur la portion, pour obtenir la contribution généralisée  $\mathbf{F}_i$  :

$$\mathbf{F}_i = \int_{\Delta L} \mathbf{f}(u) b_i(u) ds \quad (\text{III.4})$$

Dans le cas de la force de gravité s'exprimant :

$$\mathbf{f}(u) = \mu(u) \mathbf{g} \quad (\text{III.5})$$

avec  $\mu(u)$  la densité linéique de la spline matérielle, la contribution généralisée  $\mathbf{F}$  est égale à :

$$\mathbf{F} = \mathbf{g} \int_{\Delta L} \mu(u) b_i(u) \|\mathbf{r}'\| du \quad (\text{III.6})$$

Des turbulences aériennes et des courants marins peuvent également être appliqués de la sorte, un frottement visqueux étant éventuellement rajouté.

### c Frottements visqueux

Les frottements visqueux sont des frottements qui s'opposent à la vitesse. Ils servent généralement à modéliser le mouvement dans un fluide comme l'eau ou l'air. Cette raison est souvent invoquée à tort pour amortir une simulation de manière exagérée et la rendre plus stable... Un frottement visqueux  $\mathbf{f}(u)$  en un point  $\mathbf{r}(u)$  est défini comme proportionnel à sa vitesse, de même direction mais de sens opposé. Il n'intervient donc que dans un équilibre dynamique. Il s'écrit pour un point de la spline :

$$\mathbf{f} = -\eta P(u) \dot{\mathbf{r}}(u) \quad (\text{III.7})$$

$P$  étant la surface de contact entre le fluide et la spline, donc ici le périmètre de la section, et  $\eta$  la viscosité du milieu, en *Pa.s*. Pour l'eau, la viscosité est de  $1,1 \cdot 10^{-3} \text{ Pa.s}$ , et pour l'air de  $1,8 \cdot 10^{-5} \text{ Pa.s}$ . A l'aide de l'équation III.4, la contribution généralisée  $\mathbf{F}_i$ , sur une portion  $\Delta L$  de la spline matérielle, est donnée par :

$$\mathbf{F}_i = -\eta \int_{\Delta L} P(u) \dot{\mathbf{r}}(u) b_i(u) ds = -\eta \sum_{j=0}^{n-1} \frac{\Delta \mathbf{r}_j}{\Delta t} \int_{\Delta L} P(u) b_j(u) b_i(u) ds \quad (\text{III.8})$$

Nous pouvons alors remarquer que la contribution généralisée  $\mathbf{F}$  du frottement visqueux s'exprime en fonction du produit du vecteur de déplacement  $\Delta^n \mathbf{X}$  et d'une matrice  $\mathbb{D}$ , appelée *matrice de viscosité dynamique*, dont la construction est similaire à celle de  $\mathbb{M}$ . Un élément  $\mathbb{D}_{i,j}$  est donné par la relation suivante :

$$\mathbb{D}_{i,j} \begin{pmatrix} \mu(u) & 0 & 0 & 0 \\ 0 & \mu(u) & 0 & 0 \\ 0 & 0 & \mu(u) & 0 \\ 0 & 0 & 0 & I_0(u) \end{pmatrix} \int_{\Delta L} P(u) b_j(u) b_i(u) \|\dot{\mathbf{r}}\| du \quad (\text{III.9})$$

Le système dynamique II.82 devient donc :

$$\left( \frac{\mathbb{M}}{\Delta t^2} + \eta \frac{\mathbb{D}}{\Delta t} + \mathbb{K}^n \right) \Delta^{n+1} \mathbf{X} = \frac{\mathbb{M}}{\Delta t} \mathbf{V}^n + \mathbf{F} + \mathbf{P}^n \quad (\text{III.10})$$

Nous avons vu dans cette section comment appliquer des forces sur une spline matérielle. Néanmoins, leur utilisation se révèle fort peu intuitive pour la manipulation et l'imposition de contraintes de position ou d'encastrement. Les multiplicateurs de Lagrange, que nous allons maintenant aborder, se révèlent être la solution la mieux adaptée à cette problématique.

## 2 Multiplicateurs de Lagrange

De manière générale, rappelons-le, un système contraint bilatéralement peut s'écrire sous la forme d'une fonction  $f$  à minimiser et d'une autre fonction  $g$  à annuler, dépendant toutes deux des degrés de liberté  $\mathbf{q}$  du système et de leurs dérivées  $\dot{\mathbf{q}}$  :

$$\begin{cases} \text{Minimiser } f(\mathbf{q}, \dot{\mathbf{q}}) \\ \text{sujette à } g(\mathbf{q}, \dot{\mathbf{q}}) = 0 \end{cases} \quad (\text{III.11})$$

On peut montrer assez facilement (Leno4) que ce problème est équivalent à un système d'équations (dans le cas de coordonnées généralisées scalaires) :

$$\begin{cases} \frac{\partial f}{\partial q_i} - \lambda \frac{\partial g}{\partial q_i} = 0 \quad \forall i \in 0..n-1 \\ g(\mathbf{q}, \dot{\mathbf{q}}) = 0 \end{cases} \quad (\text{III.12})$$

où  $\lambda$  est appelé *multiplicateur de Lagrange*. Il se généralise à un système vectoriel à plusieurs contraintes :

$$\begin{cases} \frac{\partial f}{\partial \mathbf{q}^i} - \Lambda \cdot \frac{\partial \mathbf{g}}{\partial \mathbf{q}^i} = 0 \quad \forall i \in 0..n-1 \\ \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}) = 0 \end{cases} \quad (\text{III.13})$$

En considérant le Lagrangien comme la fonction à minimiser et en posant  $\mathbb{L}_i = \frac{\partial \mathbf{g}}{\partial \mathbf{q}^i}$ , le système dynamique III.10 devient :

$$\begin{cases} \left( \frac{\mathbb{M}}{\Delta t^2} + \eta \frac{\mathbb{D}}{\Delta t} + \mathbb{K}^n - \mathbb{L}_i^t \Lambda \right) \Delta^{n+1} \mathbf{X} = \frac{\mathbb{M}}{\Delta t} \mathbf{V}^n + \mathbf{F} + \mathbf{P}^n \\ \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}) = 0 \end{cases} \quad (\text{III.14})$$

Le système statique s'exprime de manière similaire :

$$\begin{cases} (\mathbb{K}^n - \mathbb{L}^t \Lambda) \Delta^{n+1} \mathbf{X} = \mathbf{F} + \mathbf{P}^n \\ \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}) = 0 \end{cases} \quad (\text{III.15})$$

Pour des raisons de lisibilité, nous ramenons le problème statique et le problème dynamique au système suivant :

$$\begin{cases} (\mathbb{F} - \mathbb{L}^t \Lambda) \Delta^{n+1} \mathbf{X} = \mathbf{B} \\ \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}) = 0 \end{cases} \quad (\text{III.16})$$

Comme le système d'équations se résout en déplacements  $\Delta \mathbf{X}$  et non en accélérations, nous pouvons le résoudre facilement en remplaçant la contrainte  $\mathbf{g}$  par sa différentiation :

$$\mathbf{d}\mathbf{g} = \sum_{i=0}^{n-1} \frac{\partial \mathbf{g}}{\partial \mathbf{q}^i} \mathbf{d}\mathbf{q}_i \quad (\text{III.17})$$

En discrétisant cette équation et en posant  $\mathbf{E} = \mathbf{d}\mathbf{g}$ , appelée *violation* de la contrainte, le système contraint devient :

$$\begin{cases} (\mathbb{F}^n - \mathbb{L}^t \Lambda) \Delta \mathbf{X} = \mathbf{B}^n \\ \mathbb{L} \Delta \mathbf{X} = \mathbf{E} \end{cases} \quad (\text{III.18})$$

ou encore sous forme matricielle :

$$\begin{pmatrix} \mathbb{F}^n & \mathbb{L}^t \\ \mathbb{L} & 0 \end{pmatrix} \begin{pmatrix} \Delta \mathbf{X} \\ -\Lambda \end{pmatrix} = \begin{pmatrix} \mathbf{B}^n \\ \mathbf{E} \end{pmatrix} \quad (\text{III.19})$$

Remarquons que le système contraint ainsi formé reste symétrique. Il peut aussi être mis sous forme bande, comme nous le verrons dans le chapitre IV. Sa résolution ne requiert aucun algorithme spécifique supplémentaire, contrairement à une résolution en accélération (LGM<sup>+</sup>04).

Dans la suite de cette section, nous allons exprimer différents types de contraintes utilisées dans le cadre de nos manipulations. Elles reprennent en partie les contraintes exprimées par Lenoir *et al.* (Leno4). Nous avons ajouté quelques contraintes nouvelles, notamment les contraintes d'abscisse glissante, pour simuler le comportement d'objets liés à la spline matérielle. Elles évoluent dans l'espace à une dimension de la courbe spline. D'autres contraintes, comme la contrainte de distance, de plan ou d'axes sont dédiées à l'animation en spécifiant des sous-espaces où les points contraints doivent évoluer. Pour chaque contrainte étudiée, nous précisons en détail l'utilisation de la violation pour la manipulation et expliquerons la signification du multiplicateur. Dans le cas général, celui-ci correspond à la force nécessaire pour maintenir la contrainte.

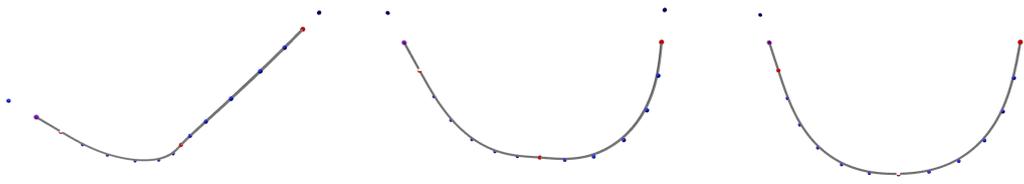


Fig. III.1 – Trois contraintes de position. Translation par violation.

### 3 Contraintes sur un point

Dans un premier temps, nous allons nous intéresser aux contraintes qui ne concernent qu'un point de la spline matérielle. Nous verrons par la suite que ces contraintes peuvent lier deux points d'une même spline ou de deux splines différentes.

#### a Contrainte de position

La *contrainte de position* (figure III.1) est la première contrainte qui vient à l'esprit. Elle consiste à fixer les degrés de liberté en un point de la spline matérielle. Violer la contrainte revient à imposer une translation du point dans l'espace et/ou à effectuer une rotation de la section. Une contrainte  $\mathbf{g}$  du point  $\mathbf{q}_0$  à l'abscisse  $u_0$  s'exprime :

$$\mathbf{g}(\mathbf{q}, u_0) = \mathbf{q}(u_0) - \mathbf{q}_0(u_0) = \mathbf{0} \quad (\text{III.20})$$

Les coordonnées généralisées  $\mathbf{q}$  peuvent varier, mais pas l'abscisse  $u_0$  : différencions  $\mathbf{g}(\mathbf{q}, u)$  par rapport aux coordonnées généralisées  $\mathbf{q}_i^1$  :

$$d\mathbf{g} = \frac{\partial \mathbf{q}}{\partial \mathbf{q}_i}(u_0) d\mathbf{q}_i = \mathbb{I} b_i(u_0) d\mathbf{q}_i \quad (\text{III.21})$$

où  $\mathbb{I}$  désigne la matrice identité. Par conséquent, la matrice  $\mathbb{L}_i$  est pour chaque coordonnée généralisée de la forme

$$\mathbb{L}_i = \begin{pmatrix} b_i(u_0) & 0 & 0 & 0 \\ 0 & b_i(u_0) & 0 & 0 \\ 0 & 0 & b_i(u_0) & 0 \\ 0 & 0 & 0 & b_i(u_0) \end{pmatrix} \quad (\text{III.22})$$

Pour ne contraindre que la position, il faut réduire le système aux trois premières équations ; pour ne contraindre que le roulis, il ne faut garder que la dernière. Le vecteur de violation  $E$  correspond à la translation imposée au point et/ou l'angle de roulis entre deux états  $n$  et  $n + 1$ . Le multiplicateur de Lagrange  $\Lambda$  correspond à la force imposée à la spline matérielle pour maintenir la position et/ou l'angle de roulis.

#### b Contrainte d'orientation

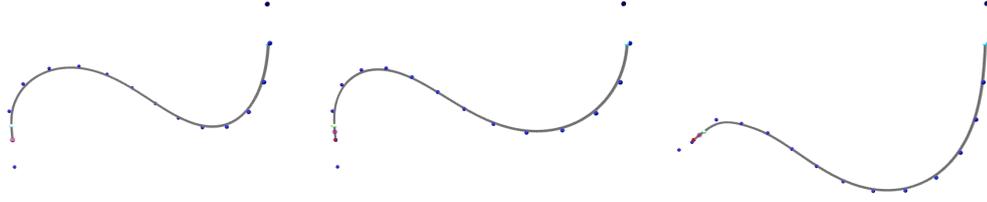
La *contrainte d'orientation* (figure III.2) complète la contrainte que nous venons de voir. Elle permet de fixer l'orientation d'une section de la spline matérielle. Violer la contrainte revient à imposer une rotation dans l'espace. Une contrainte  $\mathbf{g}$  d'orientation  $\mathbf{r}'_0$  à l'abscisse  $u_0$  s'exprime :

$$\mathbf{g}(\mathbf{r}, u_0) = \mathbf{r}'(u_0) - \mathbf{r}'_0(u_0) = \mathbf{0} \quad (\text{III.23})$$

Les coordonnées généralisées  $\mathbf{r}_i$  peuvent varier, mais pas l'abscisse  $u_0$  : différencions  $\mathbf{g}(\mathbf{r}, u)$  par rapport aux coordonnées généralisées  $\mathbf{r}_i$  :

$$d\mathbf{g} = \frac{\partial \mathbf{r}'}{\partial \mathbf{r}_i}(u_0) d\mathbf{r}_i = \mathbb{I} b'_i(u_0) d\mathbf{r}_i \quad (\text{III.24})$$

<sup>1</sup>On adopte ici la convention d'Einstein :  $\sum_{i=0}^{n-1} \frac{\partial \mathbf{q}}{\partial \mathbf{q}_i}(u_0) d\mathbf{q}_i = \frac{\partial \mathbf{q}}{\partial \mathbf{q}_i}(u_0) d\mathbf{q}_i$



**Fig. III.2** – Contraintes d'orientation. Translation de la contrainte, au milieu ; puis changement d'orientation par violation, à droite.



**Fig. III.3** – Contraintes de distance. Balancement de la spline matérielle, au milieu ; puis augmentation d'une distance, à droite.

Par conséquent, la matrice  $\mathbb{L}_i$  est pour chaque coordonnée généralisée  $\mathbf{r}_i$  de la forme :

$$\mathbb{L}_i = \begin{pmatrix} b'_i(u_0) & 0 & 0 \\ 0 & b'_i(u_0) & 0 \\ 0 & 0 & b'_i(u_0) \end{pmatrix} \quad (\text{III.25})$$

Le vecteur de violation  $E$  correspond à la rotation imposée à la section entre deux états  $n$  et  $n + 1$ . Pour la calculer, nous déterminons le vecteur d'orientation désiré  $\mathbf{t}^{n+1}$ , de même norme que  $\mathbf{r}'(u_0)$ , qui a pour support le vecteur tangent  $\mathbf{t}^n$ . La violation de direction est donc égale à :

$$E = \|\mathbf{r}'(u_0)\|(\mathbf{t}^{n+1} - \mathbf{t}^n) \quad (\text{III.26})$$

Le multiplicateur de Lagrange  $\Lambda$  correspond à la force imposée à la spline matérielle pour maintenir l'orientation. L'utilisation d'une contrainte de position et d'orientation modélise un encastrement.

Dans le même esprit, une *contrainte de courbure* peut également être construite. Elle s'exprime ainsi :

$$\mathbf{g}(\mathbf{r}, u_0) = \mathbf{r}''(u_0) - \mathbf{r}_0''(u_0) = \mathbf{0} \quad (\text{III.27})$$

L'annulation de la courbure pour une portion rectiligne de spline demande de supprimer à chaque état de simulation les équations liées de cette contrainte.

### c Contrainte de distance

La *contrainte de distance* (figure III.3) permet de restreindre l'espace de configuration d'un point de la spline matérielle dans une sphère autour d'un point de référence. Cette contrainte peut aussi avoir son utilité pour modéliser une réponse à la collision. Violier la contrainte équivaut à augmenter ou à réduire la distance. On peut aussi déplacer le point de référence entre deux pas. Une contrainte  $g$  de distance  $d$  entre la position  $\mathbf{r}_0$  à l'abscisse  $u_0$  et un point  $\mathbf{p}$  de l'espace s'exprime :

$$g(\mathbf{r}, u_0) = (\mathbf{r}(u_0) - \mathbf{p})^2 - d^2 = 0 \quad (\text{III.28})$$

Les coordonnées généralisées  $\mathbf{r}_i$  peuvent varier, mais pas l'abscisse  $u_0$  : différencions  $\mathbf{g}(\mathbf{r}, u)$  par rapport aux coordonnées généralisées  $\mathbf{r}_i$  :

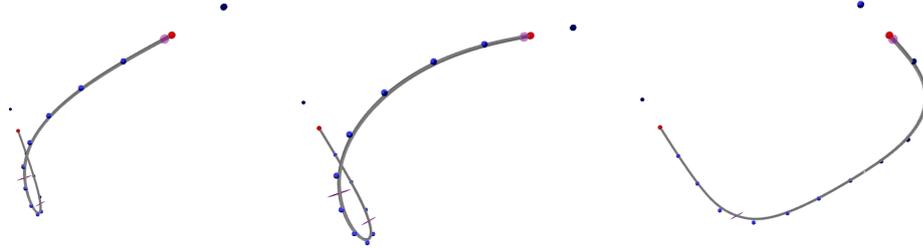
$$dg = 2 \frac{\partial \mathbf{r}}{\partial \mathbf{r}_i}(u_0)(\mathbf{r}(u_0) - \mathbf{p}) d\mathbf{q}_i = \mathbf{I}(\mathbf{r}(u_0) - \mathbf{p}) b_i(u_0) \Delta \mathbf{q}_i \quad (\text{III.29})$$

Par conséquent, la matrice  $\mathbb{L}_i$  est pour chaque coordonnée généralisée  $\mathbf{r}_i$  de la forme

$$\mathbb{L}_i = \begin{pmatrix} 2(\mathbf{r}(u_0) - \mathbf{p}) \mathbf{I} b_i(u_0) & 2(\mathbf{r}(u_0) - \mathbf{p}) \mathbf{I} b_i(u_0) & 2(\mathbf{r}(u_0) - \mathbf{p}) \mathbf{I} b_i(u_0) & 0 \end{pmatrix} \quad (\text{III.30})$$



**Fig. III.4** – Contraintes de plan. Translation de la contrainte de plan suivant sa normale par violation, au milieu ; translation de la contrainte de point fixe pour faire évoluer librement les contraintes de plan, à droite.



**Fig. III.5** – Contraintes d'axe. Violation de la contrainte d'axe dans son plan normal, au milieu ; puis évolution libre des contraintes d'axe, à droite.

Notons qu'il n'est pas nécessaire de connaître la distance  $d$  pour établir la contrainte. Si le vecteur de violation  $E$  est positif, la distance augmente, sinon elle diminue. Le multiplicateur de Lagrange  $\Lambda$  correspond à la valeur algébrique de la force, normale à la sphère, imposée à la spline matérielle pour maintenir la distance. Le sens et la direction de la force sont donnés par le vecteur  $\mathbf{r}(u_0) - \mathbf{p}$ . L'écart entre le point  $\mathbf{r}(u_0)$  de la spline et le point  $\mathbf{p}$  ne peut être nul, sinon l'équation est nulle, et la contrainte non définie.

#### d Contrainte de plan

La *contrainte de plan* (Leno4) (figure III.4) permet d'imposer à un point de rester dans un plan. La violer le translate dans un plan parallèle à ce plan. Contraindre un point  $\mathbf{r}(u_0)$  à rester dans un plan défini par une normale  $\mathbf{n}$  est formulé de la manière suivante :

$$g(\mathbf{r}, u_0) = (\mathbf{r}(u_0) - \mathbf{r}_0(u_0)) \cdot \mathbf{n} = 0 \quad (\text{III.31})$$

Les coordonnées généralisées  $\mathbf{r}$  peuvent varier, mais pas l'abscisse  $u_0$  : différencions  $g(\mathbf{r}, u)$  par rapport aux coordonnées généralisées  $\mathbf{r}_i$  :

$$d\mathbf{g} = \frac{\partial \mathbf{r}}{\partial \mathbf{r}_i}(u_0) \mathbf{n} \cdot d\mathbf{r}_i = b_i(u_0) \mathbf{n} \cdot \Delta \mathbf{r}_i \quad (\text{III.32})$$

Par conséquent, la matrice  $\mathbb{L}$  est pour chaque coordonnée généralisée  $\mathbf{r}_i$  de la forme

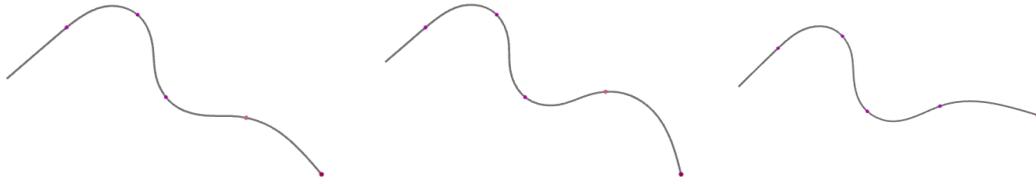
$$\mathbb{L}_i = (\mathbf{n}_x b_i(u_0) \quad \mathbf{n}_y b_i(u_0) \quad \mathbf{n}_z b_i(u_0)) \quad (\text{III.33})$$

Le vecteur de violation  $E$  correspond à l'amplitude d'une translation selon l'axe  $\mathbf{n}$ . Le multiplicateur de Lagrange  $\Lambda$  correspond à l'amplitude de la force imposée à la spline matérielle pour qu'elle reste dans le plan. La direction et le sens de la force sont donnés par  $\mathbf{n}$ .

L'utilisation conjointe d'une contrainte de plan et de distance circonscrit l'espace des positions du point considéré à un cercle.

#### e Contrainte d'axe

De la même manière qu'il faut deux plans pour définir un axe, il faut deux contraintes de plan pour définir une *contrainte d'axe* contrainte lagrangienne (Leno4) (figure III.5).



**Fig. III.6** – Contraintes de points glissants. Violation de la contrainte de droite, au milieu ; puis traction de l'extrémité gauche de la spline matérielle, à droite.

Contraindre un point  $\mathbf{r}(u_0)$  à rester sur un axe défini par deux normales  $\mathbf{n}^1$  et  $\mathbf{n}^2$ , est formulé de la manière suivante :

$$\begin{cases} g^1(\mathbf{r}, u_0) = (\mathbf{r}(u_0) - \mathbf{r}_0(u_0)) \cdot \mathbf{n}^1 = 0 \\ g^2(\mathbf{r}, u_0) = (\mathbf{r}(u_0) - \mathbf{r}_0(u_0)) \cdot \mathbf{n}^2 = 0 \end{cases} \quad (\text{III.34})$$

Les coordonnées généralisées  $\mathbf{r}$  peuvent varier, mais pas l'abscisse  $u_0$  : différencions  $g^1(\mathbf{r}, u)$  et  $g^2(\mathbf{r}, u)$  par rapport aux coordonnées généralisées  $\mathbf{r}_i$  :

$$\begin{cases} dg^1 = \frac{\partial \mathbf{r}}{\partial \mathbf{r}_i}(u_0) \mathbf{n}^1 \cdot d\mathbf{q}_i = b_i(u_0) \mathbf{n}^1 \cdot \Delta \mathbf{r}_i \\ dg^2 = \frac{\partial \mathbf{r}}{\partial \mathbf{r}_i}(u_0) \mathbf{n}^2 \cdot d\mathbf{q}_i = b_i(u_0) \mathbf{n}^2 \cdot \Delta \mathbf{r}_i \end{cases} \quad (\text{III.35})$$

Par conséquent, la matrice  $\mathbb{L}_i$  est pour chaque coordonnée généralisée  $\mathbf{r}_i$  de la forme

$$\mathbb{L}_i = \begin{pmatrix} \mathbf{n}_x^1 b_i(u_0) & \mathbf{n}_y^1 b_i(u_0) & \mathbf{n}_z^1 b_i(u_0) \\ \mathbf{n}_x^2 b_i(u_0) & \mathbf{n}_y^2 b_i(u_0) & \mathbf{n}_z^2 b_i(u_0) \end{pmatrix} \quad (\text{III.36})$$

Cette contrainte peut servir à définir une trajectoire en rendant l'axe dépendant de son paramètre. La variation d'abscisse entre deux pas  $n$  et  $n + 1$  est approximée par la distance entre la position du point concerné au pas  $n$  et celle au pas  $n + 1$ .

#### f Contrainte de point glissant

Nous adaptons ici la *contrainte glissante* (figure III.6) proposée par Lenoir *et al.* (LGM<sup>+</sup>04; LMGC04). Une contrainte glissante impose à la spline matérielle de passer par un point de l'espace, avec éventuellement une force de frottement sec en opposition. La violer consiste à translater le point de l'espace par lequel la spline matérielle doit passer. Une contrainte  $\mathbf{g}$  de point glissant s'exprime à l'aide de la position  $\mathbf{r}$  et de l'abscisse  $u$  :

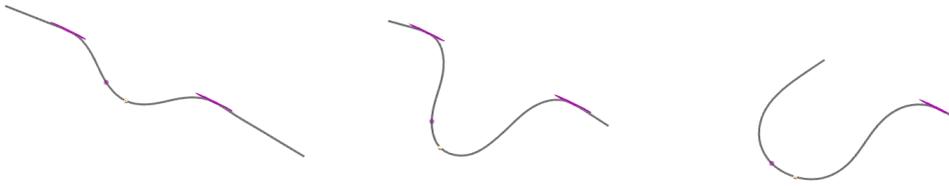
$$\mathbf{g}(\mathbf{r}, u) = \mathbf{r}(u) - \mathbf{r}_0(u_0) = \mathbf{0} \quad (\text{III.37})$$

Nous différencions  $\mathbf{g}(\mathbf{r}, u)$  par rapport aux coordonnées généralisées  $\mathbf{r}_i$ , mais aussi par rapport à  $u$  :

$$d\mathbf{g} = \frac{\partial \mathbf{r}}{\partial \mathbf{r}_i}(u) d\mathbf{r}_i + \frac{\partial \mathbf{r}}{\partial u}(u) du = \mathbb{I} b_i(u) d\mathbf{r}_i + \mathbf{r}' du \quad (\text{III.38})$$

Notons que dans leur résolution en accélération, Rémyon *et al.* ont donné à la variation d'accélération de l'abscisse, ici équivalente à  $\Delta u$ , le nom de *variable libre* (Rémo3). Par conséquent, la matrice  $\mathbb{L}_i$  est pour chaque coordonnée généralisée  $\mathbf{r}_i$  de la forme

$$\mathbb{L}_i = \begin{pmatrix} b_i(u_0) & 0 & 0 & 0 \\ 0 & b_i(u_0) & 0 & 0 \\ 0 & 0 & b_i(u_0) & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (\text{III.39})$$



**Fig. III.7** – Contraintes de points glissants orientés. Déplacement de la contrainte de position pour faire glisser la spline matérielle le long des contraintes glissantes orientées, au milieu; puis libération d'une des deux contraintes, à droite.

et pour la variable libre  $u$ , de par la symétrie, de la forme

$$\mathbb{L}_u = \begin{pmatrix} 0 & 0 & 0 & \mathbf{r}'_x \\ 0 & 0 & 0 & \mathbf{r}'_y \\ 0 & 0 & 0 & \mathbf{r}'_z \\ \mathbf{r}'_x & \mathbf{r}'_y & \mathbf{r}'_z & 0 \end{pmatrix} \quad (\text{III.40})$$

Les 3 premières composantes du vecteur de violation  $E$  permettent d'imposer une translation, comme pour une contrainte de position. La dernière composante de  $E$  affecte la variation de l'abscisse dans un sens ou dans l'autre. La contrainte glissante ainsi modélisée est parfaite : aucune résistance à l'encontre du passage de la spline matérielle n'est faite. Pour tenir compte d'un éventuel frottement visqueux  $c$ , il suffit de considérer un travail s'opposant au déplacement ; la sous-matrice  $\mathbb{L}_u$  (III.40) devient alors :

$$\begin{pmatrix} 0 & 0 & 0 & \mathbf{r}'_x \\ 0 & 0 & 0 & \mathbf{r}'_y \\ 0 & 0 & 0 & \mathbf{r}'_z \\ \mathbf{r}'_x & \mathbf{r}'_y & \mathbf{r}'_z & c \end{pmatrix} \quad (\text{III.41})$$

Après la résolution du système,  $\Delta u$  donne le décalage paramétrique de la contrainte à appliquer au calcul de  $\mathbb{L}$  au pas suivant, pour que la position du point reste fixe :

$$u \leftarrow u + \Delta u \quad (\text{III.42})$$

Le multiplicateur de Lagrange  $\Lambda$  donne deux informations : les 3 premières coordonnées donnent la force de maintien de la contrainte glissante ; celle-ci est orthogonale au vecteur tangent de la spline matérielle en l'absence de frottement. De manière similaire, on peut facilement concevoir une tangente glissante, un plan glissant, un axe glissant ou une distance glissante, même si ces contraintes ont des applications un peu moins intuitives. Remarquons que dans le cas d'une contrainte de tangente glissante, les équations peuvent être liées, du fait de la nullité de certaines composantes de  $\mathbf{r}'$ . Il faut alors éventuellement en supprimer.

### g Contrainte de point glissant orienté

Pour imposer une orientation à une contrainte glissante, on ne peut se contenter d'ajouter une contrainte d'orientation glissante à une contrainte de point glissant. Les décalages d'abscisse calculés par ces deux contraintes sont en effet différents. Pour obtenir une *contrainte de point glissant orienté* (figure III.7), nous allons plutôt ajouter une contrainte d'orientation à la contrainte de point glissant. Son abscisse paramétrique sera liée à la contrainte de point glissant. A ce stade, nous avons bien toujours une orientation fixe au niveau du point glissant. Elle ne correspond cependant pas à la position d'origine de la contrainte mais à celle du point de l'abscisse paramétrique à l'état précédent, avant que le point glissant ne s'y positionne. Il est donc en plus nécessaire d'introduire une violation

supplémentaire à la contrainte, pour imposer la même orientation en tout point. Pour ce faire, on lui associe une orientation de référence  $\mathbf{r}'_{ref}$ .

La contrainte  $\mathbf{g}$  de point glissant orienté s'exprime à l'aide de la position  $\mathbf{r}$ , de l'orientation  $\mathbf{r}'$  et de l'abscisse  $u$  :

$$\mathbf{g}(\mathbf{r}, \mathbf{r}', u) = \begin{cases} \mathbf{r}(u) - \mathbf{r}_0(u_0) = \mathbf{0} \\ \mathbf{r}'(u_0) - \mathbf{r}'_0(u_0) = \mathbf{0} \end{cases} \quad (\text{III.43})$$

Nous différencions  $\mathbf{g}(\mathbf{r}, \mathbf{r}', u)$  par rapport aux coordonnées généralisées  $\mathbf{r}_i$  et par rapport à  $u$  :

$$\mathbf{d}\mathbf{g} = \begin{cases} \mathbb{I}b_i(u)\mathbf{d}\mathbf{r}_i + \mathbf{r}'du \\ \mathbb{I}b'_i(u_0)\mathbf{d}\mathbf{r}_i \end{cases} \quad (\text{III.44})$$

Après correction de l'orientation, nous obtenons :

$$\mathbf{d}\mathbf{g} = \begin{cases} \mathbb{I}b_i(u)\mathbf{d}\mathbf{r}_i + \mathbf{r}'_{ref}du \\ \mathbb{I}b'_i(u_0)\mathbf{d}\mathbf{r}_i - \mathbf{r}'_{ref} + \mathbf{r}'_0 \end{cases} \quad (\text{III.45})$$

Le vecteur violation s'écrit donc comme la somme de la différentiation - variation imposée à la contrainte - et la correction pour maintenir l'orientation de la contrainte :

$$\mathbf{E} = \mathbf{d}\mathbf{g} + \begin{pmatrix} \mathbf{0} \\ \mathbf{r}'_{ref} - \mathbf{r}'_0 \end{pmatrix} \quad (\text{III.46})$$

Par conséquent, la matrice  $\mathbb{L}$  est pour chaque coordonnée généralisée  $\mathbf{r}_i$  de la forme

$$\mathbb{L}_i = \begin{pmatrix} b_i(u_0) & 0 & 0 \\ 0 & b_i(u_0) & 0 \\ 0 & 0 & b_i(u_0) \\ 0 & 0 & 0 \\ b'_i(u_0) & 0 & 0 \\ 0 & b'_i(u_0) & 0 \\ 0 & 0 & b'_i(u_0) \end{pmatrix} \quad (\text{III.47})$$

Pour la variable libre  $u$ , de par la symétrie et en tenant compte du frottement visqueux  $c$ , la matrice  $\mathbb{L}_u$  est de la forme :

$$\mathbb{L}_u = \begin{pmatrix} 0 & 0 & 0 & \mathbf{r}'_{ref,x} & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{r}'_{ref,y} & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{r}'_{ref,z} & 0 & 0 & 0 \\ \mathbf{r}'_{ref,x} & \mathbf{r}'_{ref,y} & \mathbf{r}'_{ref,z} & c & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (\text{III.48})$$

Cette fois, la force de maintien est obtenue en sommant les 3 premières et les 3 dernières coordonnées du vecteur de multiplicateurs de Lagrange  $\Lambda$ . La 4<sup>ème</sup> coordonnée donne le décalage d'abscisse à appliquer au calcul de  $\mathbb{L}$  au pas suivant, pour que la position du point reste fixe :

$$u \leftarrow u + \Delta u \quad (\text{III.49})$$

### h Contrainte d'abscisse glissante

Nous considérons maintenant que la contrainte glissante ne dépend plus des degrés de liberté de la spline. Nous l'appelons *contrainte d'abscisse glissante*. Elle modélise le comportement d'une particule dont la trajectoire est liée à celle de la spline et dont l'espace des configurations est restreint à la spline. Il convient donc d'ajouter une masse à la particule. L'expression de la contrainte revient finalement à une équation d'équilibre quasi-statique ou dynamique.

Une contrainte  $\mathbf{g}$  de point glissant s'exprime à l'aide de la position  $\mathbf{r}$  et de l'abscisse  $u$  :

$$\mathbf{g}(\mathbf{r}, u) = \mathbf{r}_0(u) - \mathbf{r}_0(u_0) = \mathbf{0} \quad (\text{III.50})$$

Nous différencions  $\mathbf{g}(\mathbf{r}, u)$ , cette fois seulement par rapport à  $u$  :

$$d\mathbf{g} = \frac{\partial \mathbf{r}}{\partial u}(u) du = \mathbf{r}' du \quad (\text{III.51})$$

Par conséquent, la matrice  $\mathbb{L}_u$  est pour la variable libre  $u$  de la forme

$$\mathbb{L}_u = \begin{pmatrix} 0 & 0 & 0 & \mathbf{r}'_x \\ 0 & 0 & 0 & \mathbf{r}'_y \\ 0 & 0 & 0 & \mathbf{r}'_z \\ \mathbf{r}'_x & \mathbf{r}'_y & \mathbf{r}'_z & c \end{pmatrix} \quad (\text{III.52})$$

Cette forme présente deux inconvénients. Premièrement, deux de ces équations peuvent être linéairement dépendantes, les coefficients de cette matrice pouvant très bien s'annuler. Il faut donc éliminer les équations liées à chaque pas de simulation. Deuxièmement, ce système est surdimensionné par rapport au problème : dans un espace de dimension une, une seule équation est nécessaire pour déterminer une position. C'est la raison pour laquelle nous proposons de projeter l'équation d'équilibre et la contrainte sur la tangente locale au point d'abscisse. Le système explicite (système I.62) à résoudre devient alors :

$$\left( \frac{c}{\Delta t} + \frac{m}{\Delta t^2} \right) \Delta^{n+1} u = \frac{\mathbf{r}'(u)}{\|\mathbf{r}'(u)\|} \cdot \mathbf{F} + m \frac{\Delta^n u}{\Delta t^2} \quad (\text{III.53})$$

où  $\mathbf{g}$  désigne l'accélération de gravité et  $m$  la masse de la particule. Une force de contact  $\mathbf{f}_c$  doit alors être ajoutée aux forces extérieures de la spline matérielle :

$$\mathbf{f}_c = \mathbf{F} - c \frac{\Delta^n u}{\Delta t} \frac{\mathbf{r}'(u)}{\|\mathbf{r}'(u)\|} \quad (\text{III.54})$$

La contre-partie de cette méthode est de passer en couplage lâche entre la spline matérielle et les particules, mais les résultats produits demeurent très stables et réalistes.

Toutes les contraintes que nous venons de voir peuvent non seulement s'appliquer par rapport à un point de l'espace, mais aussi par rapport à un deuxième point de la spline matérielle, voire plus.

## 4 Contraintes sur plusieurs points

Nous n'allons détailler que trois types de contraintes : les contraintes de positions et d'orientations relatives, ainsi que de distance. Elles vont nous permettre par la suite de réaliser toutes sortes d'assemblages à partir de splines matérielles.

**Fig. III.8** – Contraintes de positions relatives.

**Fig. III.9** – Contraintes d'orientations relatives.

### a Contrainte de positions relatives

La *contrainte de positions relatives* (figure III.8) lie deux points d'une ou deux splines matérielles en imposant une translation entre eux. Violier la contrainte revient à changer le vecteur de la translation. Une contrainte  $\mathbf{g}$  liant les points  $\mathbf{q}$  d'abscisse  $u_0$  et  $\mathbf{q}$  d'abscisse  $u_1$  s'exprime :

$$\mathbf{g}(\mathbf{q}, u_0, u_1) = \mathbf{q}(u_0) - \mathbf{q}(u_1) = \mathbf{q}_{10} \quad (\text{III.55})$$

où  $\mathbf{q}_{12}$  est l'écart constant entre  $\mathbf{q}(u_0)$  et  $\mathbf{q}(u_1)$ . Les coordonnées généralisées  $\mathbf{q}_i$  peuvent varier, mais pas les abscisses  $u_0$  et  $u_1$  : différencions  $\mathbf{g}(\mathbf{q}, u)$  par rapport aux coordonnées généralisées  $\mathbf{q}_i$  :

$$d\mathbf{g} = \frac{\partial \mathbf{q}}{\partial \mathbf{q}_i}(u_0) d\mathbf{q}_i - \frac{\partial \mathbf{q}}{\partial \mathbf{q}_i}(u_1) d\mathbf{q}_i = \mathbb{I}b_i(u_0) d\mathbf{q}_i - \mathbb{I}b_i(u_1) d\mathbf{q}_i \quad (\text{III.56})$$

Par conséquent, la matrice  $\mathbb{L}$  est pour chaque coordonnée généralisée de la forme

$$\mathbb{L}_i = \begin{pmatrix} b_i(u_0) - b_i(u_1) & 0 & 0 & 0 \\ 0 & b_i(u_0) - b_i(u_1) & 0 & 0 \\ 0 & 0 & b_i(u_0) - b_i(u_1) & 0 \\ 0 & 0 & 0 & b_i(u_0) - b_i(u_1) \end{pmatrix} \quad (\text{III.57})$$

Pour ne contraindre que la position, il faut réduire le système aux trois premières équations ; pour ne contraindre que le roulis, il ne faut garder que la dernière. Cependant, cela n'a vraiment de sens que lorsqu'on raccorde plusieurs splines en un point. Le vecteur de violation  $E$  correspond à la variation du vecteur de translation entre les deux points considérés, entre deux états  $n$  et  $n + 1$ . Le multiplicateur de Lagrange  $\Lambda$  donne la force imposée à la spline matérielle pour maintenir l'écart entre les points. L'intérêt principal de cette contrainte est d'être utilisée conjointement avec une contrainte d'orientations relatives.

### b Contrainte d'orientations relatives

Une *contrainte d'orientations relatives* (figure III.9) permet de garder la rotation de l'espace constante (axe et angle) entre deux points. Violier la contrainte revient à changer cette rotation. Une contrainte  $\mathbf{g}$  d'orientations relatives  $\mathbf{r}'$  d'abscisses  $u_0$  et  $u_1$  s'exprime :

$$\mathbf{g}(\mathbf{r}, u_0, u_1) = \mathbf{r}'(u_0) - \mathbf{r}'(u_1) = \mathbf{r}'_{10} \quad (\text{III.58})$$

où  $\mathbf{r}'_{12}$  est le vecteur constant liant  $\mathbf{r}'(u_0)$  et  $\mathbf{r}'(u_1)$ . Les coordonnées généralisées  $\mathbf{r}$  peuvent varier, mais pas les abscisses  $u_0, u_1$  : différencions  $\mathbf{g}(\mathbf{r}, u)$  par rapport aux coordonnées généralisées  $\mathbf{r}_i$  :

$$d\mathbf{g} = \frac{\partial \mathbf{r}'}{\partial \mathbf{r}_i}(u_0) d\mathbf{r}_i - \frac{\partial \mathbf{r}'}{\partial \mathbf{r}_i}(u_1) d\mathbf{r}_i = \mathbb{I}b'_i(u_0) d\mathbf{r}_i - \mathbb{I}b'_i(u_1) d\mathbf{r}_i \quad (\text{III.59})$$

Par conséquent, la matrice  $\mathbb{L}$  est pour chaque coordonnée généralisée  $\mathbf{r}_i$  de la forme

$$\mathbb{L}_i = \begin{pmatrix} b'_i(u_0) - b'_i(u_1) & 0 & 0 \\ 0 & b'_i(u_0) - b'_i(u_1) & 0 \\ 0 & 0 & b'_i(u_0) - b'_i(u_1) \end{pmatrix} \quad (\text{III.60})$$

Le vecteur de violation  $E$  correspond à la variation de la rotation imposée à la section entre deux états  $n$  et  $n + 1$ . Le multiplicateur de Lagrange  $\Lambda$  donne la force imposée à la spline matérielle pour maintenir la rotation entre les deux orientations.

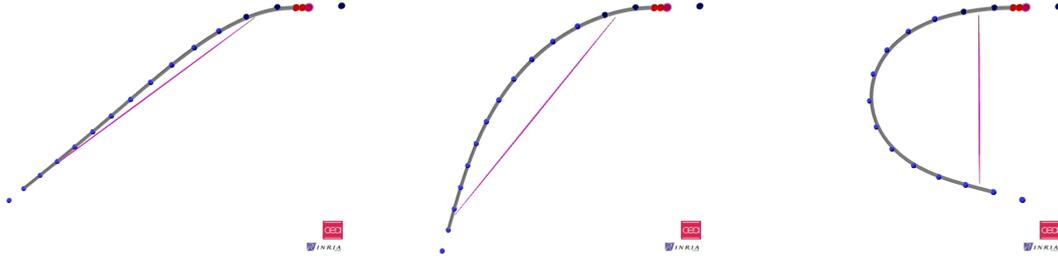


Fig. III.10 – Contraintes de distance relative.

### c Contrainte d'encastrement coplanaire relatif

L'utilisation conjointe d'une contrainte de position de translation nulle et d'orientation modélise un encastrement entre deux splines matérielles :  $\mathbf{r}_{10}$  est alors nul. Par contre si les points sont liés à distance, le problème est beaucoup plus délicat, puisque  $\mathbf{r}_{10}$  dépend des coordonnées généralisées  $\mathbf{r}_i$ . Nous ne savons résoudre que le cas particulier où  $\mathbf{r}'(u_0)$  et  $\mathbf{r}'(u_1)$  sont *coplanaires* : nous parlons alors de *contrainte d'encastrement coplanaire relatif*. Si ce n'est pas le cas, il faut exprimer  $\mathbf{r}_{10}$  en fonction des dérivées secondes, ce qui pose un problème puisqu'elles s'annulent pour une courbe droite.

Soit  $\mathbf{r}_{10}$  la combinaison linéaire de ces vecteurs :

$$\mathbf{r}_{10} = \alpha_0 \mathbf{r}'(u_0) + \alpha_1 \mathbf{r}'(u_1) \quad (\text{III.61})$$

Multiplier les deux membres de l'égalité par  $\mathbf{r}_{10}$  permet d'obtenir une deuxième équation et de déterminer  $\alpha_0$  et  $\alpha_1$ . La contrainte résultante s'exprime alors :

$$\mathbf{g}(\mathbf{r}, u_0, u_1) = \mathbf{r}(u_0) - \mathbf{r}(u_1) - \alpha_0 \mathbf{r}'(u_0) - \alpha_1 \mathbf{r}'(u_1) = \mathbf{0} \quad (\text{III.62})$$

Les coordonnées généralisées  $\mathbf{r}_i$  peuvent varier, mais pas les abscisses  $u_0, u_1$  : différencions  $\mathbf{g}(\mathbf{r}, u)$  par rapport aux coordonnées généralisées :

$$d\mathbf{g} = \frac{\partial \mathbf{r}}{\partial \mathbf{r}_i}(u_0) d\mathbf{q}_i - \frac{\partial \mathbf{r}}{\partial \mathbf{r}_i}(u_1) d\mathbf{q}_i - \alpha_0 \frac{\partial \mathbf{r}'}{\partial \mathbf{r}_i}(u_0) - \alpha_1 \frac{\partial \mathbf{r}'}{\partial \mathbf{r}_i}(u_1) \quad (\text{III.63})$$

$$d\mathbf{g} = \mathbb{I}(b_i(u_0) d\mathbf{q}_i - b_i(u_1) d\mathbf{q}_i - \alpha_0 b_i'(u_0) - \alpha_1 b_i'(u_1)) \quad (\text{III.64})$$

Par conséquent, la matrice  $\mathbb{L}_i$  est pour chaque coordonnée généralisée  $\mathbf{r}_i$  de la forme

$$\mathbb{L}_i = (b_i(u_0) - b_i(u_1) - \alpha_0 b_i'(u_0)) \mathbb{I} \quad (\text{III.65})$$

Notons qu'on peut également lier un point de l'espace à un point de la spline, pourvu que le support de la tangente passe par le point de la spline.

### d Contrainte de distance relative

La *contrainte de distance relative* (figure III.10) permet de lier deux points sur un même segment ou deux segments distincts de spline matérielles.

Violer la contrainte équivaut à augmenter ou à réduire la distance. Une contrainte  $g$  de distance  $d$  entre la position  $\mathbf{r}_0$  d'abscisse  $u_0$  et la position  $\mathbf{r}_1$  d'abscisse  $u_1$  s'exprime :

$$g(\mathbf{r}, u_0, u_1) = (\mathbf{r}(u_0) - \mathbf{r}(u_1))^2 - d^2 = 0 \quad (\text{III.66})$$

Les coordonnées généralisées  $\mathbf{r}_i$  peuvent varier, mais pas les abscisses  $u_0$  et  $u_1$  : différencions  $\mathbf{g}(\mathbf{r}, u)$  par rapport aux coordonnées généralisées  $\mathbf{r}_i$  :

$$dg = 2 \frac{\partial \mathbf{r}}{\partial \mathbf{r}_i}(u_0) (\mathbf{r}(u_0) - \mathbf{r}(u_1)) - 2 \frac{\partial \mathbf{r}}{\partial \mathbf{r}_i}(u_1) (\mathbf{r}(u_0) - \mathbf{r}(u_1)) d\mathbf{q}_i \quad (\text{III.67})$$

$$dg = \mathbf{r}(u_0) - \mathbf{r}(u_1)(b_i(u_0) - b_i(u_1))\mathbf{d}\mathbf{q}_i \quad (\text{III.68})$$

Par conséquent, la matrice  $\mathbb{L}$  est pour chaque coordonnée généralisée  $\mathbf{r}_i$  de la forme

$$\mathbb{L}_i = 2(b_i(u_0) - b_i(u_1)) \begin{pmatrix} (\mathbf{r}(u_0) - \mathbf{r}(u_1))_x & (\mathbf{r}(u_0) - \mathbf{r}(u_1))_y & (\mathbf{r}(u_0) - \mathbf{r}(u_1))_z \end{pmatrix} \quad (\text{III.69})$$

Notons que comme pour la contrainte de distance, il n'est pas nécessaire non plus de connaître la distance  $d$  pour établir la contrainte. La contrainte de distance relative ne peut également être réduite à zéro car elle est alors non définie. Si le vecteur de violation  $E$  est positif, la distance augmente, sinon elle diminue. Le multiplicateur de Lagrange  $\Lambda$  correspond à la valeur algébrique de la force, normale à la sphère, imposée à la spline matérielle pour maintenir la distance. Le sens et la direction de la force sont donnés par le vecteur  $\mathbf{r}(u_0) - \mathbf{r}(u_1)$ . L'écart entre le point  $\mathbf{r}(u_0)$  de la spline et le point  $\mathbf{p}$  ne peut être nul, sinon l'équation est nulle.

Nous avons essayé de modéliser une chaîne articulée uniquement à l'aide de contraintes en distances relatives, en supprimant les énergies d'élongation, de flexion et de torsion, mais cette tentative s'est révélée peu stable.

### e Contraintes barycentriques

Pour modéliser la forme des mailles, Nourrit *et al.* (Nou99) ont utilisé des réseaux de ressorts pour maintenir la configuration des enchevêtrements. Ces contraintes ont l'inconvénient inhérent aux ressorts, à savoir le manque de stabilité autour de l'état d'équilibre et la difficulté de choisir une raideur adéquate. Nous proposons ici l'utilisation de *contraintes barycentriques*, généralisation à  $d$  degrés de liberté  $\mathbf{q}(u_j)$  des contraintes relatives énoncées précédemment. Elles permettent de facilement imposer un motif. Violier la contrainte impose un déplacement du barycentre : translation pour des positions avec les  $\mathbf{q}(u_j)$ , rotation pour des orientations avec les  $\mathbf{q}'(u_j)$ . Soit  $G = \mathbf{q}(u_d)$  le barycentre d'un ensemble de  $d - 1$  points  $\mathbf{q}(u_j)$  et de poids  $a_j$  :

$$\mathbf{q}(u_d) = \{(\mathbf{q}(u_j), a_j)\} \quad (\text{III.70})$$

La contrainte barycentrique correspondante en position s'exprime alors :

$$\mathbf{g}(\mathbf{q}(u_j)) = \sum_{j=1}^d a_j \mathbf{q}(u_j) - \mathbf{q}(u_d) = \mathbf{0} \quad (\text{III.71})$$

Les coordonnées généralisées  $\mathbf{q}_i$  peuvent varier, mais pas les abscisses correspondantes : différencions  $\mathbf{g}(\mathbf{q}(u_j))$  par rapport aux coordonnées généralisées  $\mathbf{q}_i$  :

$$\mathbf{d}\mathbf{g} = \sum_{j=1}^{d-1} a_j \frac{\partial \mathbf{q}}{\partial \mathbf{q}_i}(u_j) \mathbf{d}\mathbf{q}_i - \frac{\partial \mathbf{q}(u_d)}{\partial \mathbf{q}_i} \mathbf{d}\mathbf{q}_i = \left( \sum_{j=1}^{d-1} a_j b_i(u_j) - b_i(u_d) \right) \mathbb{I} \mathbf{d}\mathbf{q}_i \quad (\text{III.72})$$

Par conséquent, la matrice  $\mathbb{L}_i$  est pour chaque coordonnée généralisée  $\mathbf{q}_i$  de la forme

$$\mathbb{L}_i = \left( \sum_{j=1}^{d-1} a_j b_i(u_j) - b_i(u_d) \right) \mathbb{I} \quad (\text{III.73})$$

Le multiplicateur de Lagrange  $\Lambda$  correspond à la force imposée à la ou les spline(s) matérielle(s) pour maintenir l'écart entre les points.

Si  $G$  est le centre de gravité, alors tous les coefficients  $a_j$  sont égaux à 1. Si l'on considère que  $G$  n'est pas situé sur une spline, alors sa variation par rapport aux degrés de liberté  $\frac{\partial \mathbf{q}(u_d)}{\partial \mathbf{q}_i}$  est nulle. La matrice  $L_i$  devient alors :

$$\mathbb{L}_i = \begin{pmatrix} \sum_{j=1}^{d-1} a_j b_i(u_j) & 0 & 0 & 0 \\ 0 & \sum_{j=1}^{d-1} a_j b_i(u_j) & 0 & 0 \\ 0 & 0 & \sum_{j=1}^{d-1} a_j b_i(u_j) & 0 \\ 0 & 0 & 0 & \sum_{j=1}^{d-1} a_j b_i(u_j) \end{pmatrix} \quad (\text{III.74})$$

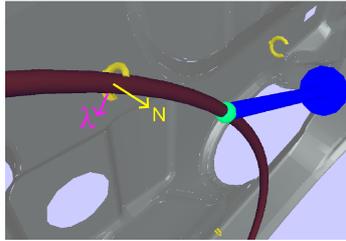


Fig. III.11 – Exemple de clip utilisé pour fixer une spline matérielle.

Notons qu’une contrainte relative sur deux points  $\mathbf{q}(u_0)$  et  $\mathbf{q}(u_1)$  équivaut à une contrainte définie par le barycentre de  $\{(\mathbf{q}(u_0), 1), (\mathbf{q}(u_1), -1)\}$

## 5 Manipulation et interaction par contraintes

Tout au long de la section précédente, nous avons défini de nombreuses contraintes mais nous n’avons que peu explicité leur utilisation dans le cadre d’une manipulation en réalité virtuelle. C’est ce à quoi nous allons maintenant nous intéresser. Nous proposons d’abord une modélisation de joints pour fixer de manière réaliste une spline matérielle sur une structure. Nous indiquerons ensuite comment coupler facilement la spline matérielle avec un autre modèle mécanique.

### a Modélisation de clips de fixation

Les contraintes lagrangiennes servent à manipuler la spline matérielle et à lui imposer des configurations géométriques, mais elles peuvent aussi permettre de l’incorporer à une structure telle qu’un bloc moteur ou une portière de voiture. Nous proposons ici une modélisation simple et efficace du comportement de clips de fixation, nous adoptons le protocole suivant. Si le test de détection de collision entre le clip et la spline matérielle est positif, une contrainte est créée au point de contact. Cela peut être une contrainte en position et en orientation si l’on considère que la spline ne va pas glisser dans le clip. Dans le cas contraire, on peut lui préférer une contrainte glissante orientée avec frottement. C’est la finesse de la détection de collision qui détermine l’adéquation à la morphologie du clip. Une primitive de type sphère à l’intérieur du clip est utilisée dans la phase de détection de collision pour savoir si la spline est située dans le clip ou pas. Elle a l’avantage de donner des résultats rapides mais elle est tributaire de la finesse de détection au niveau de la structure du clip-même. Si celle-ci est absente, une fixation dans n’importe quel sens est possible. Cependant, de par la manipulation de l’utilisateur, il est beaucoup plus probable que la spline matérielle va arriver sous un bon angle de pénétration. En pratique, nous nous sommes contentés d’une sphère intérieure et nous avons obtenu des résultats tout à fait satisfaisants. La méthode est résumée dans l’algorithme 4.

Le retrait du clip est également géré. Comme nous l’avons vu dans la dernière section, la résolution du système donne la force nécessaire au maintien de la contrainte et donc de la fixation, sous la forme du multiplicateur  $\lambda$ . En projetant cette force dans la base locale associée au joint, nous obtenons la composante  $\mathbf{N}$  suivant la direction de sortie (figure III.11). Au delà d’un certain seuil, la contrainte est supprimée et la spline matérielle est libérée. Cette phase est détaillée par l’algorithme 5.

### b Couplage

Nous allons terminer par le *couplage* entre une spline matérielle et un autre modèle physique de solides déformables. L’idée est ici d’appliquer des contraintes dures d’un côté et lâches de l’autre, pour ne pas devoir à résoudre plusieurs fois le système contraint. Il s’agit donc d’un couplage lâche. De par sa grande flexibilité et sa forte compatibilité avec

---

**Algorithme 4** : Algorithme d'insertion automatique de spline matérielle dans un joint.

---

**Entrée** : Type de contrainte à créer : *type*  
 pClip = pClipTête  
**tant que** pClip ≠ nul **faire**  
 | **si** collision entre sphère du joint et spline **alors**  
 | | i ← collision → segment  
 | | u ← collision → abscisse  
 | | // on vérifie si une contrainte de même type n'existe pas déjà  
 | | pContrainte ← Existe Contrainte (i,u,type) // algorithme 6  
 | | // auquel cas, on la récupère pour le clip ; sinon, on en crée une  
 | | **si** pContrainte = nul **alors**  
 | | | Crée Contrainte (i,u,type)  
 | | pClip → contrainte ← pContrainte  
 | | Active Clip (pClip)

---



---

**Algorithme 5** : Algorithme de libération automatique de la spline matérielle d'un joint

---

pClip = pClipTête  
**tant que** pClip ≠ nul **faire**  
 | Projette pClip → pContrainte → force sur pClip → base  
 | **si** pClip → base →  $N > pClip$  → résistance **alors**  
 | | Désactive Clip (pClip)  
 | | Supprime Contrainte (pContrainte)  
 | | pClip → pContrainte ← nul  
 | pClip ← pClip → suivant

---

les multiplicateurs de Lagrange, nous pensons que c'est à la spline matérielle de subir des contraintes bilatérales et de rendre des forces aux objets avec lesquels elle interagit. Chaque pas de simulation se déroule donc en 4 étapes :

1. Résolution du système contraint de la spline matérielle et mise à jour des positions des points de contrôles ;
2. Récupération des forces de maintien des multiplicateurs de Lagrange et application aux autres modèles ;
3. Résolution des autres modèles et mise à jour de leur configuration ;
4. Calcul des violations des contraintes de la spline matérielle correspondant aux déplacements imposés par les autres objets.

## B COLLISION

### 1 Détection de la collision

Nous avons vu au chapitre I (section a, page 47) que la collision consiste généralement, pour des objets déformables 1D, à considérer un ensemble de primitives comme des sphères ou des cylindres et à effectuer des tests entre elles. La motivation principale de ce choix est une question de rapidité de calcul de distance entre ces éléments. Nous nous sommes ici intéressés à l'utilisation de *cylindres déformés* puis des *segments de splines*. Ceux-ci peuvent permettre d'effectuer des détections précises, tout en demandant des temps de calculs du même ordre. Il est en effet possible que le coût unitaire plus élevé soit compensé par un nombre bien moins important de tests entre segments. Une autre solution est de considérer directement le maillage de la spline matérielle en effectuant des tests au niveau des triangles, mais le coût s'avère prohibitif dans une optique de temps réel.

Nous allons donc successivement détailler les calculs de distances entre sphères, cylindres et segments de spline, en prenant bien soin de préciser leur coût. Le calcul entre deux primitives est plus particulièrement développé, sa complexité augmentant plus vite que celle d'une primitive de la spline avec son environnement, généralement modélisé par des sphères ou des triangles. On considère que la spline matérielle possède une section circulaire à rayon variable  $R$ , dépendant de l'abscisse paramétrique  $u$ .

#### a Sphères

Au plus simple, un segment de spline matérielle peut être représenté par une succession de sphères contiguës de rayon  $R(u)$ . Dans le cas d'une spline, le problème n'est pas simple car l'abscisse curviligne n'est pas liée de manière régulière à l'abscisse paramétrique et elle varie d'une configuration à une autre selon les degrés de liberté de la spline. En réponse à ce problème, Lenoir *et al.* ont proposé un algorithme pour positionner des sphères contiguës le long de la spline.

Pour déterminer si deux sphères ont une intersection non nulle, il suffit de calculer la distance les séparant, qui doit être supérieure à la somme des rayons des sphères. En pratique, on calcule le carré de la distance que l'on compare au carré de la somme des rayons.

#### b Cylindres droits

Une autre approche est de considérer la spline comme une grappe de *cylindres droits* à rayon constant dont les extrémités sont communes. Des sphères peuvent venir en complément entre deux cylindres.

On considère les axes des deux cylindres droits  $C_u$  et  $C_v$ , définis chacun par un point et un vecteur directeur :

$$\begin{cases} \mathbf{X}(u) = \mathbf{X}_0 + u\Delta\mathbf{X} \\ \mathbf{Y}(v) = \mathbf{Y}_0 + v\Delta\mathbf{Y} \end{cases} \quad (\text{III.75})$$

Le carré  $D(u, v)$  de la distance  $d(u, v)$  entre les deux cylindres s'exprime :

$$D(u, v) = \sum_{\alpha} (\mathbf{Y}_0^{\alpha} - \mathbf{X}_0^{\alpha} + v\Delta\mathbf{Y}^{\alpha} - u\Delta\mathbf{X}^{\alpha})^2 \quad (\text{III.76})$$

où  $\alpha$  désigne les composantes successives des vecteurs selon  $x, y$  et  $z$ . Développons ce résultat :

$$\begin{aligned} D(u, v) = & \sum_{\alpha} (\mathbf{Y}_0^{\alpha} - \mathbf{X}_0^{\alpha})^2 + \sum_{\alpha} \Delta\mathbf{Y}^{\alpha 2} v^2 - \sum_{\alpha} \Delta\mathbf{X}^{\alpha 2} u^2 - 2 \sum_{\alpha} \Delta\mathbf{X}^{\alpha} \Delta\mathbf{Y}^{\alpha} uv \\ & + 2 \sum_{\alpha} (\mathbf{Y}_0^{\alpha} - \mathbf{X}_0^{\alpha}) \Delta\mathbf{Y}^{\alpha} v - 2 \sum_{\alpha} (\mathbf{Y}_0^{\alpha} - \mathbf{X}_0^{\alpha}) \Delta\mathbf{X}^{\alpha} u \end{aligned} \quad (\text{III.77})$$

L'expression obtenue est donc de la forme d'une cône :

$$D(u, v) = a_0 v^2 + a_1 u^2 + a_2 v u + a_3 v + a_4 u + a_5 \quad (\text{III.78})$$

Une condition nécessaire et ici suffisante pour minimiser le carré de la distance est que ses dérivées partielles s'annulent :

$$\begin{cases} \frac{\partial D^2}{\partial u} = 2a_1 u + a_2 v + a_4 = 0 \\ \frac{\partial D^2}{\partial v} = 2a_0 v + a_2 u + a_3 = 0 \end{cases} \quad (\text{III.79})$$

Ce système de deux équations à deux inconnues peut être facilement résolu. Si  $a_2^2 = 4a_0 a_1$ , les deux axes sont parallèles et la distance entre les deux axes est la même en tout point. Sinon le système possède une solution unique :

$$\begin{cases} u_{min} = \frac{2a_0 a_4 - a_2 a_3}{a_2^2 - 4a_0 a_1} \\ v_{min} = \frac{2a_1 a_3 - a_2 a_4}{a_2^2 - 4a_0 a_1} \end{cases} \quad (\text{III.80})$$

On considère qu'il y a collision entre les deux cylindres respectivement de rayon  $R_u$  et  $R_v$  si

$$D(u_{min}, v_{min}) < (R_u + R_v)^2 \quad (\text{III.81})$$

La distance entre un cylindre et un point se ramène au calcul de distance entre une droite et un point. Sa résolution est également directe. Nous voyons que comme pour les sphères, le temps de calcul est de complexité constante. Cependant, les cylindres droits n'épousent pas encore parfaitement la spline matérielle, donnant des faux positifs et/ou des faux négatifs. Une alternative est l'utilisation des cylindres généralisés.

### c Cylindres généralisés

Là encore, nous allons calculer le carré de distance entre les deux primitives, que nous allons ensuite chercher à minimiser. Mais d'abord, reprenons la définition mathématique du *cylindre généralisé* proposée par Marchal *et al.* (GM03; RGF<sup>+</sup>04). Le principe est simple. Une matrice de coordonnées homogènes (de dimension 4) permet de composer rotation, translation et homothétie. L'idée est, à partir des matrices homogènes  $\mathbb{M}_1$  et  $\mathbb{M}_2$  définies chacune pour un point de l'espace, d'effectuer l'interpolation de ces matrices à l'aide d'une fonction  $\omega$ , pour obtenir continûment une position et un repère entre ces points :

$$\mathbf{C}(u) = \omega(u)\mathbb{M}_1\mathbf{P} + (1 - \omega(u))\mathbb{M}_2\mathbf{P} \quad (\text{III.82})$$

Cette transformation a été rendue possible pour la première fois sur un GPU dans la carte d'accélération 3D *Geforce256*, la rendant beaucoup plus rapide à calculer que sur un CPU. Dans le cas des splines matérielles, la matrice  $\mathbb{M}$  en un point peut être définie à l'aide des points de contrôle  $\mathbf{r}$ , ou plus finement, des échantillons, et du repère local que nous avons défini.

Procédons maintenant au calcul de distance. Soient  $C_1$  et  $C_2$  ces deux fonctions.

$$\begin{cases} C_1(u) = \omega(u)\mathbb{M}_1^1\mathbf{P}^1 + (1 - \omega(u))\mathbb{M}_2^1\mathbf{P}^1 \\ C_2(v) = \omega(v)\mathbb{M}_1^2\mathbf{P}^2 + (1 - \omega(v))\mathbb{M}_2^2\mathbf{P}^2 \end{cases} \quad (\text{III.83})$$

Le carré  $D(u, v)$  de la distance  $d(u, v)$  entre les deux cylindres généralisés s'exprime :

$$D(u, v) = \sum_{\alpha} (C_1^{\alpha}(u, v) - C_2^{\alpha}(u, v))^2 \quad (\text{III.84})$$

Pour minimiser la distance, une condition nécessaire mais non suffisante est que ses dérivées partielles s'annulent :

$$\begin{cases} \frac{\partial D(u, v)}{\partial u} = 2 \frac{\partial C_1(u)}{\partial u} \cdot (C_1(u) - C_2(v)) \\ \frac{\partial D(u, v)}{\partial v} = -2 \frac{\partial C_2(v)}{\partial v} \cdot (C_1(u) - C_2(v)) \end{cases} \quad (\text{III.85})$$

Pour que le couple  $(u_{min}, v_{min})$  corresponde bien à un minimum, il faut que le déterminant de la matrice hessienne de  $D(u, v)$  soit strictement positif. Celle-ci s'exprime :

$$Hess(D(u, v)) = \begin{pmatrix} \frac{\partial^2 D(u, v)}{\partial u^2} & \frac{\partial^2 D(u, v)}{\partial v \partial u} \\ \frac{\partial^2 D(u, v)}{\partial u \partial v} & \frac{\partial^2 D(u, v)}{\partial v^2} \end{pmatrix} \quad (\text{III.86})$$

et son déterminant est égal à :

$$det(Hess(D(u, v))) = \frac{\partial^2 D(u, v)}{\partial u^2} \cdot \frac{\partial^2 D(u, v)}{\partial v^2} - \frac{\partial^2 D(u, v)}{\partial u \partial v} \cdot \frac{\partial^2 D(u, v)}{\partial v \partial u} \quad (\text{III.87})$$

Recherchons maintenant les couples  $(u, v)$  satisfaisant la condition imposée par le système III.85. Pour cela, calculons les dérivées partielles  $\frac{\partial C_1(u)}{\partial u}$  et  $\frac{\partial C_2(v)}{\partial v}$ . En posant  $\Delta\mathbb{M} = \mathbb{M}_1 - \mathbb{M}_2$ , nous obtenons :

$$\begin{cases} \frac{\partial C_1(u)}{\partial u} = \frac{\partial \omega(u)}{\partial u} \Delta\mathbb{M}^1 P^1 \\ \frac{\partial C_2(v)}{\partial v} = \frac{\partial \omega(v)}{\partial v} \Delta\mathbb{M}^2 P^2 \end{cases} \quad (\text{III.88})$$

Insérons les dans les dérivées partielles de  $D(u, v)$ , nous obtenons :

$$\frac{\partial D(u, v)}{\partial u} = 2 \frac{\partial \omega(u)}{\partial u} \Delta\mathbb{M}^1 P^1 \cdot (\omega(u)\mathbb{M}_1^1 P^1 + (1 - \omega(u))\mathbb{M}_2^1 P^1 - \omega(v)\mathbb{M}_1^2 P^2 + (1 - \omega(v))\mathbb{M}_2^2 P^2) \quad (\text{III.89})$$

En supposant que  $\omega$  est une fonction polynômiale de degré  $n$ , on peut ramener ce système à la forme suivante :

$$\begin{cases} \frac{\partial D(u, v)}{\partial u} = 2(\sum_{i=0}^{n-2} A_i^u u^i)(\sum_{i=0}^{n-1} B_i^u u^i - \sum_{i=0}^{n-1} C_i^u v^i) \\ \frac{\partial D(u, v)}{\partial v} = 2(\sum_{i=0}^{n-2} A_i^v v^i)(\sum_{i=0}^{n-1} B_i^v v^i - \sum_{i=0}^{n-1} C_i^v u^i) \end{cases} \quad (\text{III.90})$$

Après calculs, on aboutit aux deux formes suivantes, pour  $n = 1$  :

$$\begin{cases} \frac{\partial D(u, v)}{\partial u} = 2(a_0 u + a_1 v + a_1) \\ \frac{\partial D(u, v)}{\partial v} = 2(b_0 v + b_1 u + b_1) \end{cases} \quad (\text{III.91})$$

Nous nous retrouvons donc avec le même système que pour les cylindres droits. Le surcoût réside dans le calcul des coefficients.

Pour  $n = 2$ , le degré des équations est déjà beaucoup plus conséquent :

$$\begin{cases} \frac{\partial D(u,v)}{\partial u} = 2(a_0u^3 + a_1uv^2 + a_2u^2 + a_3v^2 + a_4uv + a_5u + a_6v + a_7) \\ \frac{\partial D(u,v)}{\partial v} = 2(b_0v^3 + b_1u^2v + b_2u^2 + b_3v^2 + b_4uv + b_5u + b_6v + b_7) \end{cases} \quad (\text{III.92})$$

Il faut alors résoudre le système suivant :

$$\begin{cases} a_0u^3 + a_1uv^2 + a_2u^2 + a_3v^2 + a_4uv + a_5u + a_6v + a_7 = 0 \\ b_0v^3 + b_1u^2v + b_2u^2 + b_3v^2 + b_4uv + b_5u + b_6v + b_7 = 0 \end{cases} \quad (\text{III.93})$$

Pour trouver  $u_{min}$  et  $v_{min}$ , on peut passer par un polynôme à une variable de degré 9, en procédant par insertions successives. Ce polynôme ne peut être résout analytiquement que dans certains cas particuliers. Cependant, l'intervalle d'étude étant restreint à  $[0, 1]$ , on peut procéder par dichotomie, à l'aide de la méthode de détermination des signes des racines réelles de Descartes.

Une fois le couple d'abscisses curvilignes  $(u_{min}, v_{min})$  correspondant au minimum de distance déterminé et la condition sur le déterminant de la hessienne vérifiée, il ne reste plus qu'à évaluer la distance entre les deux courbes. Si  $D(u_{min}, v_{min}) < (R_u + R_v)^2$ , alors il y a collision.

#### d Segments de splines

Si la *Geforce256* ne peut associer que deux matrices  $\mathbb{M}$ , les cartes récentes permettent d'associer efficacement quatre matrices. Il est donc possible de formuler le cylindre généralisé à partir des poids de la spline, qu'elle soit cardinale ou B-Spline uniforme cubique :

$$\mathbf{C}(u) = \sum b_i(u)\mathbb{M}_i\mathbf{P} \quad (\text{III.94})$$

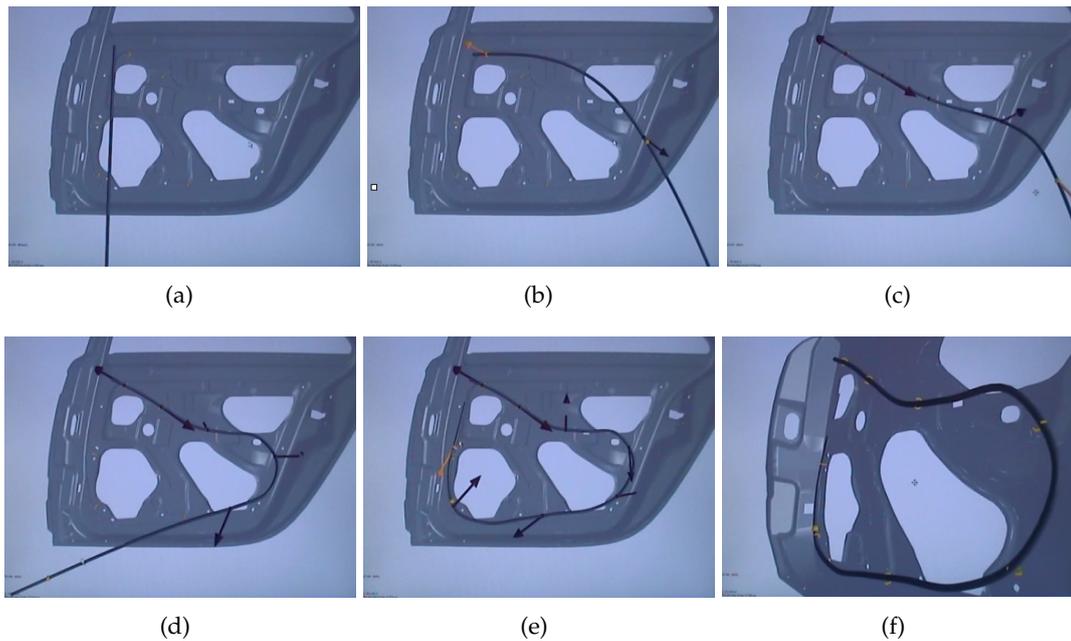
Cependant, l'utilisation des fonctions d'interpolation des splines semble d'emblée prohibitif en temps de calcul. Leur degré 3 donne un système beaucoup trop coûteux à résoudre pour le temps réel. Cette formulation est donc restreinte à l'affichage de la spline matérielle.

#### e Bilan

Nous avons évalué les temps de calcul sous Matlab, pour chacun de ces niveaux de précision. La résolution d'un polynôme de degré 9 est de l'ordre de la milli-seconde. Les deux niveaux de précision les plus élevés, segments de spline et vertex blending de degré 2, sont donc hors course. L'utilisation d'un degré 1 pour l'interpolation introduit des décalages trop importants, même pour des faibles courbures. Le gain en finesse, par rapport à l'utilisation des cylindres droits, est minime. Par conséquent, ces derniers nous semblent présenter le meilleur rapport précision/coût.

## 2 Réponse à la collision

Par manque de temps, nous n'avons pas pu réellement aborder la réponse à la collision. Néanmoins, l'utilisation de contraintes de plan pour une résolution par multiplicateurs de Lagrange semble être une bonne piste, de par son faible coût. En effet, elle ne nécessite de rajouter qu'une seule équation au système et son orientation peut évoluer : il n'est donc pas nécessaire de créer une nouvelle contrainte à chaque nouveau contact. D'une part, on peut envisager de gérer un ensemble de contraintes planaires et de n'en créer ou de n'en détruire que lorsque leur nombre évolue. D'autre part, la modélisation d'un contact continu peut éviter des discontinuités dans le système.



**Fig. III.12** – Fixation d'un câble électrique par clips sur une portière de voiture.

## C APPLICATIONS

Nous allons maintenant proposer quelques applications mettant en jeu les propositions faites dans ce chapitre.

### 1 Câble simple sur portière de voiture

Le positionnement d'un câble électrique simple sur une portière de voiture valide la stabilité de la spline matérielle, ainsi que notre modélisation de clips de fixation, en environnement de production automobile. La figure III.12 illustre quelques étapes de la mise en place du câble. D'abord maintenu à une extrémité (III.12(a)), il est vite positionné sur le premier clip à l'aide d'une deuxième contrainte de position (III.12(b)). Les flèches indiquent l'orientation et l'amplitude de la force exercée par le câble sur les clips et les points de manipulation (III.12(c), III.12(d) et III.12(e)), la couleur jaune correspondant à celle contrôlée par la souris (III.12(e)). Une fois les contraintes de manipulations supprimées, la configuration adoptée par le câble épouse de manière satisfaisante le dispositif de clips sur la portière (III.12(f)). Un temps de quelques minutes a été nécessaire pour y arriver, essentiellement dû au fait de la manipulation à l'aide d'une simple souris. Des forces sont ensuite appliquées dans diverses directions, afin de d'illustrer le caractère directionnel des joints.

### 2 Faisceau de câbles sur structure latérale de voiture

En fait, en industrie automobile, les câbles sont le plus souvent regroupés sous forme de faisceaux, afin qu'ils prennent le moins de place possible et qu'ils n'entravent pas trop le passage d'autres pièces. Dans le cadre de la modélisation de ces faisceaux, l'utilisation conjointe de contraintes relatives en position et en orientation permet de lier deux segments de spline qui partagent un point dit de *jonction*, mais avec des rayons différents. Nous avons modélisé un faisceau de câbles dont les deux extrémités présentent une arborescence différente (figure III.13). L'objectif est d'insérer toutes les extrémités dans des trous, symbolisés par des pastilles rouges. La manipulation du câble est cette fois sensiblement différente. Il ne s'agit plus seulement de positionner le câble mais aussi d'orienter ses

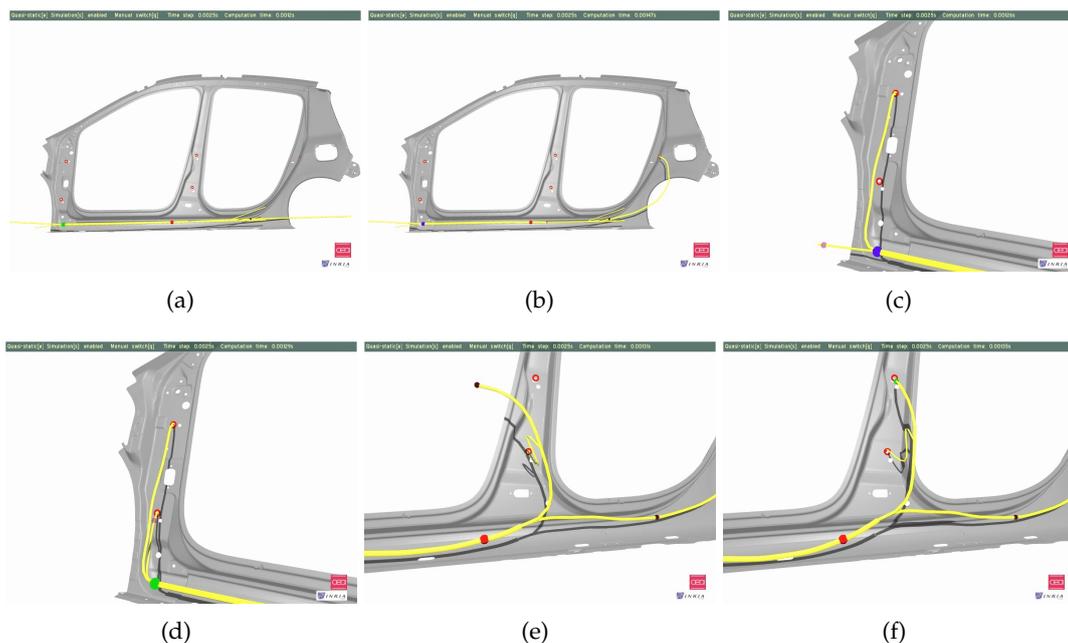


Fig. III.13 – Fixation d'un câble électrique par clips sur une portière de voiture.

extrémités. Chacune d'entre elles est donc placée en effectuant successivement des translations avec les contraintes de position, et des rotations avec les contraintes d'orientation. On observe également des amplitudes de déformation plus ou moins importantes selon le rayon du câble.

### 3 Double hélice de l'ADN

Dans un registre biologique, la spline matérielle et ses contraintes de distance relative paraissent adaptées à la modélisation de la double hélice d'ADN. Le problème des déformations de l'ADN est abondamment traité dans la littérature biologique, et sa simulation nécessite parfois des heures de calculs sur plusieurs centaines de processeurs (voir section F.2, page 171).

Nous avons construit la forme de l'ADN la plus répandue, l'ADN B, en adoptant les caractéristiques suivantes :

- un pas d'hélice a une longueur de  $3.4nm$ , pour une largeur de  $2nm$  ; elle est modélisée par une B-Spline, le positionnement des points de contrôle se faisant facilement puisque la courbure est constante le long de l'hélice ;
- les deux hélices ne sont pas exactement en opposition de phase, d'où la dénomination de *petit sillon* et de *grand sillon*. La hauteur du petit sillon est égale à  $\frac{2}{5}$  de celle du pas ;
- la raideur des hélices ne nous est pas connue, nous avons adopté une valeur arbitraire de  $500GPa$ , mais d'autres valeurs sont utilisables ;
- 10 liaisons hydrogène composent un pas d'hélice ; nous les avons représentées par des contraintes de distance relative ;
- l'ADN évoluant dans un fluide, nous pouvons négliger son poids ; la simulation est donc quasi-statique.

Il nous est également possible de modéliser la rupture des liaisons hydrogènes, sachant que les deux types de liaison (A-T et C-G) ne sont pas aussi résistantes l'une que l'autre. On sait enfin qu'il y a une valeur seuil de  $12pN$  nécessaire pour faire sauter les liaisons, mais la différence entre les deux types ne nous est pas encore connue. Pour reproduire ce comportement, il nous suffirait de supprimer la contrainte de distance correspondant à une liaison dès que la force de tension est supérieur à ce seuil. Pour le moment, nous

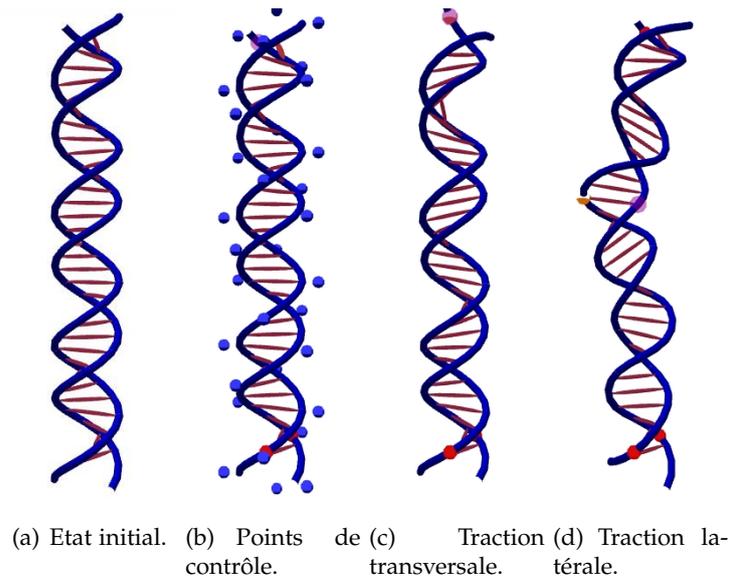


Fig. III.14 – Double hélice B-Spline, composée de 32 points de contrôle et de 35 contraintes de distance.

n'avons traité que les déformations de la double hélice. Les phénomènes de rupture de l'hélice feront l'objet d'études ultérieures. La figure III.14 illustre quelques déformations imposées à une double hélice de 3 pas, composée de 35 liaisons hydrogène. Le temps de calcul est tout à fait honorable : environ 2 ms sont nécessaires pour un pas de temps. La comparaison avec les simulateurs existants permettrait de valider la pertinence des mouvements obtenus.

#### 4 Pont

L'utilisation des contraintes de distance est également éprouvée dans la modélisation d'un pont de corde, constitué de 4 cordes principales, parallèles entre elles (figure III.15). D'une part, elles représentent les planches sur lesquelles la force d'appui des passagers va s'exercer. D'autre part, elles lient rampes et plancher. Le système obtenu est parfaitement stable. La suppression des contraintes de position à une des extrémités conduit le pont à s'effondrer de manière réaliste sur la paroi opposée du ravin.

#### 5 Collier de perles

Nous avons choisi d'illustrer l'utilisation de *contraintes d'abscisse glissante* par la simulation d'un collier de perles (figure III.16). Un faible frottement, quoique peu réaliste, permet de bien visualiser les déformations induites par le déplacement des perles sur une ficelle. C'est essentiellement le poids des perles qui lui donne sa forme.

De par le milieu 1D dans lequel évoluent les contraintes d'abscisse glissante, la détection de collision se fait simplement et efficacement. Nous nous contentons pour le moment d'une distance inférieure à la somme des rayons des deux contraintes considérées, pour valider une collision. La réponse est elle gérée de façon simple par une méthode à pénalité. Par la suite, nous la traiterons de manière plus fine en utilisant des contraintes unilatérales à base de multiplicateurs de Lagrange.

Le glissement de ces contraintes sur un ressort illustre de manière plus convaincante le couplage entre les particules et la spline matérielle (figure III.17). Elles étirent le ressort au fur et à mesure de leur descente qui va en s'accéléralant. Une fois que les particules sont sorties de la spline matérielle, celle-ci retrouve sa configuration initiale si le retour n'est pas trop brusque. Dans le cas contraire, il se peut que les spires s'emmêlent.

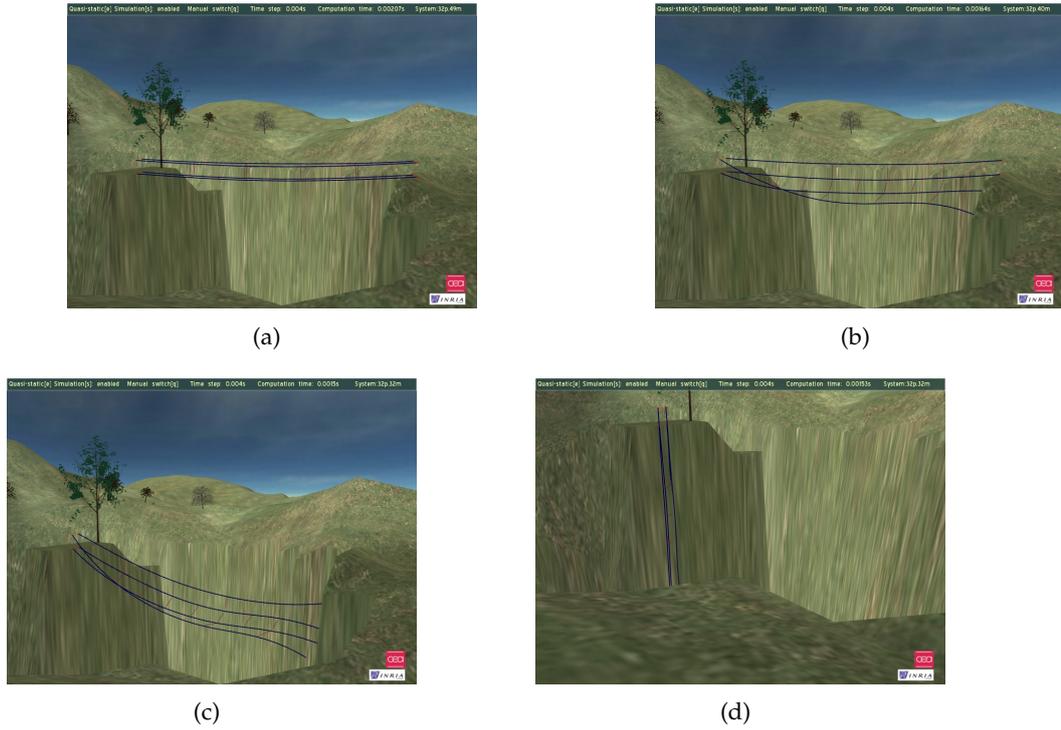


Fig. III.15 – Chute d'un pont.

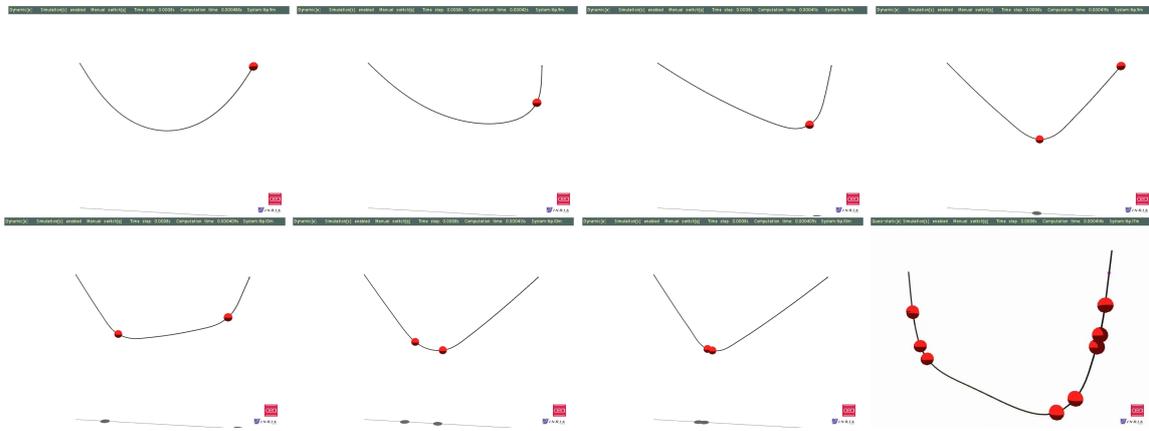
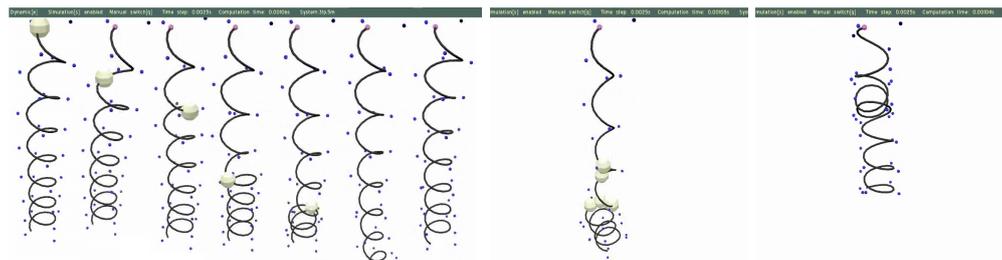


Fig. III.16 – Glissement de perles sur un collier, modélisées par des contraintes d'abscisse glissante.



(a) Etapes successives du glissement d'une particule. (b) Glissement de plusieurs particules. (c) Emmêlement consécutif.

Fig. III.17 – Glissement de contraintes d'abscisse glissante sur un ressort modélisé par une spline matérielle.

## BILAN

Dans ce chapitre, nous avons étudié les diverses façons d'interagir avec le modèle de spline matérielle. Les forces peuvent être exprimées ponctuellement ou sous forme de champs. La manipulation est cependant rendue beaucoup plus facile par l'utilisation de multiplieurs de Lagrange. Ceux-ci agissent indépendamment des paramètres mécaniques de la spline matérielle, pour imposer des positions et des orientations relatives à l'environnement ou à une autre partie de la spline matérielle. Ces contraintes permettent également d'assembler plusieurs splines entre elles et de réaliser des structures complexes comme des faisceaux de câble, un pont, ou de l'ADN. Pour cela, nous avons adapté leur résolution en déplacements : position, orientation, plan, axe, point glissant. Nous avons aussi proposé de nouvelles contraintes comme les points glissants orientés, les jonctions, les distances, les contraintes d'abscisse glissante ou les contraintes barycentriques. Enfin, elles permettent de lier la spline matérielle avec son environnement par l'utilisation des clips de fixation ou d'un couplage lâche. Dans le cadre de la détection de collision, nous avons établi sa complexité en fonction du degré des éléments représentant la spline matérielle : sphères, cylindres droits, cylindres généralisés, et même la spline elle-même. Nous n'avons cependant pas encore quantifié l'impact de l'utilisation des contraintes de Lagrange sur les performances de simulation. En effet, les travaux précédents font état du passage d'une complexité linéaire pour un système libre, à une complexité quadratique pour un système contraint, portant préjudice aux performances de manière significative.



## Chapitre IV

# Modèle géométriquement exact pour le temps-réel

*Les performances individuelles, ce n'est pas le plus important. On gagne et on perd en équipe.*

ZINÉDINE ZIDANE

**T**OUT au long des chapitres [II](#) et [III](#), nous nous sommes attachés à montrer la précision mécanique et la facilité de manipulation du modèle de spline matérielle. Nous avons considéré la simulation comme étant de complexité linéaire, sans entrer dans les détails. Si cette propriété paraît évidente pour un objet déformable 1D libre et composé d'une fibre unique, elle l'est beaucoup moins quand il est contraint ou qu'il se présente sous la forme de faisceaux. Notre objectif étant d'allier précision, interactivité et performance, nous expliquons dans ce chapitre comment nous avons obtenu une simulation rapide pour de telles configurations. D'une part, nous donnons une méthode d'assemblage de la matrice du système sous forme bande. D'autre part, nous proposons la simulation quasi-dynamique, qui allie les avantages d'une résolution quasi-statique et d'une résolution dynamique. Nous terminons par une vue d'ensemble de l'algorithme de simulation d'une spline matérielle, puis par la validation de notre méthode de construction bande, et enfin par une évaluation assez exhaustive des performances, au travers de notre implémentation.

## A ARCHITECTURE ET ALGORITHMIQUE TEMPS-RÉEL

Tout au long des chapitres II et III, nous avons considéré la simulation comme étant de complexité linéaire, sans entrer dans les détails. Nous allons maintenant justifier cette complexité, et expliquer comment réduire au maximum les calculs inutiles. Un premier point est d'opérer d'abord sur les sous-éléments du système global, les segments de spline, en limitant au maximum les calculs. Un second point est que la bande du système doit être la plus étroite possible. Cela est rendu possible par un placement optimal des points de contrôle et des contraintes, et donc des segments de spline.

### 1 Segments de splines

Comme nous le verrons plus loin dans l'évaluation des performances du modèle de spline matérielle, le remplissage du système est la partie la plus gourmande de l'algorithme. L'objectif de cette section est de minimiser le nombre d'opérations, en exploitant les nombreuses symétries. Pour ce faire, il nous faut considérer la structure d'un sous-élément du système, le segment de spline, en exploitant au mieux ses propriétés, notamment les symétries.

Chaque segment d'entre eux est entièrement déterminé par un nombre de points de contrôle égal à la localité de la spline. Dans le cas des B-Splines cubiques et des splines cardinales, elle est de 4. Quatre points de contrôle par segment sont donc nécessaires. Celui-ci se décrit par un système de la forme suivante :

$$\begin{pmatrix} \mathbb{K}_{0,0} & \mathbb{K}_{0,1} & \mathbb{K}_{0,2} & \mathbb{K}_{0,3} \\ \mathbb{K}_{1,0} & \mathbb{K}_{1,1} & \mathbb{K}_{1,2} & \mathbb{K}_{1,3} \\ \mathbb{K}_{2,0} & \mathbb{K}_{2,1} & \mathbb{K}_{2,2} & \mathbb{K}_{2,3} \\ \mathbb{K}_{3,0} & \mathbb{K}_{3,1} & \mathbb{K}_{3,2} & \mathbb{K}_{3,3} \end{pmatrix} \begin{pmatrix} \mathbf{d}\mathbf{q}_0 \\ \mathbf{d}\mathbf{q}_1 \\ \mathbf{d}\mathbf{q}_2 \\ \mathbf{d}\mathbf{q}_3 \end{pmatrix} = \begin{pmatrix} \mathbf{P}_0 \\ \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{pmatrix} \quad (\text{IV.1})$$

où  $\mathbb{K}_{i,j}$  est une sous-matrice élémentaire :

$$- \begin{pmatrix} \frac{\partial \mathbf{P}_x^i}{\partial x^j} & \frac{\partial \mathbf{P}_x^i}{\partial y^j} & \frac{\partial \mathbf{P}_x^i}{\partial z^j} & \frac{\partial \mathbf{P}_x^i}{\partial \theta^j} \\ \frac{\partial \mathbf{P}_y^i}{\partial x^j} & \frac{\partial \mathbf{P}_y^i}{\partial y^j} & \frac{\partial \mathbf{P}_y^i}{\partial z^j} & \frac{\partial \mathbf{P}_y^i}{\partial \theta^j} \\ \frac{\partial \mathbf{P}_z^i}{\partial x^j} & \frac{\partial \mathbf{P}_z^i}{\partial y^j} & \frac{\partial \mathbf{P}_z^i}{\partial z^j} & \frac{\partial \mathbf{P}_z^i}{\partial \theta^j} \\ \frac{\partial \mathbf{P}_\theta^i}{\partial x^j} & \frac{\partial \mathbf{P}_\theta^i}{\partial y^j} & \frac{\partial \mathbf{P}_\theta^i}{\partial z^j} & \frac{\partial \mathbf{P}_\theta^i}{\partial \theta^j} \end{pmatrix} \quad (\text{IV.2})$$

Au final, un segment nécessite une matrice de dimension  $16 \times 16$  et un vecteur de dimension 16.

En premier lieu, nous initialisons à zéro tous les éléments de la matrice et du vecteur du segment. Nous ajoutons ensuite chaque déformation au choix. Il est tout à fait possible de n'utiliser que la traction-compression et d'omettre la flexion ou la torsion, voire les deux. La matrice de raideur s'obtient en dérivant le vecteur de forces internes. Il est donc normal d'y retrouver des termes de ces forces. Afin d'optimiser le temps de calcul, nous les calculons conjointement. La traction-compression est la plus simple des déformations. En plus de la symétrie de la matrice du segment, elle donne une symétrie local à chaque élément  $\mathbb{K}_{i,j}$  (figure IV.1(a)). Elle n'affecte pas le roulis, tous les éléments correspondants sont nuls. L'expression de la flexion est plus complexe et ne possède pas de symétrie locale aux éléments, mais les éléments de roulis sont aussi nuls (figure IV.1(b)). Enfin, la torsion n'a pour elle que la symétrie du segment (figure IV.1(c)). Les expressions de  $\mathbb{P}$  et  $\mathbb{K}$  ont été données dans les sections 3, page 73 et b, page 78. Si le système est résout dynamiquement, la contribution de la matrice d'inertie, ainsi que celle de la vitesse et éventuellement de l'amortissement doivent être ajoutées. Une particularité importante de cette matrice est qu'elle n'est composée que d'éléments  $\mathbb{K}_{i,j}$  diagonaux. La symétrie du segment reste aussi valable. Pour toutes ces matrices, nous n'effectuons que la moitié des calculs. La symétrie ne sera appliquée qu'après assemblage du système.

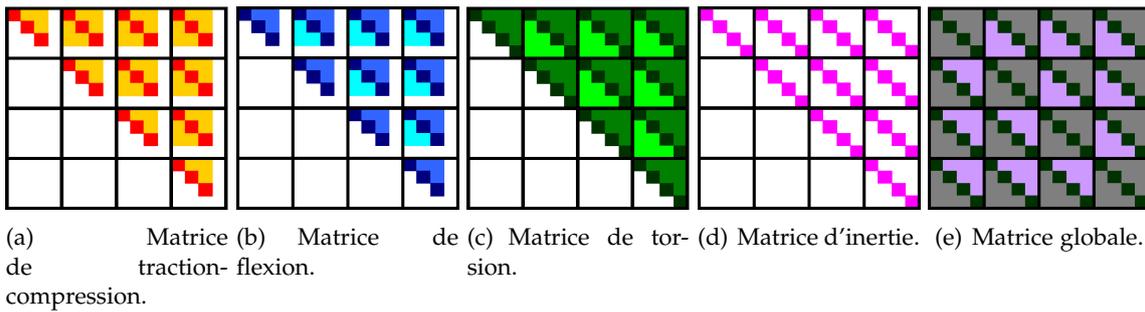


Fig. IV.1 – Mise en évidence des symétries des différentes matrices d'un segment de spline.

## 2 Placement des inconnues du système

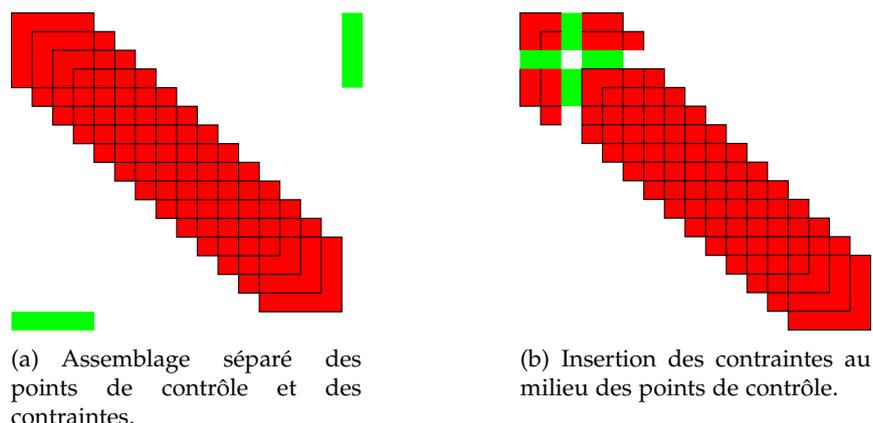
Le placement des inconnues du système conditionne la structure matricielle de manière évidente. Elles sont de deux natures : les coordonnées des points de contrôle et les contraintes, sous forme de multiplicateurs de Lagrange. Leur agencement détermine la taille de la bande matricielle. Plus celle-ci est étroite, meilleur sera le temps de calcul. Nous pouvons également quantifier les résultats en calculant son taux de remplissage. Il n'est pas nécessaire qu'il soit égal à 100% pour que la structure soit optimisée. Nous allons dans un premier temps détailler le placement des points de contrôle entre eux. Celui-ci nécessite de connaître la répartition des contraintes, mais pas de les placer en même temps. Elles sont rajoutées au système dans un deuxième temps.

### a Placement des points de contrôle

Le placement des points de contrôle dans la matrice est une tâche peu évidente et nous n'avons à ce jour pas encore validé un algorithme général. Néanmoins, à l'aide d'une connaissance préalable du problème, certaines considérations permettent d'obtenir des résultats performants dans la majorité des cas. Concernant une spline matérielle seule, ou couplée lâchement à un objet quelconque (section b, page 105), la règle qui prévaut est pour chaque segment, de placer les points de contrôle dans l'ordre. Un ensemble de  $m$  segments contigus en requiert alors  $m + 3$  points de contrôle et prend la forme illustrée dans la figure IV.3(a). Dans le cas de plusieurs splines contraintes entre elles, il s'agit d'abord d'optimiser les distances entre segments de spline dans le système. En effet, la taille de la bande sera déterminée par le plus grand écart dans le système entre deux points de contrôle, qu'ils appartiennent à un même segment ou à deux segments liés, directement entre eux ou par le biais de contraintes. Une fois que les segments liés par des contraintes relatives sont positionnés dans le système, il reste à intercaler de façon régulière les points de contrôle des segments.

### b Placement dynamique des contraintes

Le placement des contraintes dans le système matriciel conditionne également la complexité de résolution du système matriciel. L'approche de Lenoir *et al* (LMGC04) qui consiste à les regrouper après les éléments de raideur ou à les résoudre séparément est clairement sous-optimisée : la résolution du système contraint est de complexité  $o(cn^2 + cn + c^2)$ , où  $n$  est le nombre de points de contrôle et  $c$  le nombre de multiplicateurs de Lagrange, alors que le système libre est de complexité  $o(n)$ . Nous proposons une méthode de placement dynamique des contraintes dans le système, qui maintient la complexité linéaire globale du modèle de spline matérielle (figure IV.2). Elle consiste à insérer chaque contrainte au sein des segments la concernant, en vue de concentrer les éléments non nuls sur la bande du système.



**Fig. IV.2** – Deux façons différentes d’assembler le système matriciel d’un pendule. En vert, la contrainte qui fixe le pendule ; en rouge, les points de contrôle de la spline matérielle.

Toutes les contraintes sont répertoriées dans une liste chaînée maintenue ordonnée. Cette liste détermine entièrement leur placement dans le système. Afin de pouvoir facilement les intercaler ensuite entre les points de contrôle, nous leur affectons un attribut *position*, qui indique l’indice du point de contrôle après lequel mettre la contrainte. Le meilleur placement de la contrainte se situe à égale distance des points de contrôle extrêmes des segments relatifs à la contrainte. La position correspond à la moyenne des positions de ces points de contrôle dans le système, ramenée à l’entier inférieur.

L’algorithme 6 vérifie d’abord si une contrainte que l’on veut insérer dans la liste n’existe pas déjà. Cette étape est nécessaire pour éviter de se retrouver avec un système lié, qui fait échouer la résolution du système matriciel. L’algorithme 7 recherche l’endroit où placer une contrainte donnée dans la liste chaînée, caractérisé par l’adresse de la contrainte précédente. Ils sont tous les deux de complexité linéaire.

---

**Algorithme 6** : Algorithme de recherche d’une contrainte identique

---

**Entrée** : Premier point : Segment  $i_1$ , abscisse  $u_1$   
 // Si contrainte à un point, attributs du second égaux à ceux du premier

**Entrée** : Deuxième point : Segment  $i_2$ , abscisse  $u_2$

**Entrée** : type de contrainte *type*

**Sortie** : adresse de la contrainte identique si elle existe, sinon nul

$pContrainte \leftarrow pContrainteTête$

**tant que**  $pContrainte$  non nul **faire**

$position \leftarrow CalculePosition(i_1, i_2)$

**si**  $position = pContrainte \rightarrow position$  **alors**

**si**  $i = pContrainte \leftarrow i$  et  $u = pContrainte \leftarrow u$  et  $i_2 = pContrainte \leftarrow i_2$  et  $u_2 = pContrainte \leftarrow u_2$  et  $type = pContrainte \leftarrow type$  **alors**

Retourne  $pContrainte$

$pContrainteExistante \leftarrow pContrainte$

**sinon si**  $position < pContrainte \rightarrow position$  **alors**

Retourne nul

$pContrainte \leftarrow pContrainte \rightarrow suivante$

Retourne nul

---

Pour la gestion du placement des contraintes, nous distinguons deux types de contraintes : les *contraintes à abscisse fixe* qui sont fixes par rapport aux points de contrôle dans le système, et les *contraintes à abscisse variable*, dont la position dans le système évolue avec l’abscisse associée à la spline matérielle.

---

**Algorithme 7** : Algorithme de recherche de la position d'une nouvelle contrainte dans la liste chaînée.

---

**Entrée** : Premier point : Segment  $i_1$ , abscisse  $u_1$   
// Si contrainte à un point, attributs du second égaux à ceux du premier  
**Entrée** : Deuxième point : Segment  $i_2$ , abscisse  $u_2$   
**Entrée** : type de contrainte type  
**Sortie** : adresse de la contrainte précédente  
pContrainte  $\leftarrow$  pContrainteTête  
pContraintePrécédente  $\leftarrow$  nul  
position  $\leftarrow$  CalculePosition( $i_1, i_2$ )  
**tant que** pContrainte non nul **faire**  
    **si** position > pContrainte  $\leftarrow$  position **alors**  
        pContraintePrécédente  $\leftarrow$  pContrainte  
        pContrainte  $\leftarrow$  pContrainte  $\rightarrow$  suivante  
    **sinon**  
        Retourne pContraintePrécédente  
Retourne pContraintePrécédente

---

**Contraintes à abscisse fixe** Les contraintes à abscisse fixe peuvent être réparties en deux groupes : les contraintes simples portant sur un point de la spline matérielles, et les contraintes relatives, liant plusieurs points de la spline.

Pour placer correctement une contrainte simple, considérons le segment de spline associé à cette contrainte. La contrainte affecte uniquement les 4 points de contrôle du segment et aucun autre. Il apparaît donc logique de l'insérer au milieu du segment dans le système, donc entre le deuxième et le troisième point de contrôle. Cela donne, d'un point de vue matriciel statique :

$$\begin{pmatrix} \mathbb{K}_{0,0} & \mathbb{K}_{0,1} & \mathbb{L}_0^t & \mathbb{K}_{0,2} & \mathbb{K}_{0,3} \\ \mathbb{K}_{1,0} & \mathbb{K}_{1,1} & \mathbb{L}_1^t & \mathbb{K}_{1,2} & \mathbb{K}_{1,3} \\ \mathbb{L}_0 & \mathbb{L}_1 & 0 & \mathbb{L}_2 & \mathbb{L}_3 \\ \mathbb{K}_{2,0} & \mathbb{K}_{2,1} & \mathbb{L}_2^t & \mathbb{K}_{2,2} & \mathbb{K}_{2,3} \\ \mathbb{K}_{3,0} & \mathbb{K}_{3,1} & \mathbb{L}_3^t & \mathbb{K}_{3,2} & \mathbb{K}_{3,3} \end{pmatrix} \begin{pmatrix} \mathbf{dq}_0 \\ \mathbf{dq}_1 \\ \lambda \\ \mathbf{dq}_2 \\ \mathbf{dq}_3 \end{pmatrix} = \begin{pmatrix} \mathbf{P}_0 \\ \mathbf{P}_1 \\ \mathbf{E} \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{pmatrix} \quad (\text{IV.3})$$

Plusieurs contraintes peuvent bien sûr être insérées sur un même segment :

$$\begin{pmatrix} \mathbb{K}_{0,0} & \mathbb{K}_{0,1} & \mathbb{L}_0^{1t} & \mathbb{L}_0^{2t} & \mathbb{K}_{0,2} & \mathbb{K}_{0,3} \\ \mathbb{K}_{1,0} & \mathbb{K}_{1,1} & \mathbb{L}_1^{1t} & \mathbb{L}_1^{2t} & \mathbb{K}_{1,2} & \mathbb{K}_{1,3} \\ \mathbb{L}_0^1 & \mathbb{L}_1^1 & 0 & 0 & \mathbb{L}_2^1 & \mathbb{L}_3^1 \\ \mathbb{L}_0^2 & \mathbb{L}_1^2 & 0 & 0 & \mathbb{L}_2^2 & \mathbb{L}_3^2 \\ \mathbb{K}_{2,0} & \mathbb{K}_{2,1} & \mathbb{L}_2^{1t} & \mathbb{L}_2^{2t} & \mathbb{K}_{2,2} & \mathbb{K}_{2,3} \\ \mathbb{K}_{3,0} & \mathbb{K}_{3,1} & \mathbb{L}_3^{1t} & \mathbb{L}_3^{2t} & \mathbb{K}_{3,2} & \mathbb{K}_{3,3} \end{pmatrix} \begin{pmatrix} \mathbf{dq}_0 \\ \mathbf{dq}_1 \\ \lambda^1 \\ \lambda^2 \\ \mathbf{dq}_2 \\ \mathbf{dq}_3 \end{pmatrix} = \begin{pmatrix} \mathbf{P}_0 \\ \mathbf{P}_1 \\ \mathbf{E}^1 \\ \mathbf{E}^2 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{pmatrix} \quad (\text{IV.4})$$

Notons également que d'une part, les points de contrôle ne sont pas forcément contigus dans le système : d'autres points de contrôle et d'autres contraintes peuvent s'intercaler entre les points de contrôle. La contrainte est alors insérée au milieu de l'intervalle entre le deuxième et le troisième point de contrôle. L'ordre des contraintes d'un même segment n'a pas d'influence. La seule restriction est que deux contraintes portant sur la même abscisse ne doivent pas être liées, c'est-à-dire linéairement dépendantes. Toutes ces contraintes sont fixes, c'est-à-dire que leur placement dans le système par rapport aux points de contrôle est constant, et ne dépend pas des autres contraintes. Nous allons maintenant exposer comment tenir compte de l'évolution des contraintes qui dérogent à cette règle : ce sont les contraintes à abscisse variable.

**Contraintes à abscisse variable** Les *contraintes à abscisse variable* comprennent notamment les contraintes de point glissant, de point glissant orienté et d'abscisse glissante. Leur gestion passe par plusieurs vérifications. Si le placement est le même que pour les contraintes à abscisse fixe, il change à chaque fois que l'abscisse de la contrainte change de segment. Plusieurs possibilités se présentent alors. Le cas le plus fréquent est que la contrainte arrive sur un nouveau segment, et la liste doit être triée partiellement. Les contraintes qui sortent de leur segment sont marquées lors de la mise à jour de leur abscisse. L'algorithme de tri 8 permet de replacer correctement ces contraintes. La contrainte peut également être bloquée au niveau d'une jonction. Enfin, elle peut sortir de la structure. Dans ce cas, elle doit être détruite. Il est donc nécessaire qu'un graphe renseigne les connexités entre segments, pour déterminer ce qu'il faut faire.

---

**Algorithme 8** : Algorithme de remplacement des contraintes

---

```

pContrainte ← pContrainteTête
tant que pContrainte non nul faire
  si pContrainte → décalage non nul alors
    Enlève pContrainte
    Recherche pContraintePrécédente de pContrainte
    Insère pContrainte après pContraintePrécédente
    pContrainte → décalage ← nul
  pContrainte ← pContrainteTête
pContrainte ← pContrainte → suivante
Recalcule la largeur de bande du système

```

---

### c Calcul des indices des points de contrôle et des contraintes dans le système

Une fois les points de contrôle et les contraintes ordonnés séparément, il est possible de les placer de manière optimale, c'est-à-dire d'obtenir la bande la plus étroite possible, à l'aide d'un algorithme de complexité linéaire : les deux listes ne seront parcourues qu'une seule fois. Pour ce faire, nous proposons l'algorithme 9. Notons que  $D$  correspond au nombre de degrés de liberté par point de contrôle (4 avec la torsion, 3 sans).

Le placement des matrices et des forces élémentaires dans le système se fait alors très facilement grâce aux positions des points de contrôle et des contraintes données par l'algorithme. Notons qu'il faut sommer les matrices des segments là elles se chevauchent. Nous appliquons ensuite la symétrie globale du système. Nous identifions toutes les couples de coordonnées dans le système global qui correspondent à la partie supérieure des matrices de segment, en éliminant les doublons. Il ne reste qu'à copier chaque couplet dans celui obtenu en permutant ses coordonnées. Cette liste doit être reconstituée à chaque fois que la structure de la matrice évolue, c'est-à-dire que la position d'un point de contrôle ou d'une contrainte change dans la matrice.

L'ajout des éléments des contraintes dans le système se fait en parcourant leur liste chaînée. Pour chacune d'entre elles, il consiste à calculer les matrices  $\mathbb{L}_i$  et/ou la matrice  $\mathbb{L}_u$ , qui peuvent éventuellement dépendre de l'état de simulation. Les matrices  $\mathbb{L}_i$  se situent sur les lignes pContrainte ← indice à pContrainte ← indice +  $D - 1$ , et les colonnes des indices des points de contrôle correspondant au segment. La matrice  $\mathbb{L}_u$  se situe sur les mêmes lignes, mais à l'emplacement de la diagonale du système.

## 3 Résolution du système matriciel à structure bande

Pour achever la résolution du pas de temps avec une complexité linéaire, il est nécessaire de trouver la solution du système matriciel avec un algorithme qui tienne compte de la structure bande de la matrice. La matrice du système est symétrique, mais non positive

---

**Algorithme 9** : Algorithme de positionnement des points de contrôle et des contraintes dans le système.

---

**Sortie** : tableau d'indication des points de contrôle `tabIndicePointControle`

`pContrainte`  $\leftarrow$  `pContrainteTête`

`indice`  $\leftarrow$  0

**tant que** `pContrainte` non nul **faire**

**pour** `j`  $\leftarrow$  0 ; `j`  $\leq$  `pContrainte->position` ; `j`  $\leftarrow$  `j`+1 **faire**

**pour** `k`  $\leftarrow$  0 ; `k` < `D` ; `k`  $\leftarrow$  `k`+1 **faire**

`tabIndicePointControle[j][k]`  $\leftarrow$  `indice`

`indice`  $\leftarrow$  `indice`+1

`pContrainte`  $\rightarrow$  `indice`  $\leftarrow$  `indice`

`indice`  $\leftarrow$  `indice` + `pContrainte`  $\rightarrow$  `nombreDL`

`pContrainte`  $\leftarrow$  `pContrainte`  $\rightarrow$  suivante

// Plus de contraintes, on ajoute les points de contrôle restants

**pour** `j` < `n` ; `j`  $\leftarrow$  `j`+1 **faire**

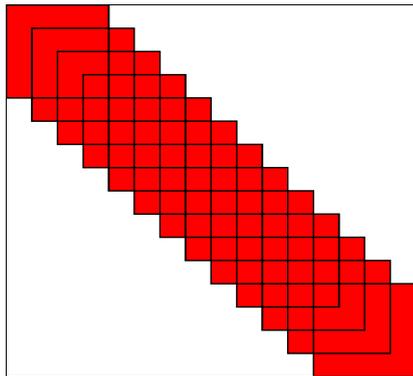
**pour** `k`  $\leftarrow$  0 ; `k` < `D` ; `k`  $\leftarrow$  `k`+1 **faire**

`tabIndicePointControle[j][k]`  $\leftarrow$  `indice`

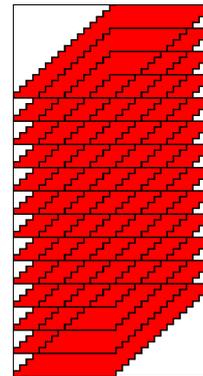
`indice`  $\leftarrow$  `indice` + 1

Retourne `pContraintePrécédente`

---



(a) Matrice bande du système, obtenue pour une spline non contrainte.



(b) Structure correspondante de stockage.

**Fig. IV.3** – Structures matricielles pour une spline matérielle de 16 points de contrôle, soit 13 segments.

à cause des multiplicateurs de Lagrange (Bar96). Un algorithme *LU Bande* paraît être la meilleure solution. Nous utilisons celui de la librairie *Intel Math Kernel Library*. Celui-ci fait appel à la fonction *dgbrtf* pour effectuer la décomposition LU, puis la fonction *dgbrts* pour résoudre le système. Il nécessite en entrée de connaître les dimensions de la bande et de travailler avec une structure matricielle adaptée, qui minimise l'espace mémoire et par conséquent les temps d'accès lors des calculs. Les colonnes de la matrice constituent les colonnes du tableau, tandis que les diagonales sont converties en lignes dans le tableau. La bande est dimensionnée par le nombre de diagonales supérieures  $ku$  et inférieures  $kl$ , en plus de la diagonale principale. Comme la matrice est symétrique, ces deux nombres sont égaux. La largeur de la structure de stockage est donc égale à  $2ku + 1$ . L'algorithme 10 permet de calculer  $ku$  et la largeur de bande. Il est effectué une fois que les points de contrôle et les contraintes sont placés dans le système (algorithme 9), et donne un résultat correct que le placement des degrés de liberté soit optimisé ou pas. Plusieurs remarques permettent d'aboutir à un calcul simple et rapide. Premièrement, le nombre de diagonales supérieures utilisé par un segment est égal à l'intervalle qu'il occupe dans le système. Il est égal à la différence entre le plus grand et le plus petit indice de point de contrôle utilisé

par le segment. Deuxièmement, les contraintes concernant un seul segment sont toujours situées à l'intérieur d'un segment dans le système. La largeur du segment en tient donc déjà compte. Par contre, les contraintes relatives à plusieurs segments peuvent utiliser plus de diagonales que les segments. Dernièrement, il est nécessaire de rajouter  $D - 1$  à l'écart entre deux points de contrôle pour obtenir l'espace occupé.

---

**Algorithme 10** : Calcul de la largeur de bande.
 

---

**Sortie** : Nombre de diagonales supérieures et inférieures  $ku$

entier  $ku \leftarrow 0$

// Calcul de l'écart maximal des segments

**pour**  $i \leftarrow 0; i < \text{nombreSegments}; i \leftarrow i+1$  **faire**

$\text{ecart} \leftarrow \max(\text{pointsSegment}[i]) - \min(\text{pointsSegment}[i])$

**si**  $\text{ecart} > ku$  **alors**

$ku \leftarrow \text{ecart}$

  // Calcul de l'écart maximal des contraintes  $p\text{Contrainte} \leftarrow p\text{ContrainteTête}$

**tant que**  $p\text{Contrainte}$  non nul **faire**

**si**  $p\text{Contrainte} \rightarrow i \neq p\text{Contrainte} \rightarrow i2$  **alors**

$\text{ecart} \leftarrow \max(\text{pointsSegment}[i], \text{pointsSegment}[i2]) - p\text{Contrainte} \rightarrow \text{indice}$

**si**  $\text{ecart} > ku$  **alors**

$ku \leftarrow \text{ecart}$

$\text{ecart} \leftarrow p\text{Contrainte} \rightarrow \text{indice} -$

$\min(\text{pointsSegment}[i], \text{pointsSegment}[i2])$

**si**  $\text{ecart} > ku$  **alors**

$ku \leftarrow \text{ecart} - D + 1$

$p\text{Contrainte} \leftarrow p\text{Contrainte} \rightarrow \text{suivante}$

$ku \leftarrow ku - D + 1$

---

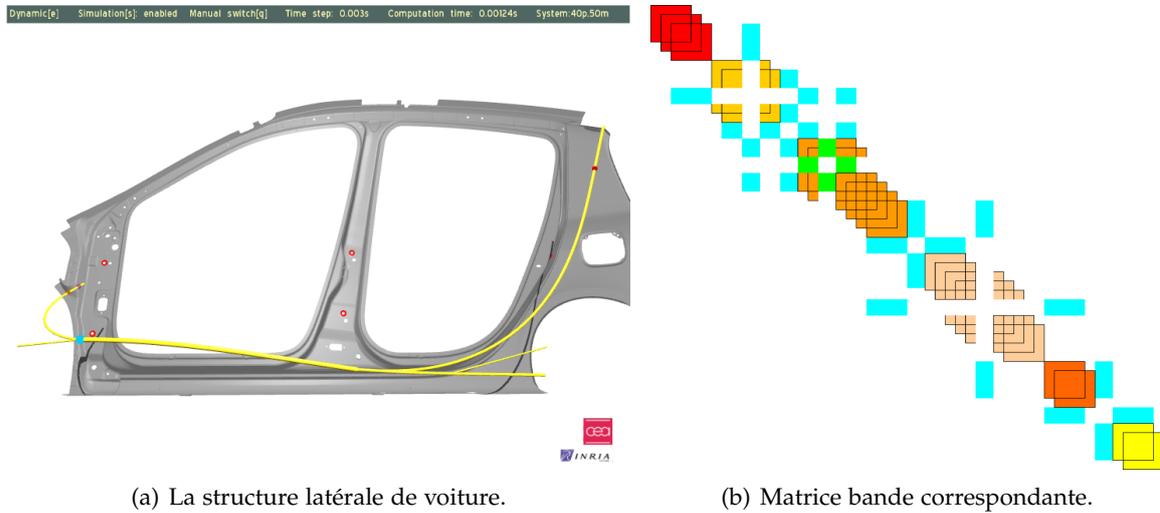
#### 4 Matrices globales obtenues

Nous donnons ici quelques exemples de matrices obtenues par notre implémentation. Elles possèdent bien toutes une structure bande. Le premier (figure IV.4) illustre l'agencement des contraintes pour modéliser des jonctions, dans le cadre de la manipulation de faisceaux menée sur une structure latérale de voiture (section 2, page 111). Pour plus de clarté sur les contraintes, nous n'avons pas entrelacé les segments.

Les deux autres exemples (figures IV.5 et IV.6) montrent l'importance de bien placer les inconnues du système. Sans un placement ordonné, les éléments seraient réparties de façon égale sur toute la matrice. La simulation de l'ADN semble être un problème particulièrement bien structuré puisque la densité de la bande est importante. Ces deux modélisations peuvent être vues comme la répétition d'un motif; en ajouter ne modifie donc pas la largeur de bande et garantit la linéarité entre la taille et le temps de calcul.

## B SIMULATION QUASI-DYNAMIQUE

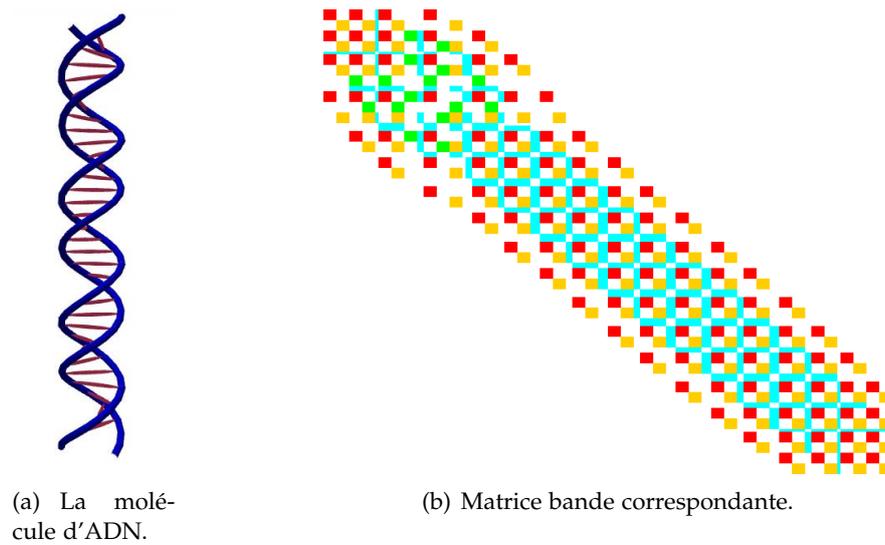
Notre système contraint est maintenant parfaitement bande. Sa résolution est donc de complexité linéaire. Un point que nous n'avons cependant pas encore abordé, et qui est important dans le contexte d'interactivité, est le choix d'une simulation dynamique ou quasi-statique.



(a) La structure latérale de voiture.

(b) Matrice bande correspondante.

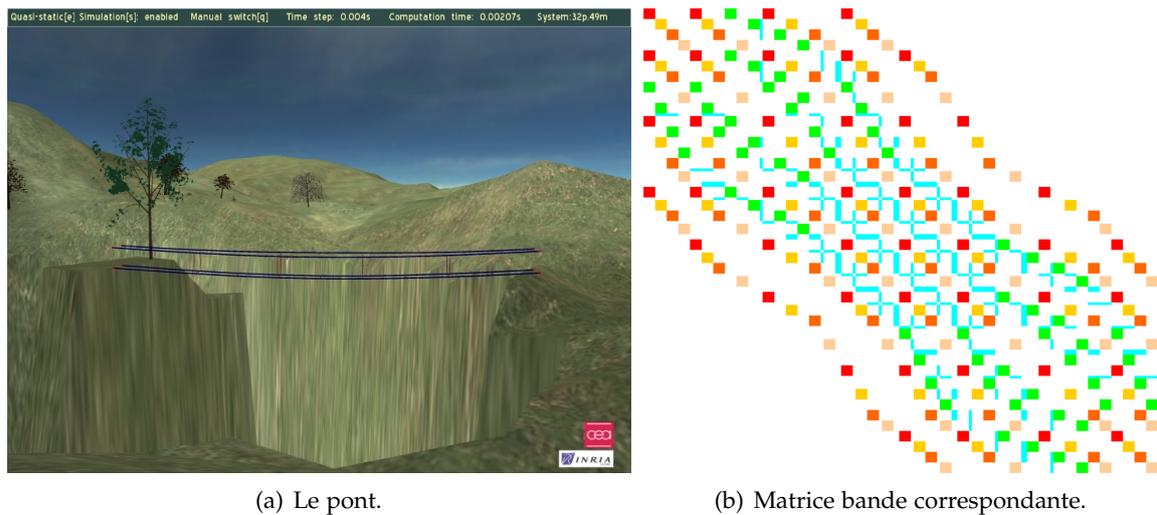
**Fig. IV.4** – Structure latérale de voiture et sa matrice bande correspondante : rouge-orange-jaune-saumon : segments distincts de spline matérielle ; vert : contraintes de position ; bleu : contraintes de jonction liant deux segments.



(a) La molécule d'ADN.

(b) Matrice bande correspondante.

**Fig. IV.5** – Une molécule d'ADN et sa matrice bande correspondante : rouge-orange-jaune-saumon : segments distincts de spline matérielle ; vert : contraintes de position ; bleu : contraintes de distance, liaisons hydrogène.



**Fig. IV.6** – Le pont et sa matrice bande correspondante : rouge-orange-jaune-saumon : segments distincts de spline matérielle ; vert : contraintes de position fixant les extrémités du pont ; bleu : contraintes de distance relative liant deux segments.

## 1 Entre simulation dynamique et simulation quasi-statique

Nous allons rappeler ici ce qu’impliquent ces deux types de simulation, en précisant leurs avantages et leurs inconvénients dans le cadre de la réalité virtuelle. Nous allons également montrer qu’elles sont relativement proches, rendant possible leur utilisation combinée.

### a Simulation dynamique

La simulation dynamique d’un objet déformable a pour vocation de reproduire tous ses mouvements et ses déformations au cours du temps. Nous avons vu précédemment qu’un schéma d’intégration temporelle est nécessaire pour résoudre numériquement un tel problème. Le principe est de calculer pour un intervalle de temps, appelé *pas de temps*, l’évolution de la configuration de l’objet, incluant déplacement et déformation. Les schémas explicites sont une solution assez fidèle à la réalité mais souffrent d’un manque important de stabilité pour les problèmes raides. Les schémas implicites constituent une réponse beaucoup plus stable, en contrepartie de l’introduction d’une viscosité et d’un filtre rigidifiant, respectivement proportionnels au pas de temps et à son carré. Le réalisme de la simulation est donc tributaire de la taille du pas de temps. Dans le cadre du temps-réel, une contrainte de taille s’ajoute : tous les calculs nécessaires au cours d’un pas de temps, qu’ils soient mécaniques ou non, doivent être effectués en un temps inférieur à celui-ci. Dans le cas contraire, la simulation accumule du retard et n’est alors plus cohérente avec la manipulation. La difficulté de modéliser correctement les frottements secs et visqueux donne soit des résultats suramortis, soit des oscillations parasites qui peuvent rendre la manipulation difficile en simulation dynamique.

Une alternative à celle-ci est beaucoup utilisée pour le temps interactif : c’est la simulation quasi-statique. Elle élimine tous les inconvénients inhérents au temps.

### b Simulation quasi-statique

Une simulation *quasi-statique* est une succession d’états d’équilibres statiques. Toutes les variables du problème sont donc indépendantes du temps ; les vitesses et les accélérations sont nulles. L’interactivité n’est plus limitée qu’au temps de calcul et elle est donc beaucoup plus facilement garantie. La fréquence de la simulation est égale au nombre d’équilibres statiques calculés en une seconde. Un objet ainsi simulé nécessite cependant d’être suffisamment contraint pour être défini. Dans le cas d’un objet 1D, il s’agit d’un

encastrement. Cela ne pose pas de problème pour des poutres encastrees, mais rend impossible la simulation d'un cable libre. Le comportement physique est egalement altere. Il l'est d'autant plus que les deformations et les deplacements sont rapides.

Chaque type de simulation possede donc des avantages et des inconvenients propres. Pourtant, nous allons voir que les simulations dynamique et quasi-statique sont tres proches.

### c Lien entre les simulations dynamiques et quasi-statiques

Dans le chapitre II, nous avons obtenu les equations d'equilibre de la spline materielle pour l'equilibre dynamique (equation II.82) :

$$\left(\frac{\mathbb{M}}{\Delta t^2} + \mathbb{K}^n\right) \Delta^{n+1}\mathbf{X} = \frac{\mathbb{M}}{\Delta t}\mathbf{V}^n + \mathbf{F} + \mathbf{P}^n \quad (\text{IV.5})$$

et l'equilibre quasi-statique (equation II.83) :

$$\mathbb{K}^n \Delta \mathbf{X} = \mathbf{F} + \mathbf{P}^n \quad (\text{IV.6})$$

Si l'on fait tendre le pas de temps  $\Delta t$  vers l'infini, l'equation II.82) tend de maniere evidente vers II.83. Pour etudier ce comportement, nous avons fait varier le pas de temps de la simulation d'un simple pendule, de 0,5ms a 1s. Comme l'on pouvait s'y attendre, au fur et a mesure que le pas de temps augmente, le pendule est de plus en plus raide et amorti. Notons que le schema Euler-Implicite linearise est extremement stable puisqu'il converge toujours, meme pour un pas de temps d'une seconde. Pour un tel pas de temps, la simulation dynamique est quasiment identique a la simulation statique, ou le pendule s'immobilise quasiment instantanement a sa position d'equilibre vertical.

Partant de ce constat et de celui que les equations d'equilibre dynamique et quasi-statique sont tres proches, nous avons eu l'idee de les utiliser conjointement. Generalement, un choix est fait avant le debut de la simulation et l'on doit ensuite s'y tenir. Nous avons alors experimente le passage de l'un a l'autre, au cours de la manipulation. Il se fait de maniere douce et stable. Nous avons appele une telle simulation *quasi-dynamique*.

### d Transitions entre dynamique et quasi-statique

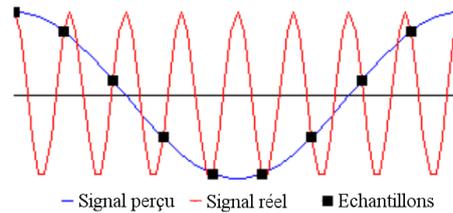
Le passage de l'un a l'autre se fait facilement, il suffit de tenir compte de quelques conditions. Le sens quasi-statique vers dynamique est toujours possible. D'un point de vue systeme matriciel, le passage du quasi-statique au dynamique consiste a rajouter l'inertie avec la matrice  $\mathbb{M}$ . La vitesse de l'etat precedent  $\mathbf{V}^n$  doit egalement etre precisee. On pourrait se contenter de la prendre nulle, mais elle traduirait mal le mouvement induit par l'utilisateur. Nous choisissons de prendre le dernier temps de calcul  $\Delta t_c$  quasi-statique pour la calculer :

$$\mathbf{V}^n = \frac{\Delta \mathbf{q}}{\Delta t_c} \quad (\text{IV.7})$$

Les forces externes  $\mathbf{F}$  et internes  $\mathbf{P}$ , ainsi que les contraintes par multiplicateurs de Lagrange restent inchangees.

Le passage du dynamique au quasi-statique n'est lui pas toujours possible. Il faut en effet que le systeme soit suffisamment contraint pour qu'il soit defini. D'autre part, la spline materielle peut etre en train d'evoluer dynamiquement a ce moment-la, et la configuration statique correspondante en etre fort eloignee. La transition est cependant assuree par une descente de Newton incomplete. Elle consiste a diviser l'amplitude des deplacements donnes par la resolution du systeme (en pratique par 3), voire a la majorer, en s'assurant que les rapports entre les amplitudes des deplacements restent inchanges.

En fait, le probleme principal reside dans le choix des criteres pour le passage de l'un a l'autre. C'est ce point que nous allons maintenant aborder.



**Fig. IV.7** – Le signal réel et le signal perçu : un échantillonnage trop faible donne une mauvaise image de la réalité.

## 2 Heuristiques de passage

Généralement, effectuer une simulation temps-réel requiert non seulement un calcul efficace de la mécanique, mais aussi de toutes les autres tâches : rendu, interface haptique, etc. Elles présentent des temps de calcul variables dont il faut tenir compte. La configuration de la spline matérielle entre aussi en jeu, selon qu'elle est suffisamment contrainte pour être simulée de manière quasi-statique. Dans cette sous-section, nous détaillons deux classes d'heuristiques pour effectuer le passage de la simulation dynamique à la simulation quasi-statique, et réciproquement : l'une est fondée sur le temps de calcul, l'autre sur le nombre de contraintes du système.

Le passage du dynamique au quasi-statique impose que le système soit suffisamment contraint pour qu'il soit défini. Nous considérons qu'il l'est si au moins deux points de la spline matérielle, ou un point et son orientation sont imposés. Pour chaque pas de simulation, nous vérifions donc si ces conditions sont respectées. Dans le cas contraire, la simulation passe obligatoirement en équilibre dynamique. Sinon, les critères qui suivent prévalent.

L'intégration dynamique impose que tous les calculs d'un pas de temps de simulation soient faits dans cet intervalle. Dans le cas contraire, du retard est accumulé et un décalage temporel se produit entre la manipulation et les déformations. Le choix du pas de temps n'est pas facile. Il dépend de plusieurs critères importants. Le premier est le *théorème de Nyquist-Shannon* et porte sur la conversion d'un signal analogique en une forme numérique :

**Théorème IV.1** *la fréquence d'échantillonnage d'un signal doit être égale ou supérieure au double de la fréquence maximale contenue dans ce signal, appelée fréquence de coupure.*

Le pas de temps détermine la résolution maximale en fréquence. En dessous de la fréquence de coupure, l'information est perdue [IV.7](#).

Le second critère porte sur la *fréquence propre* de la structure. Lorsque l'on abandonne un système stable préalablement écarté de sa position d'équilibre, il y retourne, généralement à travers une série d'oscillations propres. Celles-ci se produisent aux fréquences propres du système. Si le système n'est pas trop amorti, une excitation sinusoïdale est particulièrement amplifiée au voisinage de ces fréquences propres, c'est ce qu'on appelle la résonance. Les fréquences propres dépendent des paramètres mécaniques et de la géométrie de la structure, à savoir le module d'Young et la masse volumique, la section, et les conditions aux limites. Wu *et al.* ([WDGT01](#)) ont proposé une formule pour obtenir le pas de temps limite  $\Delta t_{crit}$ , pour un modèle à éléments finis, dans le but d'obtenir un comportement stable avec des schémas explicites :

$$\Delta t_{crit} = h \sqrt{\frac{2\rho}{E}} \quad (\text{IV.8})$$

où  $h$  est lié à la dimension des éléments,  $\rho$  est la densité du matériau, et  $E$  son module d'Young.

Pour déterminer les fréquences de résonance, on procède à une *analyse modale* (voir section c, page 41) : on recherche les pulsations  $\omega$  qui maximisent la fonction de transfert du système. La pulsation est liée à la fréquence par la relation  $\omega = 2\Pi f$ .

Dans le cas d'un système conservatif, la fonction de transfert s'écrit :

$$H(\omega) = \frac{1}{\mathbb{K} - \mathbb{M}\omega^2} \quad (\text{IV.9})$$

Le gain correspondant est

$$H(\omega) = \frac{1}{|\mathbb{K} - \mathbb{M}\omega^2|} \quad (\text{IV.10})$$

Les pulsations propres  $\omega_0$  correspondent au maximum de gain. Elles sont donc solutions de

$$\det(\mathbb{K} - \mathbb{M}\omega_0^2) = 0 \quad (\text{IV.11})$$

Si le système est amorti, la fonction de transfert devient :

$$H(\omega) = \frac{1}{\mathbb{K} - \mathbb{M}\omega^2 + i\mathbb{B}\omega} \quad (\text{IV.12})$$

Le gain est

$$|H| = \frac{1}{\sqrt{(\mathbb{K} - \mathbb{M}\omega^2)^2 + \mathbb{B}^2\omega^2}} \quad (\text{IV.13})$$

Les pulsations propres  $\omega_0$  dépendent alors de la matrice d'amortissement  $\mathbb{B}$ , et sont solution du système suivant :

$$\det((\mathbb{K} - \mathbb{M}\omega^2)^2 + \mathbb{B}^2\omega^2) = 0 \quad (\text{IV.14})$$

Ces calculs sont néanmoins coûteux et doivent être refaits au moins à chaque fois que les contraintes appliquées sur la spline évoluent. Plus elles sont nombreuses, plus les fréquences propres sont élevées et plus le système sera suramorti si le pas de temps n'évolue pas. Nous proposons une heuristique simplifiée pour le mettre à jour. Le pas de temps  $\Delta t_0$  correspondant à une configuration libre est évalué au début de la simulation. Puis, à chaque pas de temps, nous évaluons la longueur maximale  $l_{libre}$  non contrainte de la spline matérielle. Le rapport de cette longueur avec la longueur initiale  $l_{initiale}$  donne le facteur à appliquer au pas de temps initial, pour obtenir le pas de temps courant :

$$\Delta t = \frac{l_{libre}}{l_{initiale}} \Delta t_0 \quad (\text{IV.15})$$

L'augmentation du nombre de contraintes sur la spline matérielle augmente de plus son temps de calcul. Au-delà d'un certain nombre, celui-ci devient plus important que le pas de temps et le temps-réel n'est plus garanti. Il devient alors nécessaire de passer en simulation quasi-statique. Lorsque le temps de calcul redescend, il est également important de pouvoir repasser en simulation dynamique. Nous proposons donc d'évaluer à chaque pas de simulation le temps de calcul  $\Delta t_c$ , pour anticiper le moment où le temps-réel ne sera plus respecté.

S'il dépasse un certain pourcentage du pas de temps, la simulation passe en quasi-statique. En pratique, nous avons pris 95%. Pour éviter des changements permanents, nous proposons d'utiliser un cycle hysteresis, en introduisant une plus petite valeur pour passer du quasi-statique au dynamique (ici 90%). Enfin, nous commençons la simulation en dynamique. L'algorithme 11 et le schéma IV.8 résument la simulation quasi-dynamique. Notons que cet algorithme n'est valable que dans le cas d'une architecture logicielle multi-processus. Dans le cas contraire, il faut tenir compte de toutes les autres tâches du CPU dans le temps de calcul  $\Delta t_c$ .

**Algorithme 11** : Algorithme quasi-dynamique

---

```

 $\Delta t_c \leftarrow 0.9\Delta t_0$ 
Simulation dynamique  $\leftarrow$  true
tant que Simulation faire
  Commence la mesure du temps de calcul ( $\Delta t_c$ )
   $\Delta t \leftarrow \frac{f_{\text{libre}}}{f_{\text{initiale}}}\Delta t_0$  si Simulation statique alors
    |  $\mathbf{V} \leftarrow \frac{\Delta \mathbf{q}}{\Delta t_c}$ 
  si système singulier alors
    | Simulation  $\leftarrow$  dynamique
  si  $\Delta t_c > 0.95\Delta t$  alors
    | Simulation  $\leftarrow$  statique
  sinon
    | si  $\Delta t_c < 0.9\Delta t$  alors
      | | Simulation  $\leftarrow$  dynamique
    |
  Calcule la mécanique
  Finis la mesure du temps de calcul ( $\Delta t_c$ )
  Dors( $\Delta t - \Delta t_c$ )

```

---

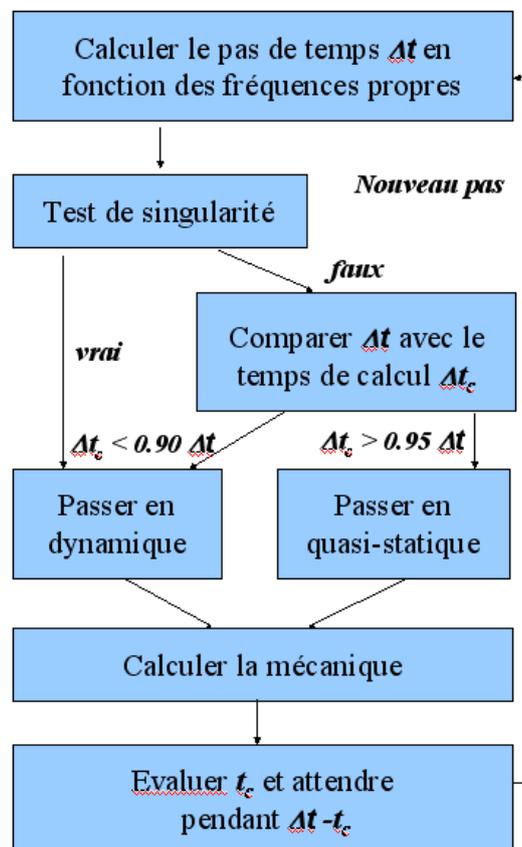


Fig. IV.8 – Schéma récapitulatif de l'algorithme quasi-dynamique.

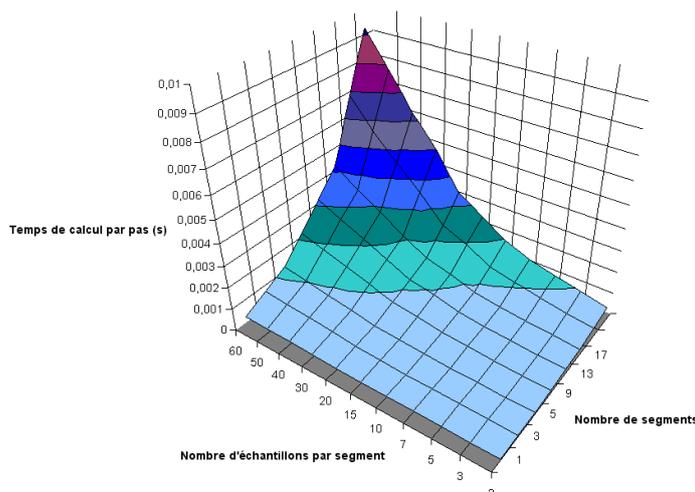
## C RÉCAPITULATIF ET RÉSULTATS

Cette section rassemble et ordonne toutes les étapes de la simulation de splines matérielles, en vue de faciliter leur implémentation. Elle offre également quelques résultats pratiques concernant les performances en fonction des segments et des contraintes qui composent la spline matérielle.

### 1 Vue d'ensemble

Nous donnons ici une vue d'ensemble de la simulation mécanique d'une spline matérielle. Tous les étapes ont déjà été traités dans les chapitres précédents. Nous indiquons les sections ou les algorithmes s'y référant. Leur ordre est important : dans de nombreux cas, les calculs découlent des étapes précédentes.

1. *Détection de la collision* : spline matérielle avec l'environnement, avec elle même et les sphères des clips (section 1, page 107, (Leno4))
2. *Gestions des clips* (section a, page 105)
  - Création des clips si nécessaire (algorithme 4)
  - Vérification des clips et suppression si nécessaire (algorithme 5)
3. *Mise à jour de la structure du système*, si au moins une contrainte évolue (ajout, suppression, décalage)
  - Placement des points de contrôle et des contraintes dans le système (algorithme 9)
  - Calcul des dimensions de la bande du système (algorithme 10)
4. *Calcul des échantillons du système et autres précalculs* (équations II.84 à II.90)
5. *Calcul des repères locaux* (section d, page 67)
6. *Bilan des forces et moments généralisés*
  - Calcul des vitesses de l'état précédent (selon résolution dynamique ou statique, voir section 2, page 128)
  - Calcul des forces et des champs de forces externes (section 1, page 92)
    - Ajout de la gravité
    - Ajout des forces externes ponctuelles (répartition des objets forces)
    - Ajout de la réponse à la collision (si méthode à pénalité)
  - Calcul de la plasticité et traitement de la rupture (algorithme 3)
  - Calcul de  $\mathbb{K}$  (section b, page 78) et  $\mathbf{P}$  (section 3, page 73), par segment (section 1, page 118)
    - Calcul des éléments en traction-compression
    - Calcul des éléments en flexion
    - Calcul des éléments en torsion
    - Calcul des éléments d'inertie, si résolution dynamique
    - Assemblage des matrices élémentaires
  - Calcul de la partie symétrique de la matrice du système
7. *Ajout des éléments de contraintes dans le système* (section 2, page 93)
8. *Ajout de la réponse à la collision* (si méthode à contraintes)
9. *Résolution bande LU du système* (bibliothèque Intel MKL) (section 3, page 122)
10. *Mise à jour des degrés de liberté du système*
  - Mise à jour des coordonnées des points de contrôles
  - Mise à jour des contraintes : récupération du multiplicateur et calcul de l'abscisse (section b, page 122, algorithme 11)
  - Purge des contraintes : suppression des contraintes d'abscisse variable qui sont sorties de la spline matérielle



**Fig. IV.9** – Temps de calcul d'un pas de simulation en fonction de l'échantillonnage par segment et du nombre de segments.

## 2 Performances et limitations

Nous avons testé notre implémentation de splines matérielles sur plusieurs types de machines. Les processeurs Pentium M se révèlent légèrement plus efficaces que les Pentium IV, sans doute à cause de la plus grande quantité de cache de niveau 1 présente. Les mesures qui suivent ont été effectuées sur un ordinateur équipé de deux processeurs Intel Core Duo 7200TM, cadencés à 2Ghz, et doté de 1Go de RAM. Pour obtenir des résultats fiables, nous avons évalué les temps en moyennant des séries de 10 mesures, l'écart-type nous donnant la précision de la mesure effectuée.

### a Déformations

Nous avons d'abord mesuré l'impact de l'échantillonnage par segment et du nombre de segments de spline sur le temps de calcul d'un pas (figure IV.9). Le minimum d'échantillons par segment est de 2. Le nombre maximal d'échantillons sur toute la spline a été limité à 2000 par la quantité de mémoire vive. Au delà, les accès au disque dur altèrent de manière significative les performances. Le graphe montre qu'il est un peu moins coûteux d'augmenter l'échantillonnage par segment que ne nombre de points de contrôle de la spline. Néanmoins, rappelons que la densité de point de contrôle conditionne la précision de la déformation obtenue. De plus, un nombre plus élevé de degrés de liberté autorise le système à être contraint plus finement.

Nous avons ensuite vérifié la linéarité de l'algorithme et évalué le temps de calcul d'un pas de simulation, en rajoutant successivement les différents types de déformation (figures IV.10, IV.11 et IV.12, et tableau IV.1). Nous avons opté pour 5 échantillons par segment. Le modèle de spline matérielle a en effet l'avantage intrinsèque de pouvoir considérer seulement les déformations désirées.

Le premier constat est que le coût de simulation est bien de complexité linéaire, en fonction du nombre total d'échantillons, de segments ou de points de contrôle, ces quantités étant linéairement dépendantes. Nous pouvons également identifier le point faible de notre algorithme. Comme l'on pouvait s'y attendre, il s'agit du calcul des déformations. Il occupe près de la moitié du temps pour la traction-compression seule, près des trois-quarts si l'on rajoute la flexion, et 90% avec en plus la torsion. Les efforts d'optimisations supplémentaires devraient donc se concentrer sur cette partie, bien qu'un travail considérable ait été déjà fourni à ce niveau. Enfin, les résolutions quasi-statiques et dynamiques ont des coûts très proches, ce qui facilite le fonctionnement de notre algorithme de simulation quasi-dynamique.

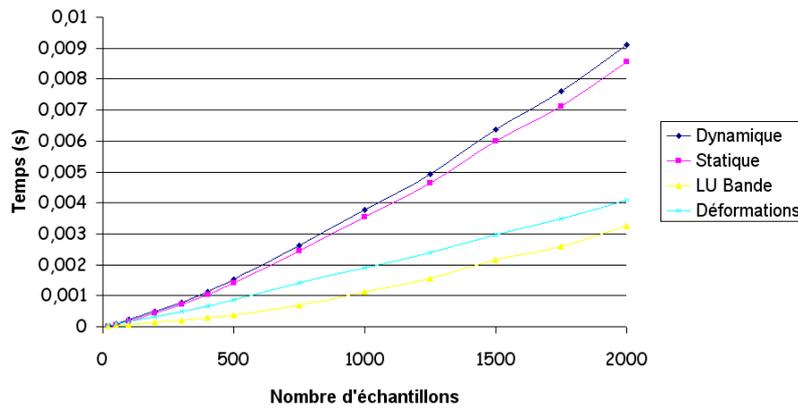


Fig. IV.10 – Temps de calcul d'un pas en fonction du nombre total d'échantillons pour la déformation de traction-compression.

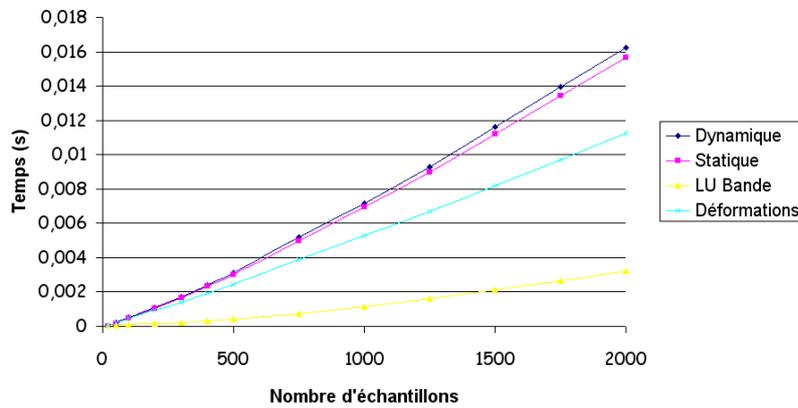


Fig. IV.11 – Temps de calcul d'un pas en fonction du nombre total d'échantillons pour les déformations de traction-compression et de flexion.

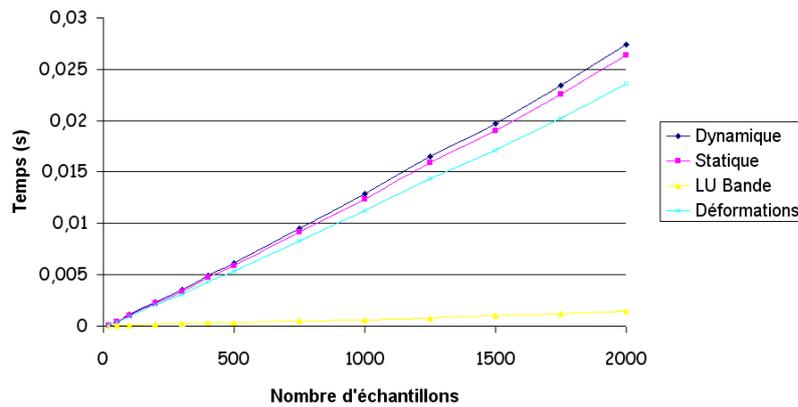


Fig. IV.12 – Temps de calcul d'un pas en fonction du nombre total d'échantillons pour les déformations de traction-compression, de flexion et de torsion.

Pts de contrôle	4	10	20	40	60	80	100	150	200	300	400
Echantillons	20	50	100	200	300	400	500	750	1000	1500	2000
LU Bande	2,89E-05	5,25E-05	0,000111	0,000156	0,000220	0,000313	0,000326	0,000487	0,000643	0,001044	0,001436
Déf. TC	5,79E-05	9,91E-05	0,000169	0,000312	0,000477	0,000661	0,000866	0,00141	0,00190	0,00297	0,00409
Statique	7,54E-07	6,88E-05	0,000183	0,000434	0,000718	0,00103	0,00142	0,00244	0,00355	0,00599	0,00854
Dynamique	1,86E-05	9,33E-05	0,000217	0,000484	0,000786	0,00112	0,00152	0,00261	0,00378	0,00637	0,00910
Déf. TC,F	7,325E-05	0,000210	0,000441	0,000917	0,00140	0,00192	0,00245	0,00391	0,00527	0,00819	0,0112
Statique	4,52E-06	0,000169	0,000465	0,00104	0,00166	0,00231	0,00302	0,00499	0,00692	0,0111	0,0156
Dynamique	2,22E-05	0,000193	0,000497	0,00109	0,00172	0,00239	0,00312	0,00516	0,00717	0,0116	0,0162
Déf. TC,F,T	0,000103	0,000432	0,000989	0,00206	0,00313	0,00436	0,00538	0,00830	0,0111	0,0171	0,0235
Statique	4,91E-05	0,000422	0,00105	0,00221	0,00340	0,00479	0,00590	0,00915	0,0123	0,0190	0,0263
Dynamique	6,93E-05	0,000453	0,00110	0,00229	0,00352	0,00495	0,00611	0,00948	0,0128	0,0197	0,0273

Tab. IV.1 – Mesures de performances, en secondes, pour une spline libre à 5 échantillons par segment, en traction-compression seule, puis avec la flexion et enfin la torsion.

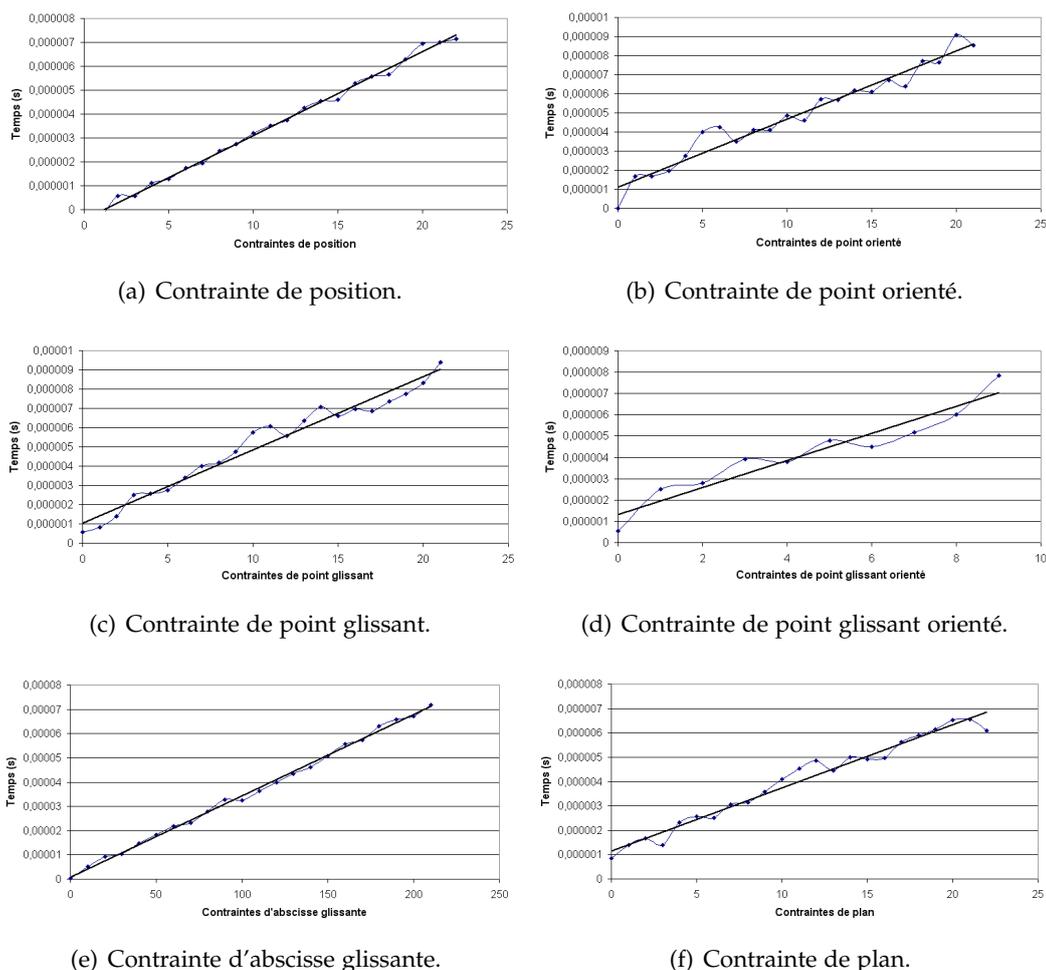


Fig. IV.13 – Temps d'ajout de diverses contraintes, en secondes.

## b Contraintes

L'évaluation des performances des contraintes n'est pas aussi simple qu'il y paraît. Premièrement, rajouter des contraintes influe sur la taille de la bande de la matrice. Selon que les contraintes sont également réparties tout le long de la spline ou concentrées sur un même segment, les temps de calcul peuvent varier d'une complexité linéaire à cubique par rapport au nombre d'inconnues du système. En effet, en cas de concentration de toutes les contraintes sur un même segment, la bande tend à occuper tout l'espace possible. Le coût de la décomposition et de la résolution LU est alors le même qu'une matrice pleine, c'est-à-dire cubique par rapport au nombre d'inconnues.

Il est par conséquent nécessaire d'isoler le coût de l'ajout de la contrainte au système, de celui de la résolution. Nous avons évalué les contraintes suivantes : position (figure IV.13(a)), position orientée (figure IV.13(b)), point glissant (figure IV.13(c)), point glissant orienté (figure IV.13(d)), abscisse glissante (figure IV.13(e)) et plan (figure IV.13(f)). Les courbes obtenues ne sont pas parfaitement droites, de par la faible amplitude des temps mesurés. Une régression linéaire est donc effectuée pour avoir une meilleure idée de la tendance générale. Les temps de calculs des contraintes sont tous linéaires. Le coût élémentaire de chaque contrainte est en dessous de la micro-seconde. La contrainte de plan se révèle donc être propice à une méthode de réponse à la collision précise, sans pour autant affecter les performances. Notons également que les contraintes d'abscisse glissante sont extrêmement peu coûteuses, d'autant plus qu'elles n'augmentent pas la taille de la bande de la matrice, où qu'elles soient situées.

Modélisation	Discretisation			Contraintes	Performance	
	Pts de contrôle	Segments	Echant./seg.	Multiplicateurs	Pas de temps	Temps de calcul
Câble simple sur portière	18	15	10	0-40	3ms	720us-1ms
Faisceau sur structure latérale	40	22	6	42-87	3ms	1-1,2ms
ADN	34	28	4	38	2ms	1,2ms
Pont	32	20	5	49	4ms	2ms
Collier	16	13	5	8-100	0,8ms	0,4-0,8ms
Particules sur ressort	31	28	4	8	1ms	0,7ms

**Tab. IV.2** – Récapitulatif des différentes expérimentations en termes de discrétisation, contraintes et performance.

### c Récapitulatifs des diverses expérimentations

Les performances dans un contexte de manipulation sont généralement plus parlantes. Nous récapitulons les performances obtenues dans diverses expérimentations effectuées dans le cadre de cette thèse (tableau IV.2). Force est de constater que nous avons rempli notre objectif de performance.

## BILAN

Dans ce chapitre, nous avons traité le dernier aspect de notre modèle de spline matérielle, à savoir la performance. Nous avons vu que le placement des points de contrôle et des contraintes dans le système joue un rôle prépondérant dans l'efficacité de la résolution du système matriciel. Nous avons donc proposé des pistes et des algorithmes permettant d'aboutir à une structure bande performante du problème. Nous avons ensuite proposé de combiner équilibre quasi-statique et équilibre dynamique pour tirer parti du meilleur des deux, en commutant automatiquement entre les deux en cours de manipulation. Le passage s'effectue à l'aide d'heuristiques portant sur la nature du matériau, les contraintes appliquées sur la spline matérielle et le temps de calcul. Nous avons enfin montré par des mesures la complexité linéaire de notre algorithme d'objet 1D déformable contraint. Enfin, nous avons déterminé le goulot d'étranglement de notre méthode. Comme l'on pouvait s'y attendre, il s'agit du calcul des raideurs, forces et moments généralisés. Bien que les expressions analytiques utilisées aient été déjà travaillées longuement, aucune tentative d'approximation n'a été faite. Nous ne souhaitons évidemment pas retomber sur les formulations en petits déplacements, mais la marge de manoeuvre avec les grands déplacements est suffisamment large pour espérer trouver un compromis qui soit plus performant, tout en gardant une précision acceptable.



# Conclusion

**C**E MÉMOIRE présente les travaux que nous avons menés à bien pour aboutir à la simulation générale d'objets déformables 1D, aussi bien utilisable en robotique, biologie ou animation. Nous récapitulons d'abord toutes les contributions que nous avons apportées dans cette thèse, puis nous dressons une liste des études en cours ou à venir.

## A RÉSUMÉ DES CONTRIBUTIONS

Pour produire un modèle adapté à la réalité virtuelle, nous avons allié précision, pouvoir d'interaction et performance sur le modèle de spline matérielle dynamique, ce qui à notre connaissance n'avait pas été réalisé auparavant.

### 1 Précision

Nos premiers efforts ont porté sur la précision et la validité des déformations de ce modèle, en émettant l'hypothèse des grands déplacements. Nous avons proposé une formulation de spline mécanique, incluant la flexion et de la torsion. Cette dernière est obtenue en la considérant comme la somme d'une composante géométrique, la torsion de Frenet, et d'une composante matérielle, le roulis, modélisé par un degré de liberté supplémentaire par point de contrôle. Nous avons non seulement tenu compte des déformations réversibles, dites élastiques, mais nous avons aussi proposé un algorithme gérant les déformations irréversibles, dites plastiques, ainsi que de la détection de rupture. Nous avons également indiqué comment prendre en compte des sections d'objets 1D non plus seulement circulaires, mais comportant deux axes de symétrie orthogonaux dans le plan de la section, qu'elles soient pleines ou creuses, ou composées d'une ou plusieurs couches de matériaux distincts. Leurs dimensions peuvent aussi varier le long de l'élément.

Pour résoudre l'équilibre mécanique, nous avons utilisé les équations de Lagrange. Nous avons appliqué un schéma numérique d'intégration Euler-Implicite linéarisé au modèle spline, pour apporter une simulation stable. La résolution se faisant par rapport aux déplacements, une solution aussi bien dynamique que quasi-statique est possible. Nous avons ensuite exprimé les termes des forces, moments et raideurs généralisés. Un effort particulier a été fourni pour les factoriser au mieux et ainsi diminuer fortement le nombre d'opérations nécessaires à leur calcul. N'ayant fait aucune approximation au niveau de la géométrie, nous pouvons parler de modèle de *spline géométriquement exacte*.

Enfin, nous avons proposé une méthode pour évaluer la précision de l'interpolation des splines et des éléments finis. Elle est fondée sur l'erreur d'approximation d'un cercle avec ces fonctions. Nous avons montré que la B-Spline est la spline qui offre la meilleure précision. De plus, nous avons pu établir la densité de points de contrôle nécessaire à une spline pour une précision donnée.

Une partie des résultats sur la précision a fait l'objet d'une publication ([TGAB07](#)).

## 2 Interaction

Nos travaux ont ensuite contribué à améliorer la manipulation des splines matérielles mécaniques .

Nous avons d'abord reformulé les contraintes lagrangiennes pour qu'ils soient utilisables avec un schéma d'Euler-Implicite linéarisé. Jusqu'à présent exprimées par rapport à des accélérations, elles l'ont été cette fois par rapport à des déplacements. Cela nous a permis d'obtenir des résolutions plus simples et directes, puisque le schéma d'intégration temporelle de Baumgarte n'est plus nécessaire. Nous avons ainsi redéfini les contraintes de position, d'orientation, qui donnent à la spline matérielle une configuration désirée ; les contraintes d'axe et de plan, qui restreignent un ou plusieurs points de la spline à ces sous-espaces ; et la contrainte de point glissant, qui impose à la spline un point de passage, celle-ci étant libre de circuler au travers. Nous proposons également de nouvelles contraintes : la contrainte de distance, qui maintient constante la distance entre un point de la spline et un point de l'espace, ou un autre point de la spline ; la contrainte de point glissant orienté, qui par rapport au point glissant, fixe en plus la direction de la spline à son niveau ; enfin la contrainte d'abscisse glissante, qui modélise le comportement d'une particule, dont la trajectoire est liée à celle de la spline, mais qui peut aussi l'affecter.

Plusieurs applications découlent de l'utilisation de ces nouvelles contraintes : les jonctions entre plusieurs splines, qui rendent possible la simulation d'arborescences telles que les faisceaux de câbles ; la gestion de structures complexes comme un brin d'ADN ou un pont ; le couplage d'une spline matérielle avec un autre modèle mécanique ; pour terminer, la modélisation de clips de fixation, pour rattacher la spline à une structure.

Une étude sur la détection de collision achève ce chapitre. Nous avons établi le coût et la complexité pour différents niveaux de représentation : sphères, cylindres droits, cylindres généralisés, et même la spline elle-même. Nous avons conclu que la primitive de type cylindre reste la meilleure solution adaptée au temps-réel.

## 3 Performance

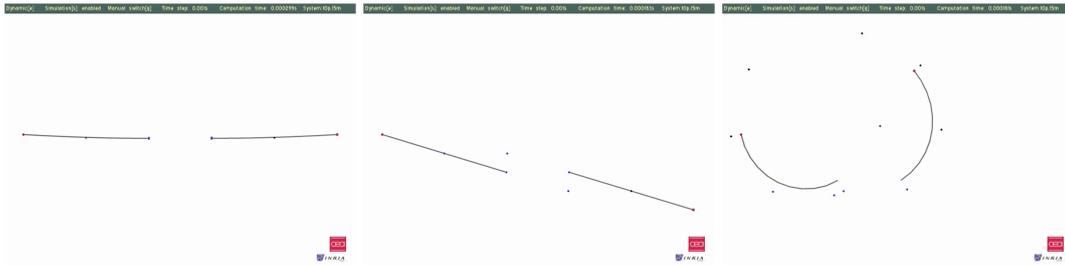
Notre dernier objectif a été la performance du modèle de spline matérielle, qui a fait l'objet d'une autre publication (TGDMo7). Nous avons d'abord proposé une méthode de placement optimisé des inconnues du système, à savoir les points de contrôle et les contraintes du système, pour garantir au système une structure bande efficace. Nous avons également exploité toutes les symétries présentes.

Nous avons enfin proposé la simulation quasi-dynamique, qui concilie les avantages d'une simulation dynamique et quasi-statique. Elle consiste à alterner automatiquement ces deux équilibres au cours de la manipulation. Le choix est fonction de quelques heuristiques que nous avons définies. Elles dépendent du matériau de l'objet simulé, des contraintes appliquées sur la spline et du temps de calcul alloué.

Nous invitons maintenant le lecteur à consulter à nouveau le tableau I.1. Notre modèle de spline géométriquement exacte remplit tous les objectifs : les déformations en traction-compression, torsion et flexion sont gérées et validées ; le modèle n'impose pas de contrainte pour être défini, et gère les contraintes dures ; les performances sont très bonnes, de l'ordre de la centaine de micro-secondes par pas de simulation pour un segment de spline, l'algorithme global étant de complexité linéaire.

## B PERSPECTIVES

De nombreuses perspectives découlent de nos travaux. Elles concernent d'une part des points qui n'ont pu être abordés, et d'autre part des applications directes à fort potentiel.



**Fig. V.1** – Mise en évidence de l’insuffisance de l’utilisation conjointe de contraintes de position et d’orientation relatives pour modéliser un encastrement relatif.

### 1 Contrainte de déplacement relatif

Nous avons proposé un nombre important de contraintes, mais une qui fait encore défaut est la contrainte de déplacement relatif, c’est-à-dire une contrainte qui lie deux points de spline comme si elles étaient encastées sur un même objet rigide. Son utilisation permettrait d’envisager la simulation multi-résolution d’une arborescence, selon l’aptitude des éléments de l’objet à se déformer. Un élément rigide ou quasiment rigide serait remplacé par une telle contrainte. Le gain de simulation serait conséquent, puisque comme nous l’avons vu, c’est le calcul des déformations qui monopolise le processeur. Nous avons bien des contraintes de position et d’orientation relatives, mais leur utilisation conjointe ne suffit pas (figure V.1). En effet, le changement d’orientation des extrémités influe sur la translation. Le cas particulier où les deux vecteurs tangents sont coplanaires mais non parallèles possède une solution sous la forme d’un encastrement coplanaire relatif (section c), mais elle reste limitée.

### 2 Collision

Un autre point qui fait défaut est le traitement de la collision. Nous n’avons guère abordé la réponse à la collision, mais l’utilisation unilatérale de contraintes de plan ou de distance peut se révéler précise et performante. En effet, elles sont peu coûteuses et précises d’une part, et fortement malléables d’une autre. La contrainte de distance symbolise une boule dans laquelle un point donné de la spline ne peut pénétrer, et l’on peut translater son centre d’un état à un autre. La contrainte de plan peut changer d’orientation et subir une translation dans l’espace. Ce plan peut donc être mis à jour à chaque état, en fonction de la position et de l’orientation d’un point de contact.

### 3 Répartition adaptative des points de contrôle

Dans ce mémoire, nous n’avons pas traité l’aspect multi-résolution des splines, qui permet d’adapter localement la précision des déformations. Des travaux précédents de Lenoir *et al.* font cependant état d’une technique permettant de raffiner et de diminuer le nombre de points de contrôle de la spline de manière géométriquement exacte. Le fait d’ajouter dynamiquement des degrés de liberté au système nécessite d’une part de le réorganiser, et d’autre part augmente son coût de résolution. Le problème réside surtout dans la suppression : il n’existe pas de méthode exacte pour une configuration non rectiligne. Dans ce cas, la géométrie est altérée et introduit des énergies supplémentaires, sources d’instabilité. Nous pensons qu’au lieu de rajouter des degrés de liberté au système, il serait plus judicieux de les répartir dynamiquement le long de la spline. Même si elle ne se faisait pas de manière exacte, une variation progressive devrait assurer un comportement suffisamment robuste. La densité de courbure nous semble être un premier critère pertinent à explorer.

#### 4 Simulation de l'ADN

Nous avons effectué une première tentative de simulation de l'ADN. Nous tenons compte de la structure en double hélice, avec la présence d'un petit et d'un grand sillon, ainsi que de la répartition correcte des liaisons entre les deux hélices. Les premiers résultats sont encourageants, de par leur rapidité. Mais de nombreux phénomènes ayant un impact sur l'ADN ne sont pas encore gérés : la rupture des liaisons, qui dépend de la nature des bases azotées (Adémine-Guanine ou Cytosine-Thymine) ; le milieu visqueux dans lequel évoluent les brins d'ADN ; les forces électro-magnétiques, etc. Les enjeux de la compréhension du mécanisme de réplication et d'expression du code génétique sont tels que cette piste de recherche gagnerait beaucoup à être approfondie.

#### 5 Simulation mécanique d'objets 3D à partir d'un squelette spline

Une autre perspective concerne la simulation mécanique d'objets 3D à partir d'un squelette spline. Nous souhaiterions animer des objets relativement filiformes, comme par exemple des arbres. Il existe bien une méthode d'animation à partir de splines (FOKGM07), mais elle n'est absolument pas mécanique, c'est à l'utilisateur de spécifier les déplacements de points de contrôle. Nous savons certes simuler une arborescence de splines matérielles, chaque segment pouvant comporter une section aux dimensions variables, mais les logiciels de conception assistée par ordinateur ne proposent évidemment pas un tel format de sortie pour les objets modélisés. Construire un squelette spline à la main est extrêmement fastidieux pour des structures câblées d'environnement de production (figure V.2(a)). L'idée est de partir du maillage de l'objet, chose que l'on peut obtenir facilement. Il existe de nombreuses techniques d'extraction de squelettes à partir d'objets 3D (CM07; ACK01), certaines paraissant plus adaptées que d'autres à des éléments 1D. Nous avons d'ores et déjà adapté une méthode pour obtenir un squelette et l'arborescence de splines avec les contraintes de jonctions (figures V.2(b) et V.2(c)). Celle-ci doit cependant encore être améliorée pour être simulée. La forme de la section en chaque point de contrôle de la spline doit également être déterminée en fonction du maillage environnant. Nous espérons être prochainement capables de simuler dynamiquement un arbre en temps-réel, et de manière réaliste.

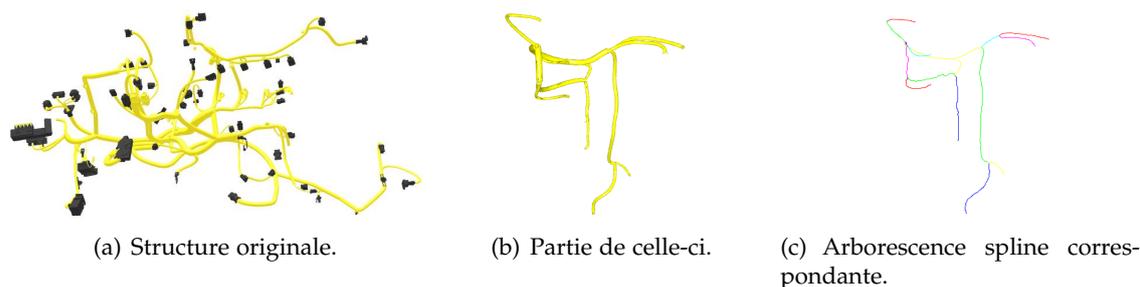


Fig. V.2 – Structure de faisceau de câbles à fixer sur un moteur, ©Renault

# Bibliographie

- [ACK01] N. Amenta, S. Choi, et R. Kolluri.. « The power crust ». *Proceedings of the sixth ACM Symposium on Solid Modeling and Applications*, pages 249–260, 2001. (Cité page 140.)
- [AGP<sup>+</sup>06] Marco Agus, Enrico Gobbetti, Giovanni Pintore, Gianluigi Zanetti, et Antonio Zorcolo. « Real-time Cataract Surgery Simulation for Training ». *Eurographics Italian Chapter Conference*, Conference held in Catania, Italy, 2006. Eurographics Association. (Cité page 175.)
- [AHS03] Irene Albrecht, Jörg Haber, et Hans-Peter Seidel. « Construction and animation of anatomically based human hand models ». *SCA '03 : Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 98–109, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association. (Cité page 173.)
- [AN05a] Basile Audoly et Sébastien Neukirch. « Fragmentation of Rods by Cascading Cracks : Why Spaghetti Does Not Break in Half ». *Physical Review Letters*, 95 :095505, 2005. (Cité page 180.)
- [AN05b] Basile Audoly et Sébastien Neukirch. « Rupture des tiges en flexion ». *Mécanique et Industries*, 6 :365–368, 2005. (Cité page 180.)
- [And95] T. L. Anderson. *Fracture Mechanics : Fundamentals and Applications*. CRC Press, Boca Raton, 1995. (Cité page 17.)
- [Ant72] S. S. Antman. « *Handbuch der Physik VIa/2* », Chapitre The Theory of Rods., pages 641–703. Springer-Verlag, 1972. (Cité pages 22 et 34.)
- [Ant95] Stuart S. Antman. *Nonlinear Problems of Elasticity*. Springer Verlag, 1995. (Cité pages 21 et 88.)
- [AUK92] Ken-ichi Anjyo, Yoshiaki Usami, et Tsuneya Kurihara. « A simple method for extracting the natural beauty of hair ». *SIGGRAPH '92 : Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 111–120, New York, NY, USA, 1992. ACM Press. (Cité pages 34, 35, 51 et 176.)
- [BAC<sup>+</sup>06] Florence Bertails, Basile Audoly, Marie-Paule Cani, Bernard Querleux, Frédéric Leroy, et Jean-Luc Lévêque. « Super-helices for predicting the dynamics of natural hair ». *SIGGRAPH '06 : ACM SIGGRAPH 2006 Papers*, pages 1180–1187, New York, NY, USA, 2006. ACM Press. (Cité pages 34, 36, 43, 53, 54, 88 et 176.)
- [Bal06] Roberto Ballarini. « The Da Vinci-Euler-Bernoulli Beam Theory ? ». *Mechanical Engineering Magazine Online*, 2006. (Cité page 21.)
- [BAQ<sup>+</sup>05] Florence Bertails, Basile Audoly, Bernard Querleux, Frédéric Leroy, Jean-Luc Lévêque, et Marie-Paule Cani. « Predicting Natural Hair Shapes by Solving the Statics of Flexible Rods ». J. Dingliana et F. Ganovelli, editors, *Eurographics (short papers)*. Eurographics, August 2005. (Cité pages 35, 43, 45 et 176.)
- [Bar81] Brian Barsky. « *The beta-spline : a local representation based on shape parameters and fundamental geometric measures* ». Thèse de doctorat, University of Utah, 1981. (Cité page 58.)
- [Bar94] David Baraff. « Fast contact force computation for nonpenetrating rigid bodies ». *SIGGRAPH '94 : Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 23–34, New York, NY, USA, 1994. ACM Press. (Cité page 49.)
- [Bar96] David Baraff. « Linear-time dynamics using Lagrange multipliers ». *SIGGRAPH '96 : Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 137–146, New York, NY, USA, 1996. ACM Press. (Cité pages 32, 50, 51 et 123.)

- [Bar97] Ronen Barzel. « Faking Dynamics of Ropes and Springs ». *IEEE Computer Graphics and Applications*, 17(3) :31–39, 1997. (Cité pages 42 et 178.)
- [Bat82] K. J. Bathe. *Finite Element Procedures in Engineering Analysis*. Prentice-Hall, Englewood Cliffs, 1982. ISBN 0-13-317305-4. (Cité page 14.)
- [Bau72] J. Baumgarte. « Stabilization of constraints and integrals of motion in dynamical systems ». *Comp. Meth. in Appl. Mech. and Eng.*, 1 :1–16, 1972. (Cité pages 39 et 50.)
- [BB83] Brian Barsky et J. Beatty. « Local control of bias and tension in beta-splines ». *Computer Graphics*, 17(3) :193–218, 1983. (Cité page 58.)
- [BBB87] Richard H. Bartels, John C. Beatty, et Brian A. Barsky. *An introduction to splines for use in computer graphics and geometric modeling*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1987. (Cité pages 60 et 61.)
- [BBBo7] Christopher Batty, Florence Bertails, et Robert Bridson. « A fast variational framework for accurate solid-fluid coupling ». *SIGGRAPH '07 : ACM SIGGRAPH 2007 papers*, page 100, New York, NY, USA, 2007. ACM Press. (Cité page 50.)
- [BCNo3] Yosuke Bando, Bing-Yu Chen, et Tomoyuki Nishita. « Animating Hair with Loosely Connected Particles ». *Computer Graphics Forum (Proceedings of Eurographics 2003)*, 22(3) :411–418, 2003. (Cité pages 176 et 177.)
- [Bero6] Florence Bertails. « *Simulation de Chevelures Virtuelles* ». Thèse de doctorat, Institut National Polytechnique de Grenoble, 2006. (Cité pages 14, 20, 30, 49, 176 et 177.)
- [Bin71] T.Q. Binford. « Visual Perception by Computer ». In *Proc. of IEEE Conf. on Systems and Control*, Miami, 1971. (Cité page 70.)
- [Bin87] Thomas O. Binford. « *Encyclopedia of Artificial Intelligence* », Chapitre Generalized cylinder representation, pages 321–323. John Wiley & Sons, 1987. (Cité page 70.)
- [Bis75] Richard L. Bishop. « There is More than One Way to Frame a Curve ». *The American Mathematical Monthly*, 82 :246–251, 1975. (Cité pages 68 et 69.)
- [BKCN03] F. Bertails, T-Y. Kim, M-P. Cani, et U. Neumann. « Adaptive Wisp Tree : a multiresolution control structure for simulating dynamic clustering in hair motion ». *SCA '03 : Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 207–213, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association. (Cité pages 29, 48 et 176.)
- [BLDM99] V. Borderie, L. Laroche, B. Delbosc, et M. Montard. « *Encyclopédie médico-chirurgicale* », Chapitre Kératoplastie transfixiante, pages 21–206–A–10. Elsevier, 1999. (Cité page 175.)
- [Bli87] J.F. Blinn. « How Many Ways Can You Draw a Circle? ». *Computer Graphics and Applications, IEEE*, 7(8) :39–44, 1987. ISSN : 0272-1716. (Cité page 63.)
- [BLMo4] Joel Brown, Jean-Claude Latombe, et Kevin Montgomery. « Real-time knot-tying simulation ». *The Visual Computer : International Journal of Computer Graphics*, 20(2) :165–179, 2004. (Cité pages 33 et 48.)
- [Blo90] Jules Bloomenthal. « *Graphics Gems* », volume 1, Chapitre Calculation of reference frames along a space curve, pages 567–571. Academic Press Professional, Inc., San Diego, CA, USA, 1990. (Cité pages 69 et 70.)
- [Blo02] Jules Bloomenthal. « Medial-based vertex deformation ». *SCA '02 : Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 147–151, New York, NY, USA, 2002. ACM Press. (Cité page 70.)
- [BMLSo1] Joel Brown, Kevin Montgomery, Jean-Claude Latombe, et Michael Stephanides. « A Microsurgery Simulation System ». *MICCAI '01 : Proceedings of the 4th International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 137–144, London, UK, 2001. Springer-Verlag. (Cité pages 33, 48 et 174.)
- [BNAo2] T. Beléndez, C. Neipp, et Beléndez A.. « Large and small deflections of a cantilever beam ». *European Journal of Physics*, 23 :371–379, 2002. (Cité page 84.)
- [Boe80] W. Boehm. « Inserting new knots into B-spline curves. ». *Computer-Aided Design*, 12 :199–201, 1980. (Cité page 60.)

- [BP05] F. Boyer et D. Primault. « Finite element of nonlinear cables : applications to robotics ». *Far East Journal of Applied Mathematics*, 19(1) :1–34, 2005. ISSN 0972-0960. (Cité pages 11, 37, 53, 54, 59, 68, 69 et 89.)
- [Bro70] C. G. Broyden. « The convergence of a class of double-rank minimization algorithms. I : General considerations ». *Journal of the Institute of Mathematics and Its Applications*, 6 :76–90, 1970. (Cité page 44.)
- [BS96] Carole Blanc et Christophe Schlick. « Accurate Parametrization of Conics by NURBS ». *IEEE Computer Graphics and Applications*, 16 :64–71, 1996. (Cité page 63.)
- [BS98] Matthias Buck et Elmar Schömer. « Interactive Rigid Body Manipulation with Obstacle Contacts ». *The Journal of Visualization and Computer Animation*, 9(4) :243–257, – 1998. (Cité page 49.)
- [But03] J. C. Butcher. *Numerical Methods for Ordinary Differential Equations*. 2003. (Cité page 45.)
- [BW90] M. I. G. Bloor et M. J. Wilson. « Representing PDE surfaces in terms of B-splines ». *Comput. Aided Des.*, 22(6) :324–331, 1990. (Cité page 39.)
- [BW92] David Baraff et Andrew Witkin. « Dynamic simulation of non-penetrating flexible bodies ». *SIGGRAPH '92 : Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 303–308, New York, NY, USA, 1992. ACM Press. (Cité page 43.)
- [BW97] D. Baraff et A. Witkin. « Partitioned Dynamics ». Technical Report CMU-RI-TR-97-33, Robotics Institute, Carnegie Mellon University, 1997. (Cité page 50.)
- [BW98] David Baraff et Andrew Witkin. « Large steps in cloth simulation ». *SIGGRAPH '98 : Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 43–54, New York, NY, USA, 1998. ACM Press. (Cité pages 46 et 77.)
- [Béz66a] Pierre Bézier. « Définition numérique de courbes et surfaces, part 1 ». *Automatisme*, 11 :625–632, 1966. (Cité pages 58 et 59.)
- [Béz66b] Pierre Bézier. « Définition numérique de courbes et surfaces, Part 2 ». *Automatisme*, 12 :17–21, 1966. (Cité pages 58 et 59.)
- [Béz77] Pierre Bézier. « Essai de définition numérique des courbes et surfaces expérimentales ». Thèse de doctorat, Université Paris VI, 1977. (Cité pages 58 et 59.)
- [CC09] Eugène Cosserat et François Cosserat. *Théorie des corps déformables*. Hermann, 1909. (Cité pages 21, 22 et 30.)
- [CCK05] Byoungwon Choe, Min Gyu Choi, et Hyeong-Seok Ko. « Simulating complex hair with robust collision handling ». *SCA '05 : Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 153–160, New York, NY, USA, 2005. ACM Press. (Cité pages 33, 49, 53, 176 et 177.)
- [CDA99] Stéphane Cotin, Hervé Delingette, et Nicholas Ayache. « Real-Time Elastic Deformations of Soft Tissues for Surgery Simulation ». *IEEE Transactions on Visualization and Computer Graphics*, 5(1) :62–73, 1999. (Cité page 14.)
- [CDL<sup>+</sup>05] Stéphane Cotin, Christian Duriez, Julien Lenoir, Paul Neumann, et Steven Dawson. « New approaches to catheter navigation for interventional radiology simulation ». *Proceedings of Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pages 534–542, Palm Springs, California, USA, 26-30 october 2005. (Cité pages 38, 49, 53 et 174.)
- [CFL67] R. Courant, K. Friedrichs, et H. Lewy. « On the partial difference equations of mathematical physics ». *IBM Journal*, - :215–234, March 1967. English translation of the 1928 German original. (Cité page 45.)
- [CG88] A. Cardona et M. Géradin. « A beam finite element non-linear theory with finite rotations ». *Int. J. Numer. Meth. Engng.*, 26 :2403–2438, 1988. (Cité page 22.)
- [CGC<sup>+</sup>02] Steve Capell, Seth Green, Brian Curless, Tom Duchamp, et Zoran Popović. « Interactive skeleton-driven dynamic deformations ». *SIGGRAPH '02 : Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 586–593, New York, NY, USA, 2002. ACM Press. (Cité page 15.)

- [Cha98] Henri Chajmowicz. « Un modèle géométriquement exact de jonction de poutres ». Technical Report, INRIA, 1998. (Cité page 168.)
- [Choo03] Nadia Chouaïeb. « Kirchhoff's Problem of Helical Solutions of Uniform Rods and their stability properties ». Thèse de doctorat, Ecole polytechnique fédérale de Lausanne, 2003. (Cité pages 20 et 21.)
- [CJY02] Johnny T. Chang, Jingyi Jin, et Yizhou Yu. « A practical model for hair mutual interactions ». *SCA '02 : Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 73–80, New York, NY, USA, 2002. ACM Press. (Cité pages 32 et 176.)
- [CK05] Byoungwon Choe et Hyeong-Seok Ko. « A Statistical Wisp Model and Pseudophysical Approaches for Interactive Hairstyle Generation ». *IEEE Transactions on Visualization and Computer Graphics*, 11(2) :160–170, 2005. (Cité pages 33 et 176.)
- [CM04] N. Chouaïeb et J. H. Maddocks. « Kirchhoff's Problem of Helical Equilibria of Uniform Rods ». *Journal of Elasticity*, 77 :221–247, 2004. (Cité page 68.)
- [CM07] Nicu D. Cornea et Patrick Min. « Curve-Skeleton Properties, Applications, and Algorithms ». *IEEE Transactions on Visualization and Computer Graphics*, 13(3) :530–548, 2007. Member-Deborah Silver. (Cité page 140.)
- [Cou80a] Jean Courbon. « Structures non linéairement élastiques ». Techniques de l'Ingénieur, traité Construction, 1980. (Cité pages 13 et 168.)
- [Cou80b] Jean Courbon. « Théorie de l'élasticité ». Techniques de l'Ingénieur, traité Construction, 1980. (Cité page 14.)
- [Cou80c] Jean Courbon. « Théorie des poutres ». Techniques de l'Ingénieur, traité Construction, 1980. (Cité pages 12, 13, 20, 23, 27, 65 et 161.)
- [Cox72] M. G. Cox. « The Numerical Evaluation of B-Splines ». *IMA Journal of Applied Mathematics*, 10(2) :134–149, 1972. (Cité page 60.)
- [CPNNR06] S. Chalençon-Piotin, O. Nocent, J.-M. Nourrit, et Y. Remion. « A Virtual Prototyping Tool for Knitwear Simulation ». *Industrial Simulation Conference 2006*, University of Palermo, Palermo, Italy, June 2006. (Cité page 169.)
- [CR74] E. Catmull et R. Rom. « A class of local interpolating splines ». *Computer Aided Geometric Design*, pages 317–326, 1974. (Cité pages 58 et 59.)
- [CS06] P. Coleman et Karan Singh. « Cords : Geometric Curve Primitives for Modeling Contact ». *IEEE Computer Graphics and Applications*, 26 :72–79, 2006. (Cité page 178.)
- [CW92] George Celniker et Will Welch. « Linear constraints for deformable non-uniform B-spline surfaces ». *SI3D '92 : Proceedings of the 1992 symposium on Interactive 3D graphics*, pages 165–170, New York, NY, USA, 1992. ACM Press. (Cité page 39.)
- [CZ92] David T. Chen et David Zeltzer. « Pump it up : computer animation of a biomechanically based model of muscle using the finite element method ». *SIGGRAPH '92 : Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 89–98, New York, NY, USA, 1992. ACM Press. (Cité pages 14, 172 et 173.)
- [Dai99] Fan Dai. *Lebendige virtuelle Welten*. Springer Verlag, 1999. (Cité page 33.)
- [dB72] Carl de Boor. « On calculating with B-splines ». *J. Approx. Theory*, 6 :50–62, 1972. (Cité page 60.)
- [DDCB01] Gilles Debunne, Mathieu Desbrun, Marie-Paule Cani, et Alan H. Barr. « Dynamic real-time deformations using space & time adaptive sampling ». *SIGGRAPH '01 : Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 31–36, New York, NY, USA, 2001. ACM Press. (Cité page 15.)
- [Deq05] Jérémie Dequidt. « Objets Autonomes en Simulation Physique Temps-réel ». Thèse de doctorat, Université de Lille 1, 2005. (Cité page 47.)
- [DSB99] Mathieu Desbrun, Peter Schröder, et Alan Barr. « Interactive animation of structured deformable objects ». *Proceedings of Graphics interface*, pages 1–8, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc. (Cité pages 15, 45, 46 et 77.)

- [DTKT93] A. Daldegan, N. M. Thalmann, T. Kurihara, et D. Thalmann. « An Integrated System for Modeling, Animating and Rendering Hair ». *Computer Graphics Forum (Eurographics '93)*, 12(3) :211–221, 1993. (Cité pages 34, 35 et 48.)
- [EH95] Matthias Eck et Jan Hadenfeld. « Knot removal for B-spline curves ». *Computer Aided Geometric Design Volume 12, Issue 3, May 1995, Pages 259–282*, 12(3) :259–282, May 1995. (Cité page 60.)
- [Els61] L. E. Elsgolc. *Calculus of Variations*. Pergamon Press, 1961. (Cité pages 37 et 45.)
- [Far88] Gerald Farin. *Curves and surfaces for computer aided geometric design : a practical guide*. Academic Press Professional, Inc., San Diego, CA, USA, 1988. (Cité pages 57 et 60.)
- [Fau99] François Faure. « Interactive Solid Animation Using Linearized Displacement Constraints ». B. Arnaldi et G. Hégron, editors, *Computer Animation and Simulation'98*, pages 61–72, 1999. (Cité page 51.)
- [FC80] F. N. Fritsch et R. E. Carlson. « Monotone Piecewise Cubic Interpolation ». *SIAM Journal on Numerical Analysis*, 17(2) :238–246, 1980. (Cité page 59.)
- [Fea87] Roy Featherstone. *Robot Dynamics Algorithms*. Kluwer Academic Publishers, 1987. (Cité page 32.)
- [FLA<sup>+</sup>05] Laure France, Julien Lenoir, Alexis Angelidis, Philippe Meseure, Marie-Paule Cani, François Faure, et Christophe Chaillou. « A layered model of a virtual human intestine for surgery simulation ». *Medical Images Analysis*, 9(2) :123–132, April 2005. (Cité pages 70 et 173.)
- [FOKGM07] Sven Forstmann, Jun Ohya, Artus Krohn-Grimberghe, et Ryan McDougall. « Deformation Styles for Spline-based Skeletal Animation ». *SCA '07 : Proceedings of the 2007 ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, pages 141–150, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association. (Cité page 140.)
- [FP63] R. Fletcher et M. J. D. Powell. « A rapidly convergent descent method for minimization ». *The Computer Journal*, 6(2) :163–168, 1963. (Cité page 45.)
- [FPF99] C. A. Felippa, K. C. Park, et C. Farhat. « Partitioned Analysis of Coupled Mechanical Systems ». Technical Report, COLLEGE OF ENGINEERING UNIVERSITY OF COLORADO, 1999. (Cité pages 50 et 51.)
- [FvDFH89] J. Foley, A. van Dam, S. Feiner, et J. Hughes. *Computer Graphics : Principles and Practice*. Addison-Wesley, 1989. (Cité page 58.)
- [Gar99] Jean Garrigues. « Statiques des poutres élastiques », 1999. (Cité pages 20, 23, 24, 27 et 161.)
- [Gar07] Jean Garrigues. *Fondements de la mécanique des milieux continus*. Editions Hermes, 2007. ISBN : 978-2-7462-1607-5. (Cité pages 11, 12, 14 et 36.)
- [GCF01] Thomas Di Giacomo, Stéphane Capo, et François Faure. « An interactive forest ». *Proceedings of the Eurographic workshop on Computer animation and simulation*, pages 65–74, New York, NY, USA, 2001. Springer-Verlag New York, Inc. (Cité page 43.)
- [GGHM05] Nico Galoppo, Naga K. Govindaraju, Michael Henson, et Dinesh Manocha. « LU-GPU : Efficient Algorithms for Solving Dense Linear Systems on Graphics Hardware ». *SC '05 : Proceedings of the 2005 ACM/IEEE conference on Supercomputing*, page 3, Washington, DC, USA, 2005. IEEE Computer Society. (Cité page 37.)
- [GHBV05] J. R. Gladden, N. Z. Handzy, A. Belmonte, et E. Villermaux. « Dynamic Buckling and Fragmentation in Brittle Rods ». *Physical Review Letters*, 94 :035503, 2005. (Cité page 88.)
- [GHK<sup>+</sup>06] Gernot Goebbels, Nils Hornung, Uwe Klein, Stefan Müller, Igor Nikitin, et Christian Wiens. « FleXengine : Highly accurate real-time simulation system for cables, hoses and wiring harnesses with contacts ». *Proceedings of the International Wire and Cable Symposium*, 2006. (Cité page 169.)
- [GKS02] Eitan Grinspun, Petr Krysl, et Peter Schröder. « CHARMS : a simple framework for adaptive simulation ». *SIGGRAPH '02 : Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 281–290, New York, NY, USA, 2002. ACM Press. (Cité page 15.)
- [GLM96] S. Gottschalk, M. C. Lin, et D. Manocha. « OBTree : a hierarchical structure for rapid interference detection ». *SIGGRAPH '96 : Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 171–180, New York, NY, USA, 1996. ACM Press. (Cité page 48.)

- [GLMo6] Russell Gayle, Ming C. Lin, et Dinesh Manocha. « Adaptive Dynamics with Efficient Contact Handling for Articulated Robots ». *Robotics : Science and Systems*, 2006. (Cité pages 32, 48, 49 et 169.)
- [GM97] S. F. Gibson et B. Mirtich. « A Survey of Deformable Models in Computer Graphics ». Technical Report TR-97-19, Mitsubishi Electric Research Laboratories, Cambridge, November 1997. (Cité page 15.)
- [GM03] Laurent Grisoni et Damien Marchal. « High Performance generalized cylinders visualization ». *SMI '03 : Proceedings of the Shape Modeling International 2003*, page 257, Washington, DC, USA, 2003. IEEE Computer Society. (Cité pages 70 et 108.)
- [GN06] Alain Goriely et Sébastien Neukirch. « The mechanics of climbing and attachment in twining plants ». *Physical Review Letters*, 97 :184302, 2006. (Cité page 180.)
- [GNRZ02] Leonidas Guibas, An Nguyen, Daniel Russel, et Li Zhang. « Collision detection for deforming necklaces ». *SCG '02 : Proceedings of the eighteenth annual symposium on Computational geometry*, pages 33–42, New York, NY, USA, 2002. ACM Press. (Cité page 48.)
- [Gou77] J. L. Gout. « Estimation de l'erreur d'interpolation d'Hermite dans  $R^n$  ». *Numerische Mathematik*, 28(4) :407–429, December 1977. ISSN0029-599X (Print) 0945-3245 (Online). (Cité page 65.)
- [Gri99] Laurent Grisoni. « *Éléments de multirésolution en modélisation géométrique* ». Thèse de doctorat, Université de Bordeaux I, 1999. (Cité pages 56 et 58.)
- [GRS95] Eduard Gröller, René T. Rau, et Wolfgang Straßer. « Modeling and Visualization of Knitwear ». *IEEE Transactions on Visualization and Computer Graphics*, 1(4) :302–310, 1995. (Cité page 169.)
- [GS06] Mireille Grégoire et Elmar Schömer. « Interactive simulation of one-dimensional flexible parts ». *SPM '06 : Proceedings of the 2006 ACM symposium on Solid and physical modeling*, pages 95–103, New York, NY, USA, 2006. ACM Press. (Cité pages 30, 31, 34, 38, 45, 49, 50, 54, 68, 85, 89, 169 et 170.)
- [GS07] Mireille Grégoire et Elmar Schömer. « Interactive simulation of one-dimensional flexible parts ». *Comput. Aided Des.*, 39(8) :694–707, 2007. (Cité pages 30, 45, 53, 54 et 87.)
- [GSLMo5] Russell Gayle, William Segars, Ming C. Lin, et Dinesh Manocha. « Path Planning for Deformable Robots in Complex Environments ». *Proceedings of Robotics : Systems and Science*, 2005. (Cité page 174.)
- [GSR<sup>+</sup>07] Russell Gayle, Avneesh Sud, Stephane Redon, Ming C. Lin, et Dinesh Manocha. « Efficient Motion Planning of Highly Articulated Chains Using Physics-based Sampling ». *Proc. of IEEE International Conference on Robotics and Automation*, 2007. (Cité pages 32, 53, 169 et 170.)
- [GTT89] J.-P. Gourret, N. M. Thalmann, et D. Thalmann. « Simulation of object and human skin formations in a grasping task ». *SIGGRAPH '89 : Proceedings of the 16th annual conference on Computer graphics and interactive techniques*, pages 21–30, New York, NY, USA, 1989. ACM Press. (Cité page 14.)
- [GvdHTNo5] V. G. A. Goss, G. H. M. van der Heijden, J. M. T. Thompson, et S. Neukirch. « Experiments on Snap Buckling, Hysteresis and Loop Formation in Twisted Rods ». *Experimental Mechanics*, 45 :101–111, 2005. (Cité pages 88 et 89.)
- [GZ02] Yang Guang et Huang Zhiyong. « A Method of Human Short Hair Modeling and Real Time Animation ». *PG '02 : Proceedings of the 10th Pacific Conference on Computer Graphics and Applications*, page 435, Washington, DC, USA, 2002. IEEE Computer Society. (Cité page 176.)
- [Hado6] Sunil Hadap. « Oriented strands : dynamics of stiff multi-body system ». *SCA '06 : Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 91–100, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association. (Cité pages 32, 49, 53, 85 et 179.)
- [Han94] A. J. Hanson. « Quaternion Frenet Frames ». Technical Report, Indiana University Computer Science Departement, 1994. (Cité page 69.)
- [HDoo] E. Hergenröther et P. Dähne. « Real-time virtual cables based on kinematic simulation ». In *Proc. WSCG 2000, The 8-th International Conference in Central Europe on Computer Graphics, Visualization and Interactive Digital Media*, pages 402–409, 2000. (Cité page 33.)

- [HFS03] G. Hirota, S. Fisher, et A. State. « An improved finite element contact model for anatomical simulations ». *The Visual Computer* 19, 5 (2003), 19 :291–309, 2003. (Cité page 49.)
- [Hiroo] Shinishi Hirai. « Energy-Based of Deformable Linear Objects », 2000. (Cité pages 37 et 45.)
- [HKS98] Thomas Y. Hou, Isaac Klapper, et Helen Si. « Removing the stiffness of curvature in computing 3-D filaments ». *J. Comput. Phys.*, 143(2) :628–664, 1998. (Cité pages 14, 22 et 36.)
- [HM95] Andrew J. Hanson et Hui Ma. « Quaternion Frame Approach to Streamline Visualization ». *IEEE Transactions on Visualization and Computer Graphics*, 1(2) :164–174, 1995. (Cité page 69.)
- [HMC01] Laurent Hilde, Philippe Meseure, et Christophe Chaillou. « A fast implicit integration method for solving dynamic equations of movement ». *ACM symposium on Virtual reality software and technology*, pages 71–76, New York, NY, USA, 2001. ACM Press. (Cité page 77.)
- [HMT00] S. Hadap et N. Magnenat-Thalmann. « Interactive Hair Styler based on Fluid Flow ». Springer-Verlag, éditeur, *Eurographics Workshop on Computer Animation and Simulation*, pages 87–99, 2000. (Cité pages 31 et 176.)
- [HMT01] Sundil Hadap et N. Magnenat-Thalmann. « Modeling dynamic hair as a continuum ». *Computer Graphics Forum, Proceedings of Eurographics'01*, 20 :329–338, 2001. (Cité pages 31 et 32.)
- [How01] Larry. L. Howell. *Compliant Mechanisms*. Wiley-Interscience, 2001. (Cité pages 13, 17, 31, 66, 83, 84, 161 et 169.)
- [Hre41] A. Hrennikoff. « Solution of problems of elasticity by the frame-work method ». *ASME J. Appl. Mech.*, 8 :619–715, 1941. (Cité page 36.)
- [HS52] Magnus R. Hestenes et Eduard Stiefel. « Methods of Conjugate Gradients for Solving Linear Systems ». *Journal of Research of the National Bureau of Standards*, 49 :6, 1952. (Cité page 44.)
- [HTK<sup>+</sup>04] B. Heidelberger, M. Teschner, R. Keiser, M. Mueller, et M. Gross. « Consistent penetration depth estimation for deformable collision response ». *Vision, Modeling, Visualization*, pages 339–346, 2004. (Cité page 49.)
- [Hub96] Philip M. Hubbard. « Approximating polyhedra with spheres for time-critical collision detection ». *ACM Trans. Graph.*, 15(3) :179–210, 1996. (Cité page 48.)
- [Ibr95] A. Ibrahimbegović. « On the finite element implementation of geometrically non-linear Reissner's beam theory : 3d curved beam element ». *Comput. Methods Appl. Mech. Eng.*, 122 :11–26, 1995. (Cité pages 22 et 37.)
- [ISF07] Geoffrey Irving, Craig Schroeder, et Ronald Fedkiw. « Volume conserving finite element simulations of deformable models ». *ACM Transactions on Graphics.*, 26(3) :13, 2007. (Cité page 15.)
- [ITF04] G. Irving, J. Teran, et R. Fedkiw. « Invertible finite elements for robust simulation of large deformation ». *SCA '04 : Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 131–140, Aire-la-Ville, Switzerland, Switzerland, 2004. Eurographics Association. (Cité pages 18 et 172.)
- [JL04] Il-Kwon Jeong et Inho Lee. « An Oriented Particle and Generalized Spring Model for Fast Prototyping Deformable Objects ». *Proceedings of Eurographics 2004 - Short Papers*, 2004. (Cité pages 30 et 53.)
- [JP99] Doug L. James et Dinesh K. Pai. « ArtDefo : accurate real time deformable objects ». *SIGGRAPH '99 : Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 65–72, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co. (Cité pages 11 et 14.)
- [JTT01] P. Jiménez, F. Thomas, et C. Torras. « 3D Collision Detection : A Survey ». *Computers and Graphics*, 25(2) :269–285, April 2001. (Cité page 47.)
- [Kas95] Michael Kass. « An introduction to continuum dynamics for computer graphics ». *SIGGRAPH '95 : Course Notes*, 1995. (Cité page 47.)
- [KB84] Doris H. U. Kochanek et Richard H. Bartels. « Interpolating splines with local tension, continuity, and bias control ». *SIGGRAPH '84 : Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, pages 33–41, New York, NY, USA, 1984. ACM Press. (Cité pages 58 et 59.)

- [KB99] P. Krysl et T. Belytschko. « Application of the element free Galerkin method to the propagation of arbitrary 3D cracks ». *International Journal for Numerical Methods in Engineering*, 44 :767–800, 1999. (Cité page 18.)
- [KGLo7] Ilknur Kabul, Russell Gayle, et Ming C. Lin. « Cable route planning in complex environments using constrained sampling ». Bruno Lévy et Dinesh Manocha, editors, *SPM '07 : Proceedings of the 2007 ACM symposium on Solid and physical modeling*, pages 395–402, New York, NY, USA, 2007. ACM Press. (Cité pages 32, 85 et 169.)
- [KGV83] Kirkpatrick, Gelatt, et Vecchi. « Optimization by simulated annealing ». *Science*, 220 :671–680, 1983. (Cité page 45.)
- [KH01] Chuan Koon Koh et Zhiyong Huang. « A simple Physics model to animate human hair modeled in 2D strips in real time ». *Proceedings of the Eurographic workshop on Computer animation and simulation*, pages 127–138, New York, NY, USA, 2001. Springer-Verlag New York, Inc. (Cité pages 34, 35 et 176.)
- [KiAT93] Tsuneya Kurihara, Ken ichi Anjyo, et Daniel Thalmann. « Hair animation with collision detection ». *Proceedings of Computer Animation '93*, pages 128–138. Springer, 1993. (Cité pages 34, 35, 48 et 49.)
- [Kir59] Gustav Robert Kirchhoff. « Ueber das Gleichgewicht und die Bewegung eines unendlich dünnen elastischen Stabes ». *Journal für die reine und angewandte Mathematik*, 56 :285–312, 1859. (Cité pages 21 et 22.)
- [Klo86] F Klok. « Two moving coordinate frames for sweeping along a 3D trajectory ». *Computer Aided Geometric Design*, 3(3) :217–229, 1986. (Cité pages 68 et 69.)
- [KSW04] Peter Kipfer, Mark Segal, et Rüdiger Westermann. « UberFlow : a GPU-based particle engine ». *HWWS '04 : Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware*, pages 115–122, New York, NY, USA, 2004. ACM Press. (Cité page 28.)
- [Kun91] Donald L. Kunz. « A survey and comparison of engineering beam theories for helicopter rotor blades ». *AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 1991. (Cité page 168.)
- [Lano3] Filip Lankas. « DNA Sequence-Dependent Deformability – Insights from Computer Simulations ». *Biopolymers*, 73(3) :327–339, 2003. (Cité page 172.)
- [LAR02] John H. Long, Bruce Adcock, et Robert G. Root. « Force transmission via axial tendons in undulating fish : a dynamic analysis ». *Comparative Biochemistry and Physiology - Part A : Molecular & Integrative Physiology*, 133(4) :911–929, December 2002. (Cité page 169.)
- [LCDN06a] Julien Lenoir, Stéphane Cotin, Christian Duriez, et Paul Neumann. « Interactive physically-based simulation of catheter and guidewire ». *Computer and Graphics*, 30(3) :417–423, June 2006. (Cité page 38.)
- [LCDN06b] Julien Lenoir, Stéphane Cotin, Christian Duriez, et Paul Neumann. « Physics-Based Models for Catheter, Guidewire and Stent Simulation ». *Proceedings of Medicine Meets Virtual Reality*, pages 305–310, Long Beach (CA) - USA, January 24-27 2006. (Cité pages 38 et 174.)
- [Leno4] Julien Lenoir. « *Modèle déformable 1D pour la simulation physique temps réel* ». Thèse de doctorat, Université des Sciences et Technologies de Lille, 2004. (Cité pages 6, 20, 41, 48, 50, 56, 67, 70, 72, 74, 75, 91, 93, 94, 97 et 131.)
- [LFo4] Julien Lenoir et Sylvère Fonteneau. « Mixing deformable and rigid-body mechanics simulation ». *Computer Graphics International, 2004. Proceedings*, pages 327–334, Hersonissos, Crete - Greece, june 16-19 2004. (Cité page 51.)
- [LGM<sup>+</sup>04] Julien Lenoir, Laurent Grisoni, Philippe Meseure, Yannick Rémond, et Christophe Chaillou. « Smooth constraints for spline variational modeling ». *GRAPHITE '04 : Proceedings of the 2nd international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, pages 58–64, New York, NY, USA, 2004. ACM Press. (Cité pages 41, 50, 61, 78, 94 et 98.)
- [LGMCo5] Julien Lenoir, Laurent Grisoni, Philippe Meseure, et Christophe Chaillou. « Adaptive resolution of 1D mechanical B-spline ». *Proceedings of Graphite*, pages 395–403, Dunedin - New Zealand, 29 november-2 december 2005. (Cité pages 41, 53, 54, 60, 61 et 77.)

- [Lioo] Jing Li. « *A geometrically exact curved beam theory and its finite element formulation/implementation* ». Thèse de doctorat, faculty of Virginia Polytechnic Institute and State University, 2000. (Cité page 37.)
- [LMo3] M. Lin et D. Manocha. *Handbook of Discrete and Computational Geometry*. 2003. (Cité page 48.)
- [LMGCo2] Julien Lenoir, Philippe Meseure, Laurent Grisoni, et Christophe Chaillou. « Surgical Thread Simulation ». Marc THIRIET, éditeur, *Proceedings of Modelling and Simulation for Computer-aided Medicine and Surgery (MS4CMS)*, pages 102–107, Rocquencourt (France), 12-15 nov 2002. EDP Sciences. (Cité pages 40, 41, 48, 49, 61, 67, 77 et 174.)
- [LMGCo4] Julien Lenoir, Philippe Meseure, Laurent Grisoni, et Christophe Chaillou. « A Suture Model for Surgical Simulation ». *Proceedings of the 2nd International Symposium on Medical Simulation (ISMS'04)*, pages 105–113, Cambridge, Massachusetts (USA), 17-18 june 2004. (Cité pages 98, 119 et 175.)
- [LN91] Bernard Le Neindre. « Constantes mécaniques. Coefficients d'élasticité ». Techniques de l'ingénieur. Constantes physico-chimiques, 1991. vol. K4, noK486a, pp. K486.1-K486.23. (Cité page 13.)
- [LP95] C.-L. Lu et N.C. Perkins. « Complex spatial equilibria of u-joint supported cables under torque, thrust and self-weight ». *International Journal of Non-Linear Mechanics*, 30 :271–285(15), May 1995. (Cité page 34.)
- [LPDo3] M. LeDuc, S. Payandeh, et J. Dill. « Toward modeling of a suturing task ». *In Graphics Interface'03 Conference*, pages 273–279, 2003. (Cité page 174.)
- [LRS07] Jie Li, William C. Regli, et Wei Sun. « Mathematical representation of the vascular structure and applications ». *SPM '07 : Proceedings of the 2007 ACM symposium on Solid and physical modeling*, pages 373–378, New York, NY, USA, 2007. ACM Press. (Cité page 174.)
- [LS01] Achim Looock et Elmar Schömer. « A Virtual Environment for Interactive Assembly Simulation : From Rigid Bodies to Deformable Cables ». *Proceedings of the 5th World Multiconference on Systemics, Cybernetics and Informatics (SCI-01) : Emergent computing and virtual engineering*, volume 3, pages 325,332, USA, 2001. (Cité pages 28 et 29.)
- [LSHL02] Itay Lotan, Fabian Schwarzer, Dan Halperin, et Jean-Claude Latombe. « Efficient maintenance and self-collision testing for Kinematic Chains ». *SCG '02 : Proceedings of the eighteenth annual symposium on Computational geometry*, pages 43–52, New York, NY, USA, 2002. ACM Press. (Cité page 48.)
- [LT06] Sung-Hee Lee et Demetri Terzopoulos. « Heads up ! : biomechanical modeling and neuromuscular control of the neck ». *ACM Trans. Graph.*, 25(3) :1188–1198, 2006. (Cité page 173.)
- [LTW95] Yuencheng Lee, Demetri Terzopoulos, et Keith Walters. « Realistic modeling for facial animation ». *SIGGRAPH '95 : Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 55–62, New York, NY, USA, 1995. ACM Press. (Cité page 173.)
- [Mat81] K. Mattiasson. « Numerical Results from Large Deflection Beam and Frame Problems Analyzed by Means of Elliptic Integrals ». *Journal for Numerical Methods in Engineering*, 17 :145–153, 1981. (Cité page 84.)
- [MDCA98] Dermot J. Monaghan, Ian W. Doherty, David Mc Court, et Cecil G. Armstrong. « Coupling 1D Beams to 3D Bodies ». *Proceedings, 7th International Meshing Roundtable, Sandia National Lab*, pages 285–293, 1998. (Cité page 51.)
- [MDM<sup>+</sup>02] Matthias Müller, Julie Dorsey, Leonard McMillan, Robert Jagnow, et Barbara Cutler. « Stable Real-Time Deformations ». *Symposium on Computer Animation*, pages 49–54, San Antonio, Texas, 2002. ACM SIGGRAPH/Eurographics symposium on Computer animation, ACM Press. (Cité pages 11, 14 et 15.)
- [Mey91] Alan Meyer. « A linear time Oslo algorithm ». *ACM Transactions on Graphics*, 10(3) :312–318, 1991. (Cité pages 41 et 60.)
- [Mil91] G. Miller. Goal-Directed Animation of Tubular Articulated Figures or how Snakes Play Golf. N. I. Badler, B. A. Barsky, et D. Zeltzer, editors, *Making Them Move : Mechanics, Control, and Animation of Articulated Figures*, pages 233–241. Morgan Kaufmann, San Mateo, CA, 1991. (Cité page 28.)

- [MKN<sup>+</sup>04] M. Müller, R. Keiser, A. Nealen, M. Pauly, M. Gross, et M. Alexa. « Point based animation of elastic, plastic and melting objects ». *SCA '04 : Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 141–151, Aire-la-Ville, Switzerland, 2004. Eurographics Association. (Cité pages 15 et 18.)
- [MM05] Maher Moakher et John H. Maddocks. « A Double-Strand Elastic Rod Theory ». *Archives for Rational Mechanics and Analysis*, 177 :53–91, 2005. (Cité page 172.)
- [MMA99] Oleg Mazarak, Claude Martins, et John Amanatides. « Animating Exploding Objects ». *Graphics Interface*, pages 211–218, 1999. (Cité page 19.)
- [MMDJ01] Matthias Müller, Leonard McMillan, Julie Dorsey, et Robert Jagnow. « Real-time simulation of deformation and fracture of stiff materials ». *Proceedings of the Eurographic workshop on Computer animation and simulation*, pages 113–124, New York, NY, USA, 2001. Springer-Verlag New York, Inc. (Cité page 18.)
- [MMK96] R.S. Manning, J.H. Maddocks, et J.D. Kahn. « A continuum rod model of sequence-dependent DNA structure ». *The Journal of Chemical Physics*, 105 :5626–5646, October 1996. (Cité pages 34 et 172.)
- [MN97] Derek Morrison et Gregory Neff. « Lofting and conics in the design of aircraft. ». *Proceedings of the ASEE Illinois/Indiana Section Conference*, pages 222–227, Indianapolis, Indiana, March 1997. (Cité page 56.)
- [Nah91] H. Nahvi. « *Static and Dynamic Analysis of Compliant Mechanisms Containing Highly Flexible Members* ». Thèse de doctorat, Purdue University, West Lafayette, Indiana., 1991. (Cité page 31.)
- [Nau05] Joris Naudet. « *Forward Dynamics of Multibody Systems : A Recursive Hamiltonian Approach* ». Thèse de doctorat, Faculteit Ingenieurswetenschappen Vakgroep Toegepaste Mechanica, 2005. (Cité page 168.)
- [Nis97] T. Nishioka. « Computational dynamic fracture mechanics ». *International Journal of Fracture*, 86 :127–159, 1997. (Cité page 17.)
- [NMK<sup>+</sup>05] A. Nealen, M. Muller, R. Keiser, E. Boxerman, et M. Carlson. « Physically based deformable models in computer graphics ». In *Eurographics : State of the Art Report.*, 2005. (Cité page 15.)
- [Noco01] Olivier Nocent. « *Animation dynamique de corps déformables continus : application à la simulation de textiles tricotés* ». Thèse de doctorat, Université de Reims Champagne-Ardenne, 2001. (Cité pages 40, 56 et 72.)
- [Nou99] Jean-Michel Nourrit. « *Modélisation, animation et visualisation de textiles à base de mailles* ». Thèse de doctorat, Université de Reims Champagne-Ardenne, 1999. (Cité pages 29, 40, 46, 61, 67, 72, 104 et 169.)
- [NR01] O. Nocent et Y. Remion. « Continuous deformation energy for dynamic material splines subject to finite displacements ». *CAS '01 : Proceedings of the Eurographic workshop on Computer animation and simulation*, pages 88–97, New York, NY, USA, 2001. Springer-Verlag New York, Inc. (Cité pages 40, 41, 56, 61, 67, 74, 77 et 169.)
- [NTB<sup>+</sup>91] Alan Norton, Greg Turk, Bob Bacon, John Gerth, et Paula Sweeney. « Animation of fracture by physical modeling ». *Vis. Comput.*, 7(4) :210–219, 1991. (Cité page 18.)
- [NTH01] Victor Ng-Thow-Hing. « *Anatomically-based models for physical and geometric reconstruction of humans and other animals* ». Thèse de doctorat, University of Toronto, 2001. Adviser-Eugene Fiume. (Cité page 172.)
- [NvdHo2] S. Neukirch et G.H.M. van der Heijden. « Geometry and Mechanics of Uniform n-Plies : from Engineering Ropes to Biological Filaments ». *Journal of Elasticity*, 69 :41–72, 2002. (Cité page 171.)
- [NW67] Ahlberg Nielson et Walsh. *The Theory of Splines and Their Applications*. Academic Press, 1967. (Cité page 56.)
- [OBHo2] James F. O'Brien, Adam W. Bargteil, et Jessica K. Hodgins. « Graphical modeling and animation of ductile fracture ». *SIGGRAPH '02 : Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 291–294, New York, NY, USA, 2002. ACM Press. (Cité page 18.)
- [Ode90] J. Tinsley Oden. « Historical comments on finite elements », 1990. (Cité page 36.)

- [OH99] James F. O'Brien et Jessica K. Hodgins. « Graphical modeling and animation of brittle fracture ». *SIGGRAPH '99 : Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 137–146, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co. (Cité page 18.)
- [Osh07] Masaki Oshita. « Real-time Hair Simulation on GPU with a Dynamic Wisp Model ». *International Conference on Computer Animation and Social Agents*, 2007. (Cité page 28.)
- [Paio2] Dinesh K. Pai. « STRANDS : Interactive Simulation of Thin Solids using Cosserat Models ». *Computer Graphics Forum, Proceedings of Eurographics'02.*, 21 :347–352, 2002. (Cité pages 30, 34, 35, 49, 53, 54, 69, 85 et 175.)
- [PAK00] D.K. Pai, U.M. Ascher, et P.G. Kry. « Forward dynamics algorithms for multibody chains and contact ». *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, volume 1, pages 857–863, 2000. (Cité page 32.)
- [PB88] John C. Platt et Alan H. Barr. « Constraints methods for flexible models ». *SIGGRAPH '88 : Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, pages 279–288, New York, NY, USA, 1988. ACM Press. (Cité page 49.)
- [PCP01] Eric Plante, Marie-Paule Cani, et Pierre Poulin. « A layered wisp model for simulating interactions inside long hair ». *Proceedings of the Eurographic workshop on Computer animation and simulation*, pages 139–148, New York, NY, USA, 2001. Springer-Verlag New York, Inc. (Cité pages 29, 48 et 176.)
- [PDA00] Guillaume Picinbono, Herve Delingette, et Nicholas Ayache. « Real-Time Large Displacement Elasticity for Surgery Simulation : Non-linear Tensor-Mass Model ». *MICCAI '00 : Proceedings of the Third International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 643–652, London, UK, 2000. Springer-Verlag. (Cité pages 11 et 14.)
- [PFTV88a] William H. Press, Brian P. Flannery, Saul A. Teukolsky, et William T. Vetterling. « *Numerical Recipes in C* », Chapitre 10. Minimization or Maximization of Functions, pages 394–455. Cambridge University Press, 1988. (Cité page 44.)
- [PFTV88b] William H. Press, Brian P. Flannery, Saul A. Teukolsky, et William T. Vetterling. « *Numerical Recipes in C* », Chapitre 4. Integration of Functions, pages 129–161. Cambridge University Press, 1988. (Cité page 74.)
- [Pie91] Les Piegl. « On NURBS : a survey ». *Computer Graphics and Applications*, 11(1) :55–71, 1991. (Cité page 58.)
- [PKA<sup>+</sup>05] Mark Pauly, Richard Keiser, Bart Adams, Philip Dutré, Markus Gross, et Leonidas J. Guibas. « Meshless animation of fracturing solids ». *SIGGRAPH '05 : ACM SIGGRAPH 2005 Papers*, pages 957–964, New York, NY, USA, 2005. ACM Press. (Cité page 19.)
- [Pla92] John Platt. « A generalization of dynamic constraints ». volume 54, pages 516–525, Orlando, FL, USA, 1992. Academic Press, Inc. (Cité page 49.)
- [PLK02] J. Phillips, A. Ladd, et L.E. Kavraki. « Simulated knot tying ». *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, volume 1, pages 841–846, 2002. (Cité pages 29, 41, 48 et 174.)
- [Prio3] Dominique Primault. « *Modélisation géométriquement exacte de poutres fines - application à la robotique*. ». Thèse de doctorat, ED STIM, 2003. (Cité pages 20, 21, 37, 59 et 68.)
- [Pro95] Xavier Provot. « Deformation constraints in a mass-spring model to describe rigid cloth behavior. ». Wayne A. Davis et Przemyslaw Prusinkiewicz, éditeur, *Proceedings of Graphics Interface '95*, pages 147–154. Canadian Human-Computer Communications Society, 1995. (Cité page 28.)
- [Prz68] J. S. Przemieniecki. *Theory of matrix structural analysis*. New York : McGraw-Hill Book Company, 1968. (Cité page 38.)
- [PRZ92] Steven Pieper, Joseph Rosen, et David Zeltzer. « Interactive graphics for plastic surgery : a task-level analysis and implementation ». *SI3D '92 : Proceedings of the 1992 symposium on Interactive 3D graphics*, pages 127–134, New York, NY, USA, 1992. ACM Press. (Cité page 14.)
- [PSW05] Dinesh K. Pai, Shinjiro Sueda, et Qi Wei. « Fast physically based musculoskeletal simulation ». *SIGGRAPH '05 : ACM SIGGRAPH 2005 Sketches*, page 25, New York, NY, USA, 2005. ACM Press. (Cité pages 172 et 173.)

- [PT97] Les Piegl et Wayne Tiller. *The NURBS book (2nd ed.)*. Springer-Verlag New York, Inc., New York, NY, USA, 1997. (Cité pages 58 et 60.)
- [PW89] Alex Pentland et J. Williams. « Good vibrations : model dynamics for graphics and animation ». *SIGGRAPH '89 : Proceedings of the 16th annual conference on Computer graphics and interactive techniques*, pages 215–222, New York, NY, USA, 1989. ACM Press. (Cité page 42.)
- [QT96] Hong Qin et D. Terzopoulos. « D-NURBS : a physics-based framework for geometric design ». *Visualization and Computer Graphics, IEEE Transactions on*, 2 :85–96, 1996. (Cité page 39.)
- [R00] Yannick Rémon. « Animation Dynamique : moteur lagrangien généraliste et applications ». HDR, 2000. (Cité pages 40 et 49.)
- [RCT91] R. Rosenblum, W. Carlson, et E. Tripp. « Simulating the structure and dynamics of human hair : Modeling, rendering and animation ». *The Journal of Visualization and Computer Animation*, 2 :141–148, 1991. (Cité page 28.)
- [Red02] Stéphane Redon. « Algorithmes de simulation dynamique interactive d'objets rigides ». Thèse de doctorat, Université d'Évry, 2002. (Cité page 47.)
- [Rei73] E. Reissner. « On a one-dimensional large displacement finite-strain beam theory ». *Stud. Appl. Math.*, 52 :87–95, 1973. (Cité page 22.)
- [RGF<sup>+</sup>04] L. Raghupathi, L. Grisoni, F. Faure, D. Marchal, M.-P. Cani, et C. Chaillou. « An intestinal surgery simulator : real-time collision processing and visualization ». *IEEE Transactions on Visualization and Computer Graphics*, 10 :708–718, 2004. (Cité pages 48, 70, 108 et 173.)
- [RGL05] Stéphane Redon, Nico Galoppo, et Ming C. Lin. « Adaptive dynamics of articulated bodies ». *SIGGRAPH '05 : ACM SIGGRAPH 2005 Papers*, pages 936–945, New York, NY, USA, 2005. ACM Press. (Cité pages 32, 49 et 88.)
- [Rie73] R. Riesenfeld. « Applications of B-spline approximation to geometric problems of computer aided-design ». Thèse de doctorat, University of Syracuse, 1973. (Cité page 58.)
- [RJ07] Alec R. Rivers et Doug L. James. « FastLSM : fast lattice shape matching for robust real-time deformation ». *ACM Trans. Graph.*, 26(3) :82, 2007. (Cité pages 11 et 15.)
- [RKLM04] Stéphane Redon, Young J. Kim, Ming C. Lin, et Dinesh Manocha. « Fast continuous collision detection for articulated models ». *SM '04 : Proceedings of the ninth ACM symposium on Solid modeling and applications*, pages 145–156, Aire-la-Ville, Switzerland, Switzerland, 2004. Eurographics Association. (Cité page 48.)
- [Rémo3] Yannick Rémon. « Prise en compte de contraintes à variables libres. ». Technical Report 03-02-01, LERI - Université de Reims, 2003. (Cité page 98.)
- [RNG99] Yannick Rémon, Jean-Michel Nourrit, et Didier Gillard. « Dynamic Animation of Spline Like Objects ». V. Skala, éditeur, *WSCG'99 Conference Proceedings*, 1999. (Cité pages 29, 40, 46, 76 et 169.)
- [RNG00] Yannick Rémon, Jean-Michel Nourrit, et Didier Gillard. « A dynamic animation engine for generic spline objects ». *Journal of Visualization and Computer Animation*, 11(1) :17–26, 2000. (Cité pages 40 et 49.)
- [RNN00] Yannick Rémon, Jean-Michel Nourrit, et Olivier Nocent. « Dynamic Animation of N-Dimensional Deformable Objects ». *WSCG'00 Conference Proceedings*, 2000. (Cité page 40.)
- [RNN01] Y. Remion, J.-M. Nourrit, et O. Nocent. « D-dimensional parametric models for dynamic animation of deformable objects. ». *The Visual Computer Journal*, 17(3) :167–178, oct 2001. (Cité page 51.)
- [RO06] J. Z. Ruscio et Alexey Onufriev. « A Computational Study of Nucleosomal DNA Flexibility ». *Biophys. J.*, 91 :4121–4132, 2006. (Cité page 172.)
- [Rob] « Projet Robea : Robotique et entités artificielles ». <http://www.laas.fr/robea/>. (Cité page 169.)
- [Rop54] Kenneth L. Roper. « Suturing in Cataract Surgery ». *Trans Am Ophthalmol Soc.*, 52 :587–749, 1954. (Cité page 175.)
- [Ruboo] M. B. Rubin. *Cosserat Theories : Shells, Rods and Points*. Kluwer Academic Publ., Dordrecht., 2000. (Cité page 34.)

- [SB84] U. Shani et D. Ballard. « Splines as embeddings for generalized cylinders ». *Computer Vision, Graphics and Image Processing*, 27 :129–156, 1984. (Cité page 69.)
- [SB99] Terence Strick et David Bensimon. « *Enroulement mécanique de l'ADN et relaxation par les topoisomérases (Mechanical supercoiling of DNA and its relaxation by topoisomerases)* ». Thèse de doctorat, Université de Paris 06, 1999. (Cité page 171.)
- [Sch46a] I. J. Schoenberg. « Contributions to the problem of approximation of equidistant data by analytic functions, Part A : On the problem of smoothing or graduation, a first class of analytic approximation formulas ». *Quart. Appl. Math.*, 4 :45–99, 1946. (Cité page 56.)
- [Sch46b] I. J. Schoenberg. « Contributions to the problem of approximation of equidistant data by analytic functions, Part B : On the problem of osculatory interpolation, a second class of analytic approximation formulae ». *Quart. Appl. Math.*, 4 :112–141, 1946. (Cité page 56.)
- [Sch81] L. L. Schumaker. *Spline Functions : Basic Theory*. John Wiley & Sons, New York., 1981. (Cité page 41.)
- [SF73] G. Strang et G. J. Fix. *An Analysis of the Finite Element Method*. Prentice-Hall, Englewood Cliffs, 1973. 306 pp. ISBN 0-13-032946-0. (Cité page 36.)
- [SF89] J. C. Simo et D. D. Fox. « On stress resultant geometrically exact shell model. Part I : formulation and optimal parametrization ». *Comput. Methods Appl. Mech. Eng.*, 72(3) :267–304, 1989. (Cité page 37.)
- [She94] Jonathan R. Shewchuk. « An Introduction to the Conjugate Gradient Method Without the Agonizing Pain ». Technical Report CS-94-125, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA, 1994. (Cité page 44.)
- [Shio5] Ofek Shilon. « Simulating Bending Behaviour of Thread and Needle ». Technical Report, Simbionix, 2005. (Cité page 175.)
- [Sim85] J. C. Simo. « A finite strain beam formulation. The three-dimensional dynamic problem. Part I ». *Computer Methods in Applied Mechanics and Engineering*, 49 :55–70, 1985. (Cité pages 22 et 34.)
- [SMH01a] Yaoming Shi, W. M. McClain, et J. E. Hearst. « A Lax Pair for the Dynamics of DNA Modeled as a Shearable and Extensible Elastic Rod », 2001. (Cité page 171.)
- [SMH01b] Yaoming Shi, W. M. McClain, et J. E. Hearst. « A Lax Pair for the Dynamics of DNA Modeled as a Shearable and Extensible Elastic Rod : II. Discretization of the Arc Length », 2001. (Cité page 171.)
- [SMH03] Yaoming Shi, W. M. McClain, et J. E. Hearst. « A Lax Pair for the Dynamics of DNA Modeled as a Shearable and Extensible Elastic Rod : III. Discretization of Arc Length and Time », 2003. (Cité page 171.)
- [SMMB00] N. Sukumar, N. Moës, B. Moran, et T. Belytschko. « Extended finite element method for three-dimensional crack modelling ». *International Journal for Numerical Methods in Engineering*, 48(11) :1549–1570, 2000. (Cité page 18.)
- [SNF05] Eftychios Sifakis, Igor Neverov, et Ronald Fedkiw. « Automatic determination of facial muscle activations from sparse motion capture marker data ». *SIGGRAPH '05 : ACM SIGGRAPH 2005 Papers*, pages 417–425, New York, NY, USA, 2005. ACM Press. (Cité page 173.)
- [SSIF07] Eftychios Sifakis, Tamar Shinar, Geoffrey Irving, et Ronald Fedkiw. « Hybrid simulation of deformable solids ». *SCA '07 : Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 81–90, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association. (Cité page 51.)
- [ST07] J. Spillmann et M. Teschner. « CORDE : Cosserat Rod Elements for the Dynamic Simulation of One-Dimensional Elastic Objects ». *SCA '07 : Proceedings of the 2007 ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, 2007. (Cité pages 38, 50, 53, 54 et 85.)
- [Sta97] Jos Stam. « Stochastic Dynamics : Simulating the Effects of Turbulence on Flexible Structures ». *Proceedings of Eurographics'97*, 1997. (Cité page 42.)
- [SVH99] G.N. Shuttleworth, L.F. Vaughn, et Hoh H.B. « Material properties of ophthalmic sutures after sterilization and disinfection ». *Journal of Cataract and Refractive Surgery*, 25 :1270–4, 1999. (Cité pages 80 et 175.)

- [Sim88] J. C. Simo et L. Vu-Quoc. « On the dynamics in space of rods undergoing large motions : a geometrically exact approach. ». *Computer Methods in Applied Mechanics and Engineering*, 66 :125–161, 1988. (Cité page 22.)
- [SWBoo] Jeffrey Smith, Andrew Witkin, et David Baraff. « Fast and Controllable Simulation of the Shattering of Brittle Objects ». *Graphics Interface*, May 2000. (Cité page 18.)
- [TC90] Jeffrey A. Thingvold et Elaine Cohen. « Physical modeling with B-spline surfaces for interactive design and animation ». *SIGGRAPH Comput. Graph.*, 24(2) :129–137, 1990. (Cité pages 28 et 39.)
- [Ter86] Demetri Terzopoulos. « Regularization of inverse visual problems involving discontinuities ». *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(4) :413–242, 1986. (Cité page 17.)
- [TF88] Demetri Terzopoulos et Kurt Fleischer. « Modeling inelastic deformation : viscoelasticity, plasticity, fracture ». *SIGGRAPH '88 : Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, pages 269–278, New York, NY, USA, 1988. ACM Press. (Cité pages 17 et 18.)
- [TGAB07] Adrien Theetten, Laurent Grisoni, Claude Andriot, et Brian Barsky. « Geometrically Exact Dynamic Splines ». *Journal of Computer Aided Design : special issue on Constrained Design of Curves and Surfaces*, 2007. (Cité pages 55, 89 et 137.)
- [TGDM07] Adrien Theetten, Laurent Grisoni, Christian Duriez, et Xavier Merlhiot. « Quasi-dynamic splines ». Bruno Lévy et Dinesh Manocha, editors, *SPM '07 : Proceedings of the 2007 ACM symposium on Solid and physical modeling*, pages 409–414, New York, NY, USA, 2007. ACM Press. (Cité pages 55 et 138.)
- [Til92] Wayne Tiller. « Knot-removal algorithms for NURBS curves and surfaces ». *Computer-Aided Design*, 24(8) :445–453, 1992. (Cité page 60.)
- [TKH<sup>+</sup>05] M. Teschner, S. Kimmerle, B. Heidelberger, G. Zachmann, L. Raghupathi, A. Fuhrmann, M.-P. Cani, F. Faure, N. Magnenat-Thalmann, W. Strasser, et P. Volino. « Collision Detection for Deformable Objects ». *Computer Graphics Forum*, 24 :61–81, 2005. (Cité page 47.)
- [TPBF87] Demetri Terzopoulos, John Platt, Alan Barr, et Kurt Fleischer. « Elastically deformable models ». *SIGGRAPH '87 : Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 205–214, New York, NY, USA, 1987. ACM Press. (Cité pages 14, 36, 39 et 46.)
- [TQ94] Demetri Terzopoulos et Hong Qin. « Dynamic NURBS with geometric constraints for interactive sculpting ». *ACM Trans. Graph.*, 13(2) :103–136, 1994. (Cité pages 39 et 40.)
- [TSFo5] Winnie Tsang, Karan Singh, et Eugene Fiume. « Helping hand : an anatomically accurate inverse dynamics solution for unconstrained hand motion ». *SCA '05 : Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 319–328, New York, NY, USA, 2005. ACM Press. (Cité page 173.)
- [TW88] Demetri Terzopoulos et Andrew Witkin. « Physically Based Models with Rigid and Deformable Components ». *IEEE Computer Graphics and Applications*, 8(6) :41–51, 1988. (Cité page 36.)
- [Twi03] Christopher Twigg. « Catmull-Rom splines », March 2003. (Cité page 59.)
- [UL07] Scott Uttley et Stephen Lane. « *Ophthalmic Microsurgical Suturing Techniques* », Chapitre Microsurgical Suturing Techniques : Closure of the Cataract Wound, pages 29–35. Springer Berlin Heidelberg, 2007. (Cité page 175.)
- [vdB97] G. van den Bergen. « Efficient collision detection of complex deformable models using AABB trees ». *J. of Graphics Tools*, 2(4) :1–13, 1997. (Cité page 48.)
- [vdHNGT03] G.H.M. van der Heijden, Sébastien Neukirch, V. G. A. Goss, et J.M.T. Thompson. « Instability and self-contact phenomena in the writhing of clamped rods ». *International Journal of Mechanical Science*, 45 :161–196, 2003. (Cité page 35.)
- [Vel92] R. C. Veltkamp. « Survey of continuities of curves and surfaces ». *Computer Graphics Forum*, 11(2) :93–112, 1992. (Cité page 68.)
- [Ver75] Kenneth James Versprille. « *Computer-aided design applications of the rational b-spline approximation form*. ». Thèse de doctorat, 1975. (Cité page 63.)

- [VMT06] Pascal Volino et Nadia Magnenat-Thalmann. « Real-Time Animation of Complex Hairstyles ». *IEEE Transactions on Visualization and Computer Graphics*, 12 :131–142, 2006. (Cité page 176.)
- [VQDE96] L. Vu-Quoc, H. Deng, et I.K. Ebcioğlu. « Multilayer Beams : A Geometrically Exact Formulation ». *Journal of Nonlinear Science*, 6 :239–270, 1996. (Cité pages 22 et 37.)
- [WAH06] Hidefumi Wakamatsu, Eiji Arai, et Shinichi Hirai. « Knotting/Unknotting Manipulation of Deformable Linear Objects ». *The International Journal of Robotics Research*, 25(4) :371–395, 2006. (Cité page 37.)
- [Wak98] Wakamatsu. « Analysis of Hysteresis in Deformation of Rodlike Objects Toward Their Manipulation ». *Proceedings of the 4th Japan-France Congress & 2nd Asia-Europe Congress on Mechatronics*, 1998. (Cité page 37.)
- [WBC07] Jamie Wither, Florence Bertails, et Marie-Paule Cani. « Realistic Hair from a Sketch ». *Shape Modeling International*, June 2007. (Cité pages 176 et 177.)
- [WBD<sup>+</sup>05] F. Wang, E. Burdet, A. Dhanik, T. Poston, et C. L. Teo. « Dynamic Thread for Real-Time Knot-Tying ». *WHC '05 : Proceedings of the First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 507–508, Washington, DC, USA, 2005. IEEE Computer Society. (Cité pages 30, 48, 53 et 175.)
- [WBK<sup>+</sup>07] Kelly Ward, Florence Bertails, Tae-Yong Kim, Stephen R. Marschner, et Marie-Paule Cani. « A Survey on Hair Modeling : Styling, Simulation, and Rendering ». *IEEE Transactions on Visualization and Computer Graphics*, 13(2) :213–234, 2007. Member-Ming C. Lin. (Cité pages 20 et 177.)
- [WBVB05] F. Wang, E. Burdet, R. Vuillemin, et H. Bleuler. « Knot-tying with Visual and Force Feedback for VR Laparoscopic Training ». *Proceedings of the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, 2005. (Cité pages 30, 48 et 175.)
- [WC53] James Dewey Watson et Francis Crick. « Molecular structure of Nucleic acids : A structure for Deoxyribose Nucleic Acid ». *Nature*, 171 :738–740, 1953. (Cité page 171.)
- [WDGT01] X. Wu, M. Downes, T. Goketin, et F. Tendick. « Adaptive nonlinear finite elements for deformable body simulation using dynamic progressive meshes ». *Proc. Eurographics*, volume 20, pages 349–358, 2001. (Cité page 128.)
- [Wei02] H. Weiss. « Dynamics of Geometrically Nonlinear Rods : II. Numerical Methods and Computational Examples ». *Nonlinear Dynamics*, 30 :383–415, 2002. (Cité page 166.)
- [Wes64] H.M. Westergaard. *Theory of elasticity and plasticity*. New York : Dover Publications, 1964. (Cité page 12.)
- [WGL04] Kelly Ward, Nico Galoppo, et Ming C. Lin.. « Modeling Hair Influenced by Water and Styling Products ». *Proceedings of Computer Animation and Social Agents*, 2004. (Cité page 176.)
- [WGL06] Kelly Ward, Nico Galoppo, et Ming C. Lin. « A Simulation-based VR System for Interactive Hairstyling ». *Proceedings of IEEE Virtual Reality Conference, Applications and Research Sketches.*, 2006. (Cité pages 34 et 176.)
- [WH04] Hidefumi Wakamatsu et Shinichi Hirai. « Static Modeling of Linear Object Deformation Based on Differential Geometry ». *The International Journal of Robotics Research*, 23 :293–311, 2004. (Cité pages 37, 53, 68 et 89.)
- [WHI95] H. Wakamatsu, S. Hirai, et K. Iwata. « Modeling of linear objects considering bend, twist and extensional deformations ». *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, volume 1, pages 433–438, 1995. (Cité pages 37, 43 et 45.)
- [WLo3] Kelly Ward et Ming C. Lin. « Adaptive Grouping and Subdivision for Simulating Hair Dynamics ». *PG '03 : Proceedings of the 11th Pacific Conference on Computer Graphics and Applications*, page 234, Washington, DC, USA, 2003. IEEE Computer Society. (Cité pages 34, 35 et 43.)
- [WLL<sup>+</sup>03] Kelly Ward, Ming C. Lin, Joohi Lee, Susan Fisher, et Dean Macri. « Modeling Hair Using Level-of-Detail Representations ». *Proc. of Computer Animation and Social Agents*, 2003. (Cité page 176.)
- [WMHA97] Hidefumi Wakamatsu, Takumi Matsumura, Shinichi Hirai, et Eiji Arai. « Dynamic Analysis of Rodlike Object Deformation towards Their Dynamic Manipulation ». *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 196–201, Grenoble, France, 1997. (Cité page 37.)

- [WMW<sup>+</sup>00] Takahiro Wada, Brenan J. McCarragher, Hidefumi Wakamatsu, Shinichi Hirai, Takeshi Yonezawa, et Shinichi Tokumoto. « Modeling of Hysteresis in Deformation of Rodlike Objects Toward Their Manipulation ». *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, volume 1, pages 203–208, October 2000. (Cité pages 37 et 45.)
- [WR99] E. Wilson et Clough R.. « Early Finite Element Research at Berkeley ». *Fifth U.S. National Conference on Computational Mechanics*, 1999. (Cité page 36.)
- [WSG05] Martin Wicke, Denis Steinemann, et Markus Gross. « Efficient Animation of Point-Based Thin Shells ». *Proceedings of Eurographics '05*, pages 667–676, 2005. (Cité page 19.)
- [WW90] Andrew Witkin et William Welch. « Fast animation and control of nonrigid structures ». *SIGGRAPH '90 : Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, pages 243–252, New York, NY, USA, 1990. ACM Press. (Cité pages 29 et 40.)
- [WZM<sup>+</sup>01] Roger W. Webster, Dean I. Zimmerman, Betty J. Mohler, Michael G. Melkonian, et Randy S. Haluck. « A Prototype Haptic Suturing Simulator ». *Proceedings of Medicine Meets Virtual Reality*, 2001. (Cité page 174.)
- [YOH00] Gary D. Yngve, James F. O'Brien, et Jessica K. Hodgins. « Animating explosions ». *SIGGRAPH '00 : Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 29–36, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co. (Cité page 19.)
- [Yuo01] Yizhou Yu. « Modeling Realistic Virtual Hairstyles ». IEEE Computer Society Press, éditeur, *Proceedings of Pacific Graphics*, pages 295–304, 2001. (Cité page 176.)
- [ZC00] Yan Zhuang et John F. Canny. « Haptic Interaction with Global Deformations ». *Proceedings of ICRA*, 2000. (Cité pages 14 et 15.)
- [ZCCD04] Victor Brian Zordan, Bhrigu Celly, Bill Chiu, et Paul C. DiLorenzo. « Breathe easy : model and control of simulated respiration for animation ». *SCA '04 : Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 29–37, Aire-la-Ville, Switzerland, Switzerland, 2004. Eurographics Association. (Cité page 173.)
- [ZRLK07] Xinyu Zhang, Stephane Redon, Minkyung Lee, et Young J. Kim. « Continuous Collision Detection for Articulated Models using Taylor Models and Temporal Culling ». *Proceedings of Siggraph 2007*, 2007. (Cité page 48.)

# Annexes

## A NOTATIONS

Dans un but de clarté, nous résumons ici l'ensemble des notations utilisées dans ce document. Les vecteurs sont notés en gras ( $\mathbf{v}$ ), les matrices par des lettres doubles ( $\mathbb{M}$ ),  $\mathbb{I}$  désignant la matrice identité. La  $i^{\text{ème}}$  composante d'un vecteur est noté  $v_i$ .

### Algèbre

- $\mathbf{u} \cdot \mathbf{v}$  Produit scalaire entre les deux vecteurs  $\mathbf{u}$  et  $\mathbf{v}$
- $\mathbf{u} \times \mathbf{v}$  Produit vectoriel entre les deux vecteurs  $\mathbf{u}$  et  $\mathbf{v}$
- $\mathbf{u}^t$  Transposée du vecteur  $\mathbf{v}$
- $\mathbb{M}^t$  Transposée de la matrice  $\mathbb{M}$

### Opérateurs différentiels

Soit  $f$  une fonction scalaire des variables  $x_1, \dots, x_i, \dots, x_n$  du temps  $t$  et de l'abscisse  $u$ .

- $f'$  dérivée première de  $f$  par rapport à  $u$  :  $\frac{df}{du}$
- $\dot{f}$  dérivée temporelle de  $f$  :  $\frac{df}{dt}$
- $\frac{\partial f}{\partial x_i}$  dérivée partielle de  $f$  par rapport à  $x_i$ , sous la forme d'un vecteur de  $i$  composantes.
- $\frac{\partial^2 f}{\partial x_j \partial x_i}$  dérivée partielle de  $f$  par rapport à  $x_i$  puis à  $x_j$ , d'une matrice de  $i$  lignes et  $j$  colonnes

### Grandeurs mécaniques et géométriques d'une spline matérielle

- $g$  gravité
- $\rho$  masse volumique
- $\mu$  masse linéique
- $E$  module d'Young
- $\nu$  coefficient de Poisson
- $G$  module de cisaillement
- $D$  diamètre de la section
- $A$  aire de la section
- $J$  module de rigidité à la torsion
- $I_0$  moment d'inertie polaire
- $I_y$  moment d'inertie par rapport à l'axe  $y$

### Systèmes matriciels

- $\mathbf{F}$  vecteur des forces externes
- $\mathbf{P}$  vecteur des forces/moments internes
- $\mathbf{X}$  vecteur de position
- $\Delta \mathbf{X}$  vecteur de translation
- $\mathbf{V}$  vecteur vitesse
- $\Lambda$  vecteur de multiplicateurs de Lagrange
- $\mathbf{E}$  vecteur de violation de multiplicateur de Lagrange
- $\mathbb{K}$  matrice de rigidité
- $\mathbb{M}$  matrice des masses/d'inertie
- $\mathbb{D}$  matrice de viscosité dynamique
- $\mathbb{H}$  matrice de Hooke
- $\mathbb{J}$  matrice de densité d'inertie
- $\mathbb{L}$  matrice de multiplicateur de Lagrange

## B LEXIQUE

Nous donnons ici les correspondances entre les termes français et anglais de mécanique et d'informatique graphique que l'on trouve dans la littérature. Certains prêtent en effet à confusion et peuvent induire la mauvaise compréhension d'une publication.

Area moment of inertia	Moment d'inertie axial
Assumption	Hypothèse
Beam	poutre
Boundary Element Method	Méthode des éléments de surface
Brittle	Cassant
Compliant	Déformable
Creep	Fluage
Deflection	Déformation (globale)
Deformation	Déformation (globale)
Ductile	Ductile
Failure	Défaillance
Fatigue	Fatigue
Finite Element Method	Méthode des éléments finis
Linear spring	ressort longitudinal
Lattice	Réseau
Load	Charge
Plate	pile (de pont)
Polar moment of inertia	Moment d'inertie polaire
Rod	tige
Strain	Déformation (locale)
Stress	contrainte
Safety factor	facteur de sécurité
Shearable/unshearable	pouvant (ne) être (pas) cisailé
Stiffness	raideur, rigidité
Strength	résistance
Tensile	traction
Torque	moment
Torsion	torsion (géométrique)
Torsion spring	ressort angulaire
Twisting	torsion (matérielle)
Wrench	torseur

## C MATÉRIAUX

Les paramètres des matériaux ne sont pas toujours évidents à trouver. Afin d'éviter des recherches laborieuses au lecteur, nous donnons les quelques paramètres dont nous nous sommes servis.

Matériau	Mod. d'Young $E$ (GPa)	Coef. de Poisson $\nu$	Masse volumique $\mu$ (g/cm <sup>3</sup> )
Nylon	2,7	0,33	1,1
Cheveu	0,1-3	0,3-0,48	1,3
Aluminium	71,7	0,34	2,8
Cuivre	120,7	0,35	8,9
Acier, carbone	206,8	0,28	7,8
Acier, sans étain	189,6	0,28	7,8

## D PROPRIÉTÉS DES SECTIONS

Cette annexe contient les résultats nécessaires aux calculs de raideurs de diverses sections d'objets 1D :

- le périmètre  $P$  extérieur de la section
- l'aire  $A$
- les moments d'inertie axiaux  $I_x$  et  $I_y$ , donnés par les intégrales suivantes :  $I_x = \int x^2 dA$  et  $I_y = \int y^2 dA$
- le moment d'inertie polaire  $I_o$  par rapport à l'axe orthogonal au plan de la section, défini par  $I_o = \int r^2 dA = \int (x^2 + y^2) dA = I_x + I_y$
- le module de rigidité à la torsion  $J$ , à ne pas confondre avec le moment d'inertie polaire, même si identiques pour une section circulaire (Cou8oc; Gar99). Pour une section quelconque possédant deux axes de symétrie rectangulaires, il peut être calculé à l'aide de la formule suivante (Cou8oc) :  $J = \frac{A^4}{4\pi^2(I_x + I_y)}$ .

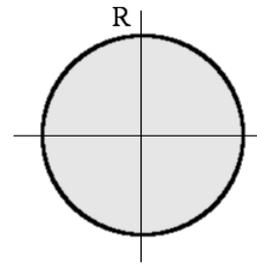
Le repère  $(G, x, y, z)$  a toujours pour origine le centre de gravité de la section.

Nous vous proposons les sections circulaire, tubulaire, elliptique, triangulaire, carrée, rectangulaire et en double T. Pour d'autres types de sections, veuillez consulter l'annexe B de (Howo1).

### D.1 Section circulaire

Pour une section circulaire de rayon  $R$  et de diamètre  $D = 2R$  :

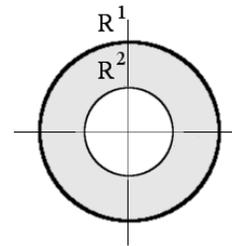
$$\begin{aligned} P &= \pi D = 2\pi R \\ A &= \frac{\pi D^2}{4} = \pi R^2 \\ I_x &= I_y = \frac{\pi D^4}{64} = \frac{\pi R^4}{4} \\ I_o &= J = \frac{\pi D^4}{32} = \frac{\pi R^4}{2} \end{aligned}$$



### D.2 Section tubulaire

Pour une section tubulaire de grand diamètre  $D = 2R^1$  et de petit diamètre  $d = 2R^2$  :

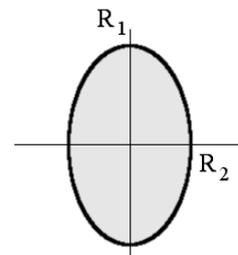
$$\begin{aligned} P &= \pi D = 2\pi R \\ A &= \frac{\pi}{4}(D^2 - d^2) \\ I_x &= I_y = \frac{\pi}{64}(D^4 - d^4) \\ I_o &= I_x + I_y = J = \frac{\pi}{32}(D^4 - d^4) \end{aligned}$$



### D.3 Section elliptique

Pour une section elliptique de demi-axes ou rayons principaux  $R_1$  ( $x$ ) et  $R_2$  ( $y$ ) ( $R_1 < R_2$ ) :

$$\begin{aligned} P &\approx \text{sqrt}2\pi\sqrt{(R_1)^2 + (R_2)^2} \\ A &= \pi R_1 R_2 \\ I_x &= \frac{\pi}{4} R_1 (R_2)^3 \\ I_y &= \frac{\pi}{4} (R_1)^3 R_2 \\ I_o &= I_x + I_y = \frac{\pi}{4} R_1 R_2 \left( (R_1)^2 + (R_2)^2 \right) \\ J &= \pi \frac{(R_1)^3 (R_2)^3}{(R_1)^2 + (R_2)^2} \end{aligned}$$



L'excentricité  $e$  de l'ellipse vaut  $e = \sqrt{1 - \frac{(R_2)^2}{(R_1)^2}}$ , strictement comprise entre 0 et 1.

### D.4 Section carrée

Pour une section carrée de côté  $2R$  :

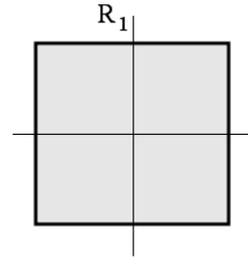
$$P = 8R$$

$$A = 4R^2$$

$$I_x = I_y = \frac{4R^4}{3}$$

$$I_o = I_x + I_y = \frac{8R^4}{3}$$

$$J = 24 \frac{R^4}{\pi^2}$$



### D.5 Section rectangulaire

Pour une section rectangulaire de base  $2R_1b$  et de hauteur  $2R_2h$  :

$$P = 4(R_1 + R_2)$$

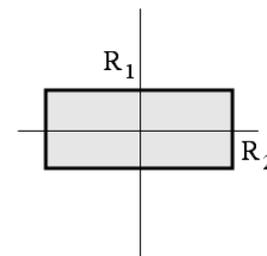
$$A = 4R_1R_2$$

$$I_x = \frac{4R_1(R_2)^3}{3}$$

$$I_y = \frac{4(R_1)^3R_2}{3}$$

$$I_o = I_x + I_y = \frac{4R_1R_2}{3}((R_1)^2 + (R_2)^2)$$

$$J = \frac{24(R_1)^3(R_2)^3}{\pi^2((R_1)^2 + (R_2)^2)}$$



### D.6 Section en double T

Pour une section en T, on pose  $b = 2R_2^1$ ,  $h = 2R_1^1$ ,  $t = 2(R_1^1 - R_1^2)$  et  $w = 2(R_2^1 - R_2^2)$  :

$$P = 2h + 4b - 2w$$

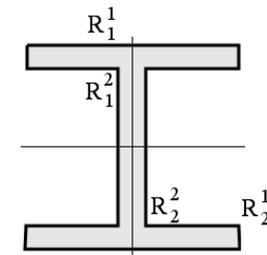
$$A = 2bt + wd$$

$$I_x = \frac{bh^3}{12} - \frac{(b-w)(h-2t)}{12}$$

$$I_y = \frac{b^3t}{6} + \frac{w^3(h-2t)}{12}$$

$$I_o = I_x + I_y$$

$$J = \frac{A^4}{4\pi^2 I_o}$$



## E EQUATION DE LA CHAÎNETTE ET AUTRES RÉSULTATS RELATIFS

Nous cherchons l'équation décrite par un fil maintenu à ses extrémités, ainsi que l'expression de la normale en fonction de l'abscisse.

### E.1 Bilan des forces

Raisonnons sur une portion locale du fil. Soit  $H$  la composante horizontale de la force interne exercée d'un côté de la portion du fil, et  $V$  sa composante verticale;  $H + dH$  et  $V + dV$  correspondent aux forces de l'autre côté de la portion du fil.  $P$  désigne son poids. Le bilan des forces donne, en projetant sur chacun des axes :

– sur l'axe des abscisses :

$$-H + (H + dH) = 0$$

– sur l'axe des ordonnées :

$$-V + (V + dV) - P = 0$$

Nous en déduisons que la composante horizontale est constante tout le long du fil. La pente de la courbe est égale à  $y' = \frac{V}{H}$ .

Différencions cette égalité :  $dV = Hy'' dy'$ . De plus, le poids  $P$  est donné par  $P = \mu g dl$ , nous obtenons :  $\mu g dl = Hy'' dy'$ .

D'après Pythagore,  $dl = \sqrt{1 + y'^2} dx$ . L'ordonnée de la chaînette est donc solution de l'équation suivante :

$$\frac{\mu g}{H} dx = \frac{y'' dy'}{\sqrt{1 + y'^2}} \quad (.1)$$

Intégrons le résultat obtenu :

$$\frac{\mu}{H} \int_0^x dx = \int_0^y \frac{y'' dy'}{\sqrt{1 + y'^2}}$$

$$\frac{\mu x}{H} = \operatorname{argsh} y'$$

$$y' = \operatorname{sh} \frac{\mu g x}{H}$$

Réintégrons :

$$y = \int_0^x \operatorname{sh} \left( \frac{\mu g t}{H} \right) dt$$

$$y = \left[ \frac{H}{\mu g} \operatorname{ch} \left( \frac{\mu g x}{H} \right) \right]_0^x$$

L'équation de la chaînette est donc :

$$y = \frac{H}{\mu g} \left( \operatorname{ch} \left( \frac{\mu g x}{H} \right) - 1 \right) \quad (.2)$$

Cependant, nous ne connaissons pas encore la valeur de la composante horizontale de la force de traction  $H$ . Celle-ci s'obtient à partir de l'expression de la longueur  $l$  de la chaînette.

## E.2 Longueur de la chaînette

La longueur  $l$  de la chaînette est l'intégrale de l'élément de longueur sur toute la chaînette :

$$\begin{aligned}
 l &= \int_{-\frac{l}{2}}^{\frac{l}{2}} dl \\
 l &= \int_{-\frac{l}{2}}^{\frac{l}{2}} \sqrt{1 + y'^2} dx \\
 l &= \int_{-\frac{l}{2}}^{\frac{l}{2}} \sqrt{1 + \text{sh}^2\left(\frac{\mu g x}{H}\right)} dx \\
 l &= \int_{-\frac{l}{2}}^{\frac{l}{2}} \text{ch}\left(\frac{\mu g x}{H}\right) dx \\
 l &= \left[ \frac{H}{\mu g} \text{sh}\left(\frac{\mu g x}{H}\right) \right]_{-\frac{l}{2}}^{\frac{l}{2}}
 \end{aligned}$$

La longueur  $l$  de la chaînette s'exprime donc :

$$l = \frac{2H}{\mu g} \text{sh}\left(\frac{\mu g L}{2H}\right) \quad (.3)$$

Pour obtenir  $H$ , il faut résoudre cette équation elliptique. On se contente d'une approximation numérique. Notons que  $L = \frac{2H}{\mu g} \text{argsh}\left(\frac{\mu g l}{2H}\right)$ .

## E.3 Flèche

La flèche  $h$ , amplitude maximale de déformation, est égale à  $y\left(\frac{l}{2}\right)$  :

$$h = \frac{H}{\mu g} \left( \text{ch}\left(\frac{\mu g L}{2H}\right) - 1 \right) \quad (.4)$$

## F LES OBJETS UNIDIMENSIONNELS DANS NOTRE ENVIRONNEMENT

Si les objets 1D sont à foison dans notre environnement, sous de multiples formes, les disciplines attenantes le sont aussi : Animation, architecture, biologie, chirurgie, mécanique, transports... Afin répertorier finement les modèles existants, nous avons effectué un état des lieux exhaustif des différentes problématiques existantes, liées aux objets déformables 1D dans notre environnement. Il se veut aussi large que possible. En effet, il était inutile de développer un outil spécifique s'il en existait déjà un plus général qui convenait à nos attentes. Le classement utilisé dans cet annexe est assez subjectif : il place d'un côté l'étude des matériaux et des mécanismes, qui cherche à améliorer la conception d'outils pour l'homme ; de l'autre les sciences du vivant et l'animation, qui s'attachent plus à comprendre et à reproduire des phénomènes naturels.

### F.1 Matériaux et mécanismes

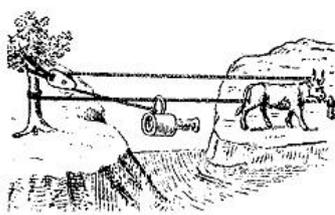
Si l'usage des cordes est d'abord un moyen pour l'homme de se déplacer à part entière, il devient vite partie intégrante de mécanismes de plus en plus ambitieux. D'abord comme élément mobile de véhicules tels que le téléphérique et le funiculaire, puis comme élément de suspension, associé à des poutres métalliques puis bétonnées, pour donner naissance aujourd'hui à de fantastiques ponts et autres ouvrages architecturaux.

#### Transport

Bien avant notre ère, se déplacer ou transporter des charges sur des reliefs abrupts n'était guère facile : les obstacles étaient nombreux, la neige y était fréquente et les moyens d'alors peu efficaces. Pourtant les anciens surent assez vite traverser des ravins grâce à des cordes sur lesquelles glissaient des nacelles (figure .1(a)). Ces systèmes étaient également utilisés pour véhiculer des objets d'une rive à une autre d'un cours d'eau (figure .1(b)). C'était une préfiguration de ce que l'on a appelé, par la suite, des téléphériques. Les premiers documents que l'on connaisse à ce sujet sont, pour l'essentiel, une estampe japonaise de 1250, suivie d'un texte de Konrad Kyner de 1405, puis de la gravure accompagnant un texte de Faustus Verantius de 1616 qui représente le premier téléphérique qui soit d'une conception correcte. C'est en 1644 que von Harlingen conçut et construisit à Dantzig un premier transporteur monocâble à matériaux (figure .1(c)).



(a) Traversée de torrents



(b) Transport de charge.



(c) Premier téléphérique répertorié.

**Fig. .1** – Premiers systèmes de transports à l'aide de cordes. Le câble ne fera son apparition que bien plus tard.

L'emploi de cordes en fibres végétales limitait considérablement les performances des appareils. Les plus anciens câbles métalliques connus ont été retrouvés à Pompéi et sont composés de fils de bronze. Au quinzième siècle, Léonard de Vinci mentionne l'existence de câbles d'acier et de bronze. Il semble que le premier vrai câble métallique ait été réalisé en 1834 par le suédois de Klansthal, bien que, peu avant, en 1825, l'autrichien Pukinje dise en avoir utilisé un pour un funiculaire. Grâce au câble métallique, les chemins de fer de

montagne, en devenant funiculaires, purent échapper aux lourdes contraintes de la crémaillère et des locomotives en adoptant la traction par câble à partir de l'une des stations. Les virages restaient possibles grâce à des poulies déviant le câble. Les téléphériques bénéficièrent également du progrès apporté par le câble : les installations n'étaient désormais plus des agencements de fortune avec cordes et nacelles, mais des structures solides et plus longues bien que, jusque vers 1910, on ne savait pas encore fabriquer des câbles de plus de 1000 mètres.

Moyen de transport similaire, le tramway à traction par câble (en anglais Cable car on rail) diffère des tramways classiques, car il n'y a pas de système de propulsion à bord. Un moteur fixe assure le mouvement du câble sous la chaussée, et le véhicule s'y agrippe pour se déplacer. Il ressemble en cela au principe du funiculaire, mais en plus complexe, car les trajets parcourus suivent la voie publique et peuvent avoir plusieurs montées et descentes successives. De plus, la cabine n'est pas solidaire du câble. Elle s'y agrippe par une grande pince au démarrage, le relâche avant l'arrêt suivant et utilise des freins classiques. L'exemple le plus célèbre et pérenne est le cable car de San Francisco en Californie, qui a été conçu en 1873 par Andrew Smith Hallidie. Il a été plusieurs fois rénové et fonctionne encore de nos jours, fournissant à la ville une de ses caractéristiques touristiques (figure .2(b)). Hong Kong possède une ligne du même type, le Peak Tram qui mène du centre ville au Victoria Peak, 396 m plus haut. Plusieurs dizaines de lignes de ce type ont été créées à la fin du XIXe siècle pour gravir les pentes inaccessibles aux tramways classiques. La plupart ont été fermées avant la Seconde Guerre mondiale, pour être souvent remplacés par des autobus. En France, Paris a été équipée d'une ligne comparable, le Tramway funiculaire de Belleville (.2(a)).

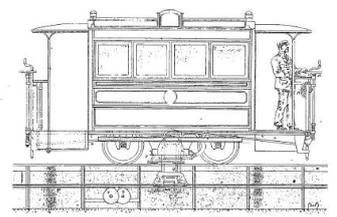


Fig. 2. — *Élévation simple d'une voiture funiculaire.* — 12. Levier du grip. — 13. Levier de commande du frein à patin. — 14. Levier de commande du frein à patin. — 15. Frein à patin. — 16. Manivelle du grip. — 17. Manivelle du frein. — 18. Gâche. — 19. Poulie du câble.



(a) Le funiculaire de Belleville à Paris. (b) Le tramway de San Francisco, encore en service.

**Fig. .2** – Deux tramways à traction par câble.

Plus récemment, depuis le début du XIXe siècle, l'utilisation de véhicules sous-marins contrôlés à distance permet par exemple la recherche d'objets sous la mer ou l'explorations de ressources dans les profondeurs sous-marines. Ces engins à câble dragués sont alimentés en énergie et communiquent via des câbles par leur bateau mère. La longueur considérable du câble qui est entraîné affecte de manière sensible le comportement de l'appareillage. Il est donc nécessaire de comprendre son comportement mécanique. Ce problème est notamment étudié par Weiss *et al.* (Weio2). C'est l'émergence des fibres optiques et du numérique qui a permis dans les années 1990 de dominer les interventions dans l'offshore pétrolier jusqu'à des fonds atteignant 3000 m. Et les années 2000 voient l'arrivée à maturité des engins inhabités civils sans câble, dits engins autonomes, qui font suite aux précurseurs des années 1980.

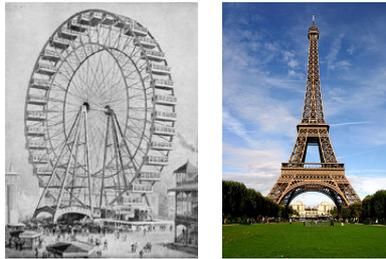
Si les câbles sont encore largement utilisés pour tracter des véhicules dans les airs, sur terre ou dans la mer, ils ont rapidement laissé place à des structures plus solides pour la traversée de cours d'eau ou de fossés par des véhicules autonomes.

## Architecture

Les *ponts* ont été longtemps le fer de lance de l'architecture, illustrant de manières remarquables les avancées technologiques, dont a profité ensuite le reste du secteur du bâtiment. Les ponts mettent en jeu essentiellement des éléments 1D : cordes, poutres en bois, en métaux, puis en béton armé, câbles, etc. C'est pourquoi il reflètent particulièrement bien les progrès accomplis avec notamment le développement de théories sur les objets monodimensionnels, comme celle des poutres.

L'origine des ponts suspendus, sous la forme de passerelles accrochées à des lianes (figure .4(a)), est très lointaine. L'usage du bois dans les ponts date de l'Antiquité, mais les édifices ne résistaient pas longtemps aux intempéries. Les ponts en pierre sont ensuite apparus en Asie puis chez les romains. Ce n'est qu'au XVIIIème siècle, avec les barres et les fils en fer ou en acier, que naissent les premiers ponts suspendus. Les méthodes de construction ainsi que le choix des matériaux utilisés sont alors très limités.

Cependant, des techniques de plus en plus développées ont permis aux ingénieurs de fabriquer des édifices de plus en plus fiables. C'est en effet à la fin du XIXème siècle que la *théorie des poutres*, une branche de la mécanique, a fait son apparition dans la construction. La science et l'industrie étaient auparavant vu comme des domaines très distincts, et l'on doutait sérieusement que les mathématiques et la physique puissent apporter des applications pratiques exploitables. Ce sont la Tour Eiffel et la grand-roue de Ferris ont démontré la validité de cette théorie à une grande échelle (figures .3(b) et .3(a)). La forme de la tour Eiffel, construite à la fin du XIXème siècle, a été calculée pour résister aux vents.



(a) La grand-roue de Ferris. (b) La tour Eiffel.

Fig. .3 – Deux édifices remarquables, construits à l'aide de la théorie des poutres.

Parmi les structures à base de poutres et de câbles, les ponts constituent de remarquables applications de la théorie des poutres. Les *ponts suspendus* sont constitués d'un tablier supporté par des câbles métalliques prenant appui sur deux pylônes et accrochés sur chaque rive à des massifs d'ancrage ; le tablier est également relié aux deux grands câbles porteurs par des suspentes (câbles verticaux). Le plus connu est le pont du Golden Gate, qui traverse la bouche de la Baie de San Francisco sur l'Océan Pacifique (figure .4(c)). Achievé en 1937, il a été jusqu'en 1964 le pont suspendu le plus long du monde. Les ponts métalliques apparaissent également, comme le Viaduc de Garabit de Gustave Eiffel, uniquement constitué de poutres (.4(b)). L'apparition du béton armé puis du béton précontraint au XXIème siècle rend les ingénieurs de plus en plus audacieux. Ces types de ponts se sont alors effacés au profit d'une autre technique, plus légère et plus audacieuse : le pont à haubans. Son tablier est cette fois supporté en plusieurs points de la travée par des câbles d'acier dont l'autre extrémité est accrochée à un pylône. Les ponts à haubans, sont en quelque sorte une sous catégorie des ponts suspendus. Ils ne tiennent pas, comme les ponts suspendus, grâce à deux câbles principaux, mais grâce à de nombreux câbles obliques partant d'un pilier supportant le tablier. Le plus spectaculaire est certainement le Pont de Normandie .4(d). C'est l'un des plus longs ponts à haubans du monde : 856m de portée centrale et 2143,21m de longueur. Ses deux pylônes font 214 mètres, soit la hauteur de la Tour Montparnasse à Paris (56 étages). Résultat de six ans de travaux, c'est un pro-

duit de haute technologie par ses méthodes de construction et ses matériaux, sophistiqué jusque dans la forme de son tablier en aile d'avion, qui lui permet de défier les vents de plus de 250km/h.



(a) Un pont primitif. (b) Le viaduc de Garabit. (c) Le Golden Gate. (d) Le pont de Normandie.

**Fig. 4** – Les ponts à travers les âges. Notez la différence de conception entre le pont du Golden Gate et le pont de Normandie.

Les ponts mettent en jeu des forces phénoménales. Leur étude nécessite aujourd'hui des calculs adaptés et complexes, que l'on peut par exemple trouver dans (Cou80a). Des aspects dynamiques peuvent également intervenir dans les édifices. Inauguré en 1940, le pont de Tacoma s'écroula six mois plus tard de manière spectaculaire, à cause d'un phénomène alors mal connu des concepteurs d'avant-guerre, la résonance. Les oscillations de l'ouvrage initiées par les rafales de vent finirent par s'auto-entretenir et s'amplifier (figure .5). Depuis quelques années, les progrès mathématiques et informatiques permettent de modéliser les phénomènes et ainsi de les prévenir. Des laboratoires se sont même spécialisés dans ces recherches.



**Fig. 5** – Les oscillations du pont de Tacoma qui ont provoqué sa destruction en 1940.

## Industrie

De taille moindre, les mécanismes comportant des éléments 1D font l'objet de tout autant d'attention. Les bras souples, par exemple, sont des systèmes de corps flexibles reliés par des articulations permettant divers types de mouvements relatifs, d'amplitudes généralement limitées par des butées mécaniques. Certains sont composés d'éléments rigides considérés comme indéformables, dont le mouvement s'effectue au niveau de liaisons éventuellement motorisées. D'autres sont plus flexibles et nécessitent d'être étudiés comme des structures déformables jointes (Cha98). Ces types de systèmes interviennent dans l'étude des bras manipulateurs de robot, les pales d'hélicoptère (Kun91), les satellites, les bras de navette spatiale (Nau05), etc. (voir figure .6). Des chercheurs de Nantes désirent quant à eux reproduire le comportement des poissons. En effet, comparées à nos réalisations technologiques, leurs performances font rêver : accélération pouvant atteindre jusqu'à 20 fois la gravité, vitesse excédant les 70 km/h, extraordinaire manoeuvrabilité : virage à 180° sans ralentir et sur des rayons de courbure de l'ordre du dixième de leur longueur, tandis que les véhicules actuels doivent ralentir de moitié et prendre des rayons de courbures de l'ordre de 10 fois leur longueur. Ces chiffres motivent à eux seuls les efforts actuels pour comprendre et reproduire les solutions des poissons sur nos systèmes

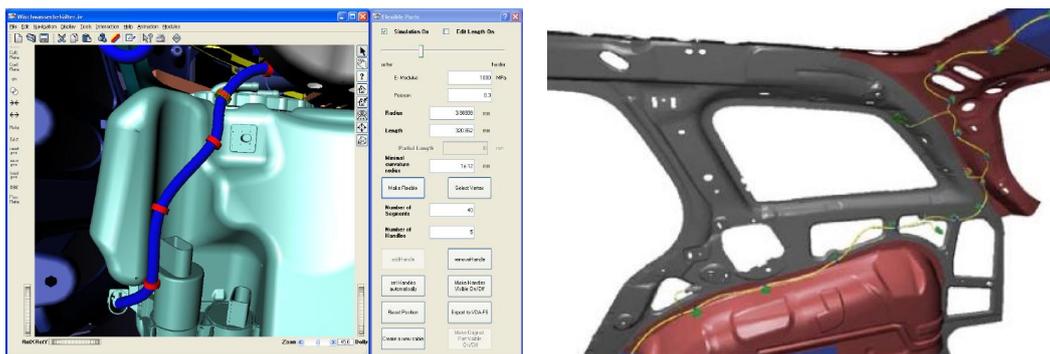


Fig. .6 – Liaisons motorisées d'objets de type poutre.

robotiques. Le caractère continu des poissons constitue la difficulté essentielle de la recherche dans ce domaine. C'est l'objet du projet Robea (Rob) que réaliser un prototype de robot anguille. Cet enjeu est à mettre en rapport avec les résultats récents de physiologistes (LAR02) travaillant sur l'usage des muscles dans la nage des poissons : le corps d'un poisson fléchit au cours de la nage à la manière d'une poutre homogène continue, pour toutes les espèces à l'exception du thon. Ce comportement simple est surprenant car la structure sous jacente des tendons, celle des muscles et leur comportement sont complexes.

Un autre domaine d'application important est l'industrie automobile où l'on cherche à mieux modéliser le comportement des composants des suspensions et des boîtes de vitesse, en prenant en compte leur flexibilité. Il en effet existe également de nombreux mécanismes dont le fonctionnement est fondé sur leur aptitude à se déformer : pinces, ressorts... (How01). Pendant les phases de conception, on s'intéresse en premier lieu à l'installation de câbles, tuyaux et fils électriques dans des structures : on cherche à vérifier si les composants peuvent être assemblés facilement et si certains n'entravent pas les autres. En deuxième lieu, les industriels travaillent sur la recherche de chemins pour passer les câbles au travers de structures complexes. Une modélisation informatique permet souvent d'éviter de passer par le processus long et coûteux de construction et de tests de prototypes. Pour les objets rigides, les simulations apportent des réponses satisfaisantes. Malheureusement, en ce qui concerne les objets 1D flexibles, il en est tout autre chose. Faute de simulation valide, les concepteurs ont toujours recours à la méthode traditionnelle, qui consiste à produire des modèles préliminaires pour déterminer et tester des solutions. Un besoin excessif en prototypes ralentit la phase de conception, freine la créativité et se révèle fort coûteux. C'est pourquoi un réel besoin de simuler leur comportement mécanique est ressenti, particulièrement dans les secteurs automobiles et aéronautiques. De nombreux travaux ont été publiés ces deux dernières années dans ce sens. Ils se concentrent sur deux aspects principaux de la simulation : des déformations convaincantes et la gestion des contacts avec l'environnement ( (GS06), figure .7(a) et (GHK<sup>+</sup>06), .7(b)), ainsi que la recherche de chemins au travers de structures complexes ( (GLM06; GSR<sup>+</sup>07; KGL07), figure .8).

Dans le domaine du textile, afin de limiter le recours à la fabrication de prototypes et donc d'augmenter sa réactivité par rapport aux évolutions du marché, l'industrie est également très demandeuse d'outils de conception assistée par ordinateur. Ces outils font particulièrement défaut dans le domaine de la maille, de par la complexité du processus du tricot. En collaboration avec l'Institut Français du Textile et de l'Habillement, le LERI a développé des algorithmes et des modèles adaptés à la simulation des mailles d'un point de vue géométrique (trajectoires et entrelacements des fils) et d'un point de vue mécanique (comportement élastique) (RNG99; Nou99; NR01). Dernièrement, Chalencon-Piotin *et al.* (CPNNR06) ont proposé un modèle de simulation destiné au concepteur de textiles tricotés, qui tient compte de la relaxation des efforts de tension après tricot. Groller *et al.* (GRS95) se sont intéressés essentiellement à la topologie et au rendu, la mécanique n'intervenant pas.



(a) Câble fixé par des clips dans Veo, plate-forme de réalité virtuelle, ©DaimlerChrysler. (GS06) (b) Branchement d'un câble en environnement de production, ©Adam Opel AG / fleXilution, 2005.

Fig. 7 – Simulatoin de câble pour l'industrie automobile.

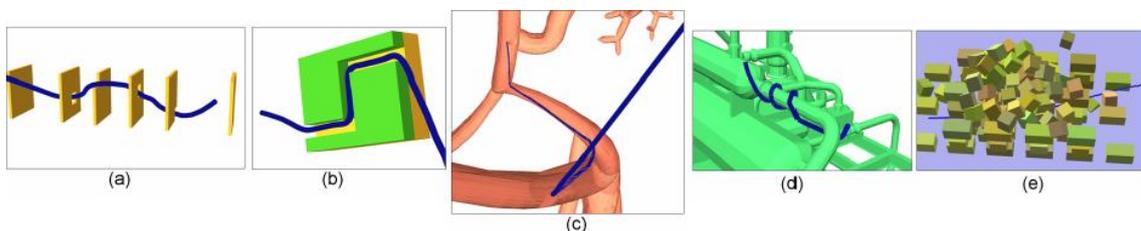


Fig. 8 – (GSR<sup>+</sup>07) : Recherche de chemin à travers des murs (a), dans un tunnel (b), d'un cathéter dans des vaisseaux (c), autour de tuyaux (d), dans des débris (e)

**La théorie des cordes**

La *théorie des cordes* est l'une des voies envisagées depuis le siècle dernier pour régler une des questions majeures de la physique théorique : fournir une description de la gravité quantique c'est-à-dire l'unification de la mécanique quantique (inévitable pour décrire la physique aux petites échelles) et de la théorie de la relativité générale (nécessaire pour décrire la gravitation de manière relativiste). Elle repose sur deux hypothèses assez révolutionnaires. Première hypothèse, *un élément filiforme (une corde) serait l'élément constituant la matière plutôt que des particules* (voir figure .9. Ce que nous percevons comme des particules de caractéristiques distinctes (masse, charge électrique, etc) ne seraient que des cordes vibrant différemment. Elles posséderaient diverses propriétés, leur permettant d'adopter différentes formes, en s'étirant, se tordant. Elles peuvent aussi se diviser, se fusionner à d'autres cordes et interagir avec l'espace dans lequel elles se mouvent. Deuxième hypothèse, *l'univers contiendrait plus de trois dimensions spatiales*. Certains chercheurs avancent l'hypothèse que les cordes vibreraient dans un univers à six ou onze dimensions.

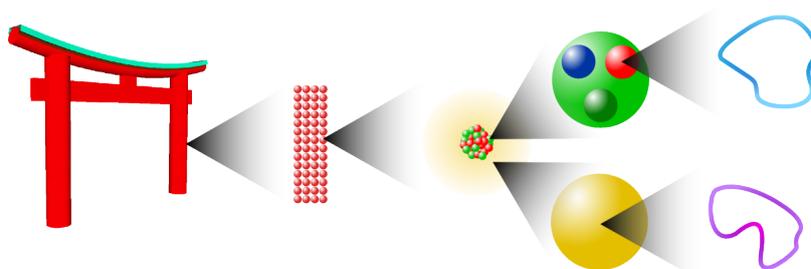


Fig. 9 – Théorie des cordes. Les niveaux de grossissements : monde macroscopique, monde moléculaire, monde atomique, monde subatomique, monde des cordes.

La théorie des cordes a obtenu des premiers résultats partiels très prometteurs. Dans le cadre de la thermodynamique des trous noirs elle permet de reproduire la formule de Be-

kenstein et Hawking pour l'entropie des trous noirs. Elle possède également une richesse mathématique remarquable : en particulier elle a permis de découvrir la symétrie miroir en géométrie. Toutefois la théorie des cordes reste incomplète. D'une part, une multitude de solutions aux équations de la théorie des cordes existe, ce qui pose un problème de sélection de notre univers et, d'autre part, même si beaucoup de modèles voisins ont pu être obtenus, aucun d'entre eux ne permet de rendre compte précisément du modèle standard de la physique des particules. Cette théorie, bien que prétendant pouvoir unifier toutes les théories, est bien trop abstraite et complexe pour être utilisée à des fins de simulation d'objets 1D dans un monde virtuel. Elle n'est ici mentionnée que pour démontrer l'importance des éléments 1D dans notre univers.

## F.2 Etude et animation du vivant

La modélisation des déformations des objets 1D peut s'avérer bien plus importante que l'on pourrait le croire dans l'étude du vivant. Elle touche l'ADN, support de l'information génétique : son comportement dépend de la mécanique des molécules qui le compose. Le fait de pouvoir prédire le comportement d'objets déformables 1D tels que les vaisseaux sanguins, les intestins ou les outils opératoires comme les cathéters ou les fils de chirurgie, permet d'améliorer grandement la réussite et la qualité des opérations chirurgicales. Dans un cadre plus cosmétique, le comportement de la chevelure révèle un fort potentiel financier. Enfin, l'homme satisfait sa curiosité naturelle sur quelques phénomènes tels que la rupture de spaghettis ou la pousse de lierre...

### L'A.D.N.

L'acide désoxyribonucléique (souvent abrégé en ADN) est une molécule que l'on retrouve dans tous les organismes vivants. On dit que l'ADN est le support de l'hérédité car il constitue le génome des êtres vivants et se transmet en totalité ou en partie lors des processus de reproduction. L'ADN contient donc toutes les informations susceptibles de créer et de faire vivre un organisme. Si le contenu de l'ADN humaine était mise sous forme d'une encyclopédie, il faudrait à peu près 500 volumes de 800 pages chacun. C'est en fait une succession de motifs appelés nucléotides, qui contiennent des phosphates, des sucres (désoxyribose) et des bases azotées. Ces bases sont au nombre de quatre : l'adénine (notée A), la thymine (notée T), la cytosine (notée C) et la guanine (notée G). Les séquences permettent à l'ADN de porter une information génétique sous forme d'un message codé.

Ces dernières années, un nombre croissant d'expériences et d'études théoriques ont été menées sur la structure, la dynamique et autres propriétés physico-chimiques de l'ADN. La simulation est une méthode qui permet de comprendre les détails de son mécanisme, quelque chose qui est très difficile, voire impossible à réaliser de manière expérimentale.

L'ADN est composé de deux brins formant une double hélice [.10](#). Cette structure fut découverte en 1953 par James Dewey Watson et Francis Crick ([WC53](#)) grâce aux travaux de Rosalind Franklin. Sa forme est justifiée par l'existence de nombreuses interactions dans la molécule. Une interaction existe tout d'abord au sein même d'une simple chaîne, ce qui va avoir pour conséquence un repliement en hélice. Un deuxième processus existe entre chaque hélice puisque les bases peuvent faire face à face et se stabiliser par des liaisons hydrogènes. Il existe une interaction à deux liaisons hydrogènes entre Adénine et Thymines et une interaction à trois liaisons hydrogènes entre Guanine et Cytosine. Deux à deux, les bases azotées sont donc associées par liaisons hydrogènes. Ceci assure la stabilité de l'ensemble.

Strick et Bensimon ([SB99](#)) ont décrit le processus d'enroulement mécanique de l'ADN et sa relaxation. Neukirch *et al.* ([NvdH02](#)) s'intéressent aux enroulements multiples de tiges de manière hélicoïdale. Shi *et al.* ([SMH01b](#); [SMH01a](#); [SMH03](#)) tiennent compte de

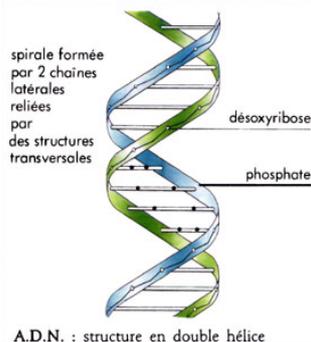


Fig. .10 – Structure en double hélice de l'ADN.

l'extension et du cisaillement, en plus de la flexion et de la torsion, dans leur modélisation de la double-hélice d'ADN.

Les déformations de l'ADN dépendent aussi des séquences : suivant les bases azotées qui les composent, celles-ci sont plus ou moins raides. Manning et al (MMK96) proposent un modèle qui tient compte de cet aspect. Pour modéliser la double hélice d'ADN, Moakher *et al.* (MM05) décrivent un modèle mécanique continu de deux tiges élastiques interagissant de manière élastique. Celles-ci sont décrites à l'aide de variables globales moyennées auxquelles sont rajoutées des petites perturbations locales. Lankas (Lano3) recense les résultats récents dans l'étude des déformations dépendant de la séquence d'ADN, en insistant sur les aspects informatiques de la modélisation. La simulation d'ADN reste néanmoins limitée à de petites longueurs de brins et rencontre des problèmes pour modéliser l'enroulement de la double hélice sur d'autres structures. Ruscio *et al.* (RO06) ont dû utiliser une grille de 128 processeurs pour obtenir la première simulation d'un brin d'ADN de 147 paires. C'est la longueur nécessaire pour former un nucléosome, unité fondamentale de l'ADN dans les cellules vivantes, qui mesure 500Å. Leur recherche a révélé que l'ADN est bien plus flexible qu'on ne le croyait jusqu'à présent.

L'étude mécanique des brins d'ADN, bien que prolifique, est généralement dédiée à une configuration particulière, voire spécifique aux propriétés de l'ADN, et donc souvent inadaptée à d'autres problèmes. Une modélisation plus générale est par conséquent nécessaire, mais peut-être pourrions-nous appliquer par la suite notre travail à leur simulation.

### Le corps humain

Le corps humain contient de nombreux organes développés en longueur : muscles et tendons, vaisseaux sanguins, intestins, cheveux, etc. Ils font l'objet d'une attention particulière essentiellement pour des raisons médicales : on essaie de comprendre le corps humain pour mieux le guérir, aussi s'intéresse-t-on également au comportement du fil de chirurgie.

**L'appareil moteur** Afin de comprendre le comportement du système nerveux de l'homme, l'importance d'établir un modèle biomécanique de l'appareil moteur a été mise en avant dès le début du siècle dernier. Les neurologues Lombard et Abbot l'expriment en 1907 par ces mots : "avant que nous puissions arriver à des conclusions fiables quant à l'action du système nerveux central sur les réflexes, nous devons obtenir une image satisfaisante du mécanisme des membres". Chen et Zeltzer (CZ92) ont introduit la modélisation biomécanique de muscles dans l'animation par ordinateur, en utilisant des éléments finis non-linéaires pour simuler les tissus musculaires. Plus récemment, des méthodes plus sophistiquées ont été proposées, comme l'utilisation de solides composés de B-splines (NTH01), d'éléments finis inversibles (ITF04) ou des fibres musculaires macroscopiques (PSW05). Néanmoins, la littérature est florissante en ce qui concerne des modèles spéci-

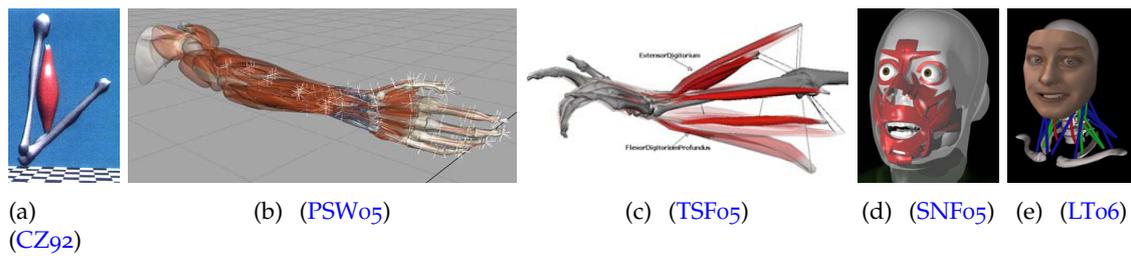


Fig. .11 – Diverses modélisations musculaires.

riques à certaines parties du corps, telles que les mains (TSF05; AHS03) le torse (ZCCD04), le visage (SNF05; LTW95) et le cou (LTo6) (figure .11).

Malgré ces nombreuses études, il n'existe pas encore de modèle qui décrit de manière satisfaisante le fonctionnement du système moteur dans l'espace. Les modèles sont soit trop simplistes (modèles particuliers), soit trop lourds et coûteux (éléments finis). Le modèle esquissé par Pai (PSW05) s'avère toutefois prometteur.

D'autres travaux de recherches ont pour but de promouvoir la chirurgie minimalement invasive. Elle est de plus en plus populaire, essentiellement grâce à des douleurs post-opératoires moindres, des infections moins fréquentes et un rétablissement plus rapide. Elle nécessite beaucoup d'entraînement avant d'être suffisamment maîtrisée pour être tentée sur un humain. Ce sont donc sur des animaux que les chirurgiens apprennent à ressentir les organes. Cela pose des problèmes évidents d'éthiques et d'argent. La réalité virtuelle est une solution envisageable. Elle permet aux chirurgiens de suivre et de manipuler des organes comme lors d'une vraie opération. Elle peut donc servir à l'apprentissage, à un test pré-opératoire ou même à une aide en cours d'opération.

**Les intestins** Les intestins, étant situés à l'extérieur du corps, sont une cible privilégiée pour la chirurgie minimalement invasive (CMI). Depuis quelques années déjà est pratiquée la colioscopie qui consiste à faire voyager une caméra miniature au travers du gros intestin et d'une partie de l'intestin grêle. Les polypes et les tumeurs naissantes peuvent ainsi être repérées et certaines de taille raisonnable retirées. Une résection du colon, ou colectomie peut toutefois être nécessaire et dans ce cas on pratique quelques incisions. Le comportement du cordon de la caméra et des outils, ainsi que des intestins que l'on peut approximer par des tubes constituent des objets 1D déformables évidents dont la simulation réaliste est ici cruciale.

Raghupathi *et al.* (RGF<sup>+</sup>04), ainsi que France *et al.* (FLA<sup>+</sup>05) s'y sont intéressés avec une modélisation simplifiée mais néanmoins prometteuse des intestins. Ces derniers montrent également que leur approche peut aussi s'appliquer aux trompes de Fallope.

**Le système sanguin** Depuis les vingt dernières années, les interventions médicales telles que l'angioplastie, la pose d'un stent ou l'administration de médicaments par cathéter ont nettement amélioré l'espérance de vie des patients atteints de maladies cardio-vasculaires ou neuro-vasculaires.

Un cathéter est un instrument médical consistant en un tuyau souple et fin (moins d'un millimètre d'épaisseur), qui est destiné à être placé en permanence dans une veine pour réaliser des injections ou perfusions prolongées ou répétées, ce qui améliore le confort du patient en évitant de le piquer à plusieurs reprises, surtout si l'accès à une veine est difficile, et en lui permettant de se mobiliser avec moins de précautions. Le cathéter a également d'autres utilisations, telles que l'angioplastie et la pose de stent.

L'angioplastie est une technique utilisée pour élargir sans chirurgie une artère rétrécie. Le principe de base de l'angioplastie consiste à positionner à l'intérieur de la partie rétrécie de l'artère un cathéter dont l'extrémité est équipée d'un ballon gonflable. Celui-ci

est gonflé et écarte la partie rétrécie et la paroi environnante de l'artère. Il ouvre l'artère rétrécie en déchirant et comprimant la plaque et en étirant légèrement la paroi de l'artère.

Un stent (voir figure .12(a)), du nom d'un dentiste anglais, Stent, est une endoprothèse. Un stent coronaire ou endoprothèse coronaire est un fin treillis métallique cylindrique que l'on place à l'intérieur des artères coronaires pour maintenir leur calibre au diamètre idéal. Ces endoprothèses sont serties sur un ballonnet qui, après avoir été positionné à l'endroit à traiter, est gonflé, donnant ainsi la forme correcte à l'endoprothèse coronaire. Le but du stent est de maintenir le diamètre obtenu par gonflement du ballonnet en empêchant le rétrécissement élastique qui inévitablement se produit lorsqu'on se contente de dilater avec un ballonnet seul lors d'une angioplastie. Le stent peut également servir à combler un anévrisme, dilatation localisée de la paroi d'une artère aboutissant à la formation d'une poche de taille variable.

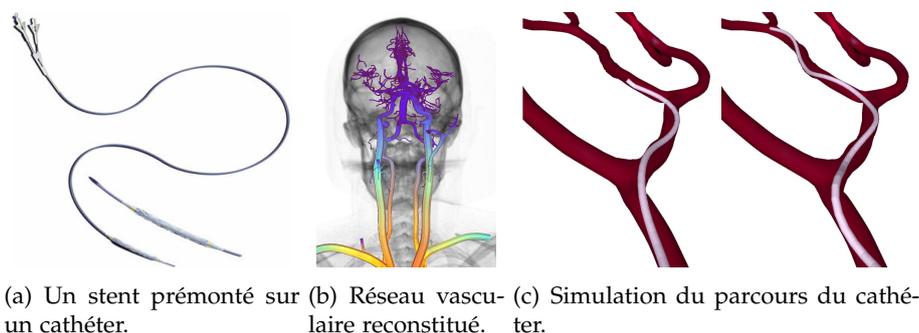


Fig. .12 – Vue d'ensemble de l'utilisation d'un stent ; .12(a) et .12(c) : (CDL<sup>+</sup>05).

Ces techniques demandent un important retour tactile et visuel et donc une longue période d'apprentissage classiquement sur des animaux puis des patients. L'utilisation d'une simulation informatique peut néanmoins constituer une alternative sérieuse. De plus les techniques de scan de l'appareil sanguin de l'homme sont aujourd'hui suffisamment efficaces pour reconstituer en 3D l'appareil sanguin (CDL<sup>+</sup>05; LRS07) (figure .12). Une simulation pertinente de l'opération avant l'intervention réelle est donc possible. Cotin *et al.* (CDL<sup>+</sup>05; LCDN06b) proposent un modèle réaliste de cathéter guidé dans le système sanguin qui gère de manière efficace le contact entre le cathéter et les vaisseaux, notamment au niveau des bifurcations (voir figure .12(c)). Gayle *et al.* (GSLM05) modélisent le déplacement d'un cathéter dans l'appareil sanguin, cette fois dans le but d'administrer un traitement médicamenteux directement dans les tumeurs du foie. D'autres problématiques entrent ici en jeu : la manipulation de cathéters dans de petits vaisseaux cause souvent des spasmes qui empêchent qu'un flux adéquat arrive jusqu'à la tumeur. Un autre problème peut survenir quand le diamètre du cathéter est trop proche de celui de vaisseau. Dans ce cas, le flux peut également être réduit ; des reflux peuvent aussi se produire vers d'autres artères.

**La suture de tissus** La suture de tissus présente également les mêmes enjeux que précédemment, à savoir entraîner les chirurgiens à moindre coût et apporter une aide avant et pendant l'opération. Le comportement le plus difficile à modéliser se révèle être celui des déformations du fil de suture et des contacts avec lui-même (noeuds) et son environnement (les tissus, les pinces pour le manipuler).

Certains modèles ne simulent pas le comportement du fil à proprement dit mais le remplacement par l'action d'une ou plusieurs forces sur le maillage ou de l'ajout de contraintes en position sur certains points du maillage. Le fil est alors une représentation purement géométrique, simple trait ou spline (BMLS01; WZM<sup>+</sup>01; LPD03).

Lenoir *et al.* (LMGC02), ainsi que Philips *et al.* (PLK02) à un moindre niveau, ont proposé des modèles visuellement assez convaincants, pour approcher la physique du fil par

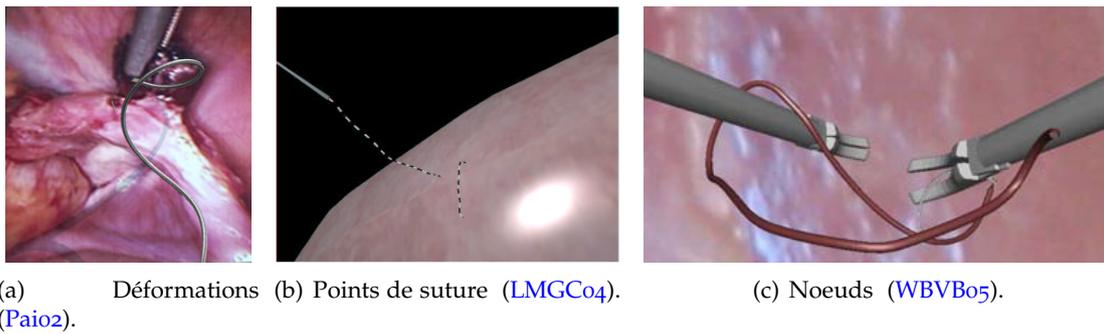


Fig. 13 – Trois problèmes de la modélisation du comportement du fil de chirurgie.

des splines animées par des ressorts. Lenoir *et al.* proposent également une méthode d'interaction du fil avec les tissus (LMGC04) (figure I.26(a)). Chez Simbionix, Shilon (Shio5) propose une simulation plus continue des déformations du fil, mais le modèle reste encore éloigné du comportement réel : ces modèles gèrent correctement l'extension, mais la flexion de manière approximative ; ils négligent la torsion. L'ajout de la torsion sous forme de ressorts par Wang *et al.* (WBD<sup>+</sup>05; WBVB05) ne suffit pas à rendre la manipulation réaliste et stable, bien que temps-réel (figure 13(c)). Un modèle physique plus poussé est donc nécessaire.

Pai (Paio2) a été le premier à proposer un modèle de fil de chirurgie mécaniquement valide, qui gère la flexion et la torsion (figure 13(a)). Ce modèle présente néanmoins l'inconvénient majeur de n'être facilement manipulable qu'aux extrémités. Il ne tient pas compte des effets d'inertie. Il est de plus inextensible, contrairement à un vrai fil de chirurgie, dont les déformations réelles s'avèrent finalement plus complexes puisqu'elles ne sont pas seulement élastiques (réversibles) mais plastiques (irréversibles) (SVH99).

Pour l'opération de la cataracte, simuler la suture d'un greffon revêt une importance supplémentaire. Chaque année, environ 3 000 interventions sont réalisées en France, alors que les besoins de la population sont estimés à environ 6 000. La *kératoplastie transfixiante* (BLDM99) consiste à extraire le cristallin de l'oeil, à glisser un implant ou lentille dans l'oeil, puis à recoudre un morceau de cornée par dessus 14(a). La qualité de la suture et de la vue post-opératoire du patient sont étroitement liés, puisqu'un noeud pas assez ou trop serré altérera le résultat final. Les noeuds évoluant au cours des mois suivant l'opération, le chirurgien peut être amené à les resserrer mais il ne dispose d'aucun moyen autre que son expérience pour prédire le résultat (Rop54; UL07). Depuis quelques années, la greffe de cornée a été remplacée par une incision périphérique de 2 mm, et la lentille est maintenant insérée repliée 14(c). Cette opération n'a à notre connaissance pas encore été simulée entièrement, malgré tous les bénéfices qu'elle pourrait apporter. Pour le moment, seuls la partie de découpe de la cornée et le procédé d'extraction par ultra-sons, appelé phaco-émulsification 14(b), ont été simulés (AGP<sup>+</sup>06).



(a) Cornée après kératoplastie transfixiante (BLDM99). (b) Phaco-émulsification. (c) Insertion de la lentille par incision.

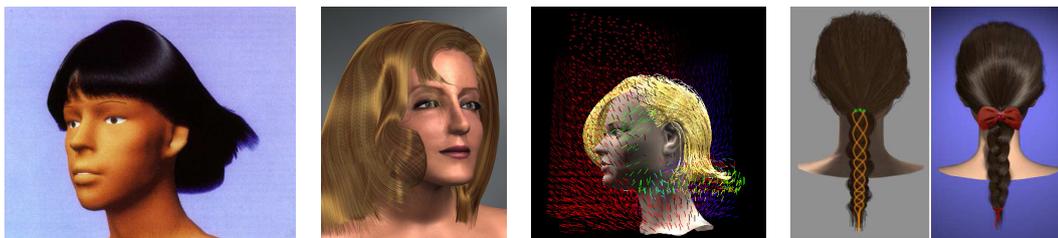
Fig. 14 – Techniques utilisées lors de l'opération de la cataracte

## Les cheveux

Les grands groupes industriels de cosmétique et de coiffure investissent des fonds importants dans l'étude des cheveux. L'idée est de créer un salon de coiffure virtuel dans lequel on pourrait reproduire la pousse, la forme, le mouillage, la coupe et le séchage des cheveux...

On s'est d'abord intéressé à simuler la forme au repos des cheveux sur le crâne, en les regardant individuellement, comme des brins ou des courbes en trois dimensions dans l'espace. Partant de ce constat, les chercheurs ont tenté d'expliquer comment les cheveux interagissent entre eux pour donner le volume d'une chevelure. Bien que de nombreuses pistes aient été explorées, modéliser une chevelure complète reste un problème ouvert, à cause de la complexité de la géométrie des cheveux et de leurs collisions. De nombreuses méthodes s'inspirent de modèles physiques plus ou moins sophistiqués : poutres linéaires (AUK92), fluides (HMT00), champs de vecteurs (Yu01) ou contraintes (CK05), qui ne tiennent pas vraiment compte des propriétés mécaniques du cheveu (voir figure .15).

Ce n'est que tout récemment que Bertails *et al.* (BAQ<sup>+</sup>05; WBC07) ont proposé des modèles visuellement convaincants, qui reproduisent des comportements tels que les boucles, à partir des propriétés physiques des cheveux.



(a) Poutres linéaires (AUK92). (b) Fluides (HMT00). (c) Champs de vecteurs (Yu01). (d) Contraintes (CCK05).

Fig. .15 – Différents modèles physiques utilisés pour la modélisation de chevelures.

Quant à l'animation de cheveux, plusieurs hypothèses sont généralement faites sur le comportement global des cheveux pour permettre une simulation efficace : les cheveux ne sont pas simulés seuls mais par mèches disjointes, (PCP01; KH01; BKCNO3; WLL<sup>+</sup>03; CCK05; BAC<sup>+</sup>06; VMT06), ou comme un milieu continu (HMT00; CJY02; BCNO3).

Si des modèles capables d'animer une chevelure dynamique en temps réel existent depuis quelques années (VMT06; WGL06; BCNO3; GZ02), ceux-ci restent relativement éloignés de la réalité mais permettent une animation visuellement plausible sans trop y regarder. Les animations convaincantes, traitant de manière correcte les déformations et les non-interpénétrations de cheveux, prennent encore jusqu'à quelques secondes par image pour une chevelure complète (BAC<sup>+</sup>06; CCK05). Grâce au développement des architectures parallèles et des cartes graphiques, on peut espérer obtenir une simulation de cheveux temps-réel d'ici quelques années.

Le salon de coiffure virtuel est une application directe de la simulation du comportement physique des cheveux. Pour réaliser une coiffure, on mouille généralement les cheveux : leur cohésion augmente, ils sont ainsi plus facile à couper. Mais leurs propriétés géométriques et mécaniques changent aussi : leur rayon augmente et leur raideur diminue, les rendant beaucoup plus déformable. Ward *et al.* ont proposé un modèle simple qui tient compte de ces spécificités (WGL04; WGL06), au travers d'une interface haptique 3D pour créer des coiffures stylées. Elle comporte le mouillage local des cheveux, la découpe par ciseaux et le séchage par sèche-cheveux. Notons que la coupe étant purement géométrique, la chevelure globale n'est pas affectée. Le modèle statique de cheveux de Bertails *et al.* (Bero6) permet d'obtenir un comportement plus convaincant lors des phases de séchage et de mouillage, puisqu'il intègre les paramètres physiques du cheveu et leurs variations mesurées expérimentalement. En outre, il peut également simuler la pousse des cheveux

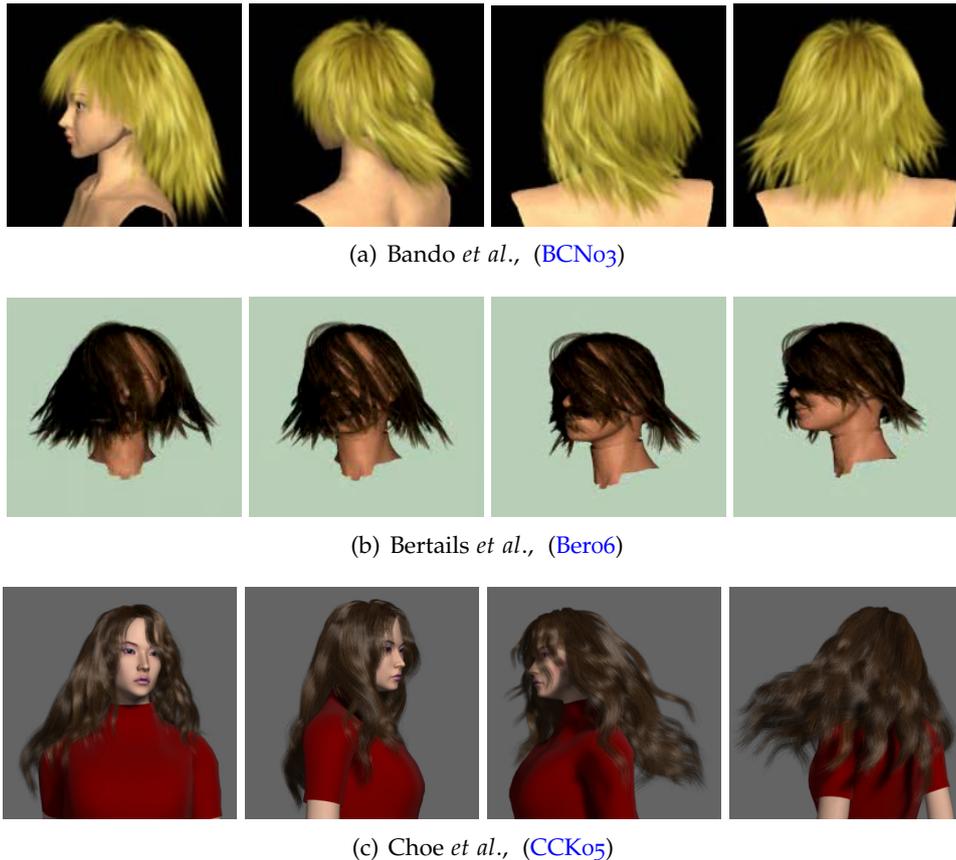


Fig. .16 – Plusieurs techniques réalistes d'animation de cheveux

et reproduire le phénomène de flambage : au-delà d'une certaine longueur et sous l'effet de la gravité, le cheveu passe d'une configuration plus ou moins rectiligne à une forme beaucoup plus courbée. Wither *et al.* (WBC07) proposent une interface à base de "coups de crayons" qui permet d'obtenir rapidement une chevelure virtuelle.

Une autre application intéressante pourrait être la pose virtuelle d'extensions. Des mèches de cheveux viennent s'ajouter à la chevelure pour obtenir de la longueur et / ou du volume. Les plus utilisés sont issues de chevelures de femmes hindoues, la tradition voulant qu'à leur mariage elles coupent leurs nattes en guise de sacrifice. La nature de leurs cheveux correspond à la majorité des femmes : ils sont relativement fins et ont une épaisseur qui varie entre 0.06 et 0.09 mm. Les mèches sont fixées, à environ 0.5 cm de la racine de vos cheveux, avec de la Kératine qui sèche puis durcit pour devenir quasi imperceptible au toucher. La durée des extensions varie de un à quatre mois. Le tarif d'une mèche étant aux alentours de 5 euros, le prix peut aller jusqu'à 1500 euros. Prévoir le résultat à l'aide d'une simulation permettrait de mieux choisir la répartition des mèches.

Pour plus d'information sur la simulation mécanique de cheveux et les recherches connexes, nous invitons le lecteur à consulter l'état de l'art de Ward *et al.* (WBK<sup>+</sup>07). La simulation dynamique de chevelures fait partie des éléments 1D qui font l'objet d'une attention spéciale en animation.

### Films d'animation

Le monde de l'animation dispose de moyens colossaux pour développer des techniques d'animation. Si dans un premier temps les éléments 1D étaient faits à la main à des moments clés puis interpolés (figure .17(a)), des techniques ont vite fait leur apparition pour alléger le travail fastidieux des animateurs. Il est en effet très difficile de réaliser une animation convaincante en essayant de reproduire soi-même les déformations dynamiques.

Sorti en 1997, *Toy Story* est le premier film entièrement réalisé en images de synthèse. Il



Fig. .17 – L'évolution de la modélisation des objets 1D dans l'animation.

comporte de nombreux éléments 1D : des cordes, des ressorts, etc. (figure .17(b)). Ceux-ci ont été animés à l'aide d'un modèle visuel, non fondé sur la mécanique, mis au point par Barzel (Bar97). Ce choix est justifié par le fait que la plupart du temps les animations sont caricaturées. L'anticipation et l'exagération des mouvements, défiant les lois de la physique, permettent de soutenir l'histoire et les personnages, et diriger l'attention du public.

Sur les écrans en 2001, le film d'animation *Final Fantasy : les créatures de l'esprit* a beaucoup fait parler de lui. Il prétend remplacer les humains par des acteurs virtuels, grâce à une animation et un rendu exceptionnels. Un point a fait l'objet d'une attention toute particulière : la chevelure d'Aki Ross .17(d), personnage principal, a monopolisé un cinquième du temps de calcul nécessaire à la réalisation du film. Tous les autres protagonistes ont le crâne rasé ou les cheveux très courts, quasiment inanimés. Même si sa chevelure est deux fois moins dense que celle d'un humain, ce sont tout de même 60.000 cheveux qui ont été animés individuellement ! Quelques années plus tard, SquareEnix proposera un autre film d'animation sans lien avec le précédent, *Final Fantasy : Advent Children* (2005). Les personnages ont cette fois quasiment tous une chevelure prolifique, mais l'animation se fait par mèches. Le résultat est toutefois bluffant.

*Ryan* (figure .17(f)) met en oeuvre une technique d'animation (CS06) qui, si elle n'est pas physique, permet d'éviter les interpénétrations d'objets complexes à base de courbes, tout en donnant un contrôle géométrique précis aux animateurs.

Pour certaines scènes, l'animation mécanique est indiscutablement le meilleur choix. Dans *Monster Inc.*, un des personnages principaux, la fourrure de Sully (voir figure .17(e)) est modélisée par 2.320.413 poils bleus, simulés individuellement pour obtenir une simu-



(a) Tomb Raider, ©Eidos, 1996. (b) Tomb Raider 2, ©Eidos, . (c) Tomb Raider 3, ©Eidos, . (d) Tomb Raider Legend, ©Eidos, 2006.

**Fig. .18** – L'évolution de la natte de Lara Croft : d'une texture à une chaîne articulée de plus en plus évoluée... mais peu réaliste !

lacion réaliste. Pixar a développé son propre moteur, *Fitz*, dans ce seul but. *Fitz* gère le mouvement mais garantit aussi que les poils ne vont pas avoir d'intersection entre eux.

Dans les *Indestructibles*, l'animation des cheveux mélange simulation physique et retouche manuelle laborieuse pour aboutir à des résultats probants quoique parfois inexacts, dus aux interpénétrations entre cheveux et autres parties du corps. La scène dans laquelle Violet, seul personnage aux cheveux longs du film, se fait consoler par sa mère qui lui rabat une mèche de cheveux derrière les oreilles, a été l'une des plus éprouvantes pour les réalisateurs.

Les techniques utilisées dans les films sont rarement divulguées. Hadap a cependant pu publier la méthode utilisée par les studios Dreamworks pour simuler les déformations approximées de la physique et les collisions d'objets 1D dans *Shrek 2* et *Madagascar* (Hado6).

Sortis cette année, *Ratatouille* et *Bee* élèvent encore le niveau de réalisme et la qualité des animations d'objets 1D.

Si les techniques de l'animation d'objets 1D, spécialement les cheveux, se sont considérablement améliorées depuis vingt ans, l'animation de cheveux reste un problème difficile et pas toujours traité convenablement. On peut déplorer par exemple la trop grande raideur des chevelures dans *Arthur et les Minimoys* .17(j) pour contourner le problème de la modélisation de leurs déformations. De plus, les animations sont loin d'être interactives : une image prend généralement plusieurs minutes de calculs sur des fermes de plusieurs centaines, voire plusieurs milliers de processeurs. C'est donc logiquement que nous nous tournons maintenant vers les jeux vidéos, où l'interactivité est inhérente au fonctionnement et à l'appréciation d'un jeu.

### Moteurs physiques de jeux vidéos

Un nombre croissant de jeux vidéos incorpore un moteur physique pour donner plus de vie à l'environnement et augmenter les interactions des personnages avec les objets qui le constituent. Seuls les objets rigides sont généralement simulés. Les objets déformables comme des cordes sont généralement constitués de chaînes articulées ou de systèmes masse-ressort. Ces modèles sont connus pour être très efficaces, mais n'offrent en contrepartie que des comportements simples ou approximatifs. Ainsi, au fil des épisodes de *Tomb Raider*, la modélisation de la natte de la très célèbre Lara Croft n'a que peu évolué... (figure .18). On commence cependant à voir apparaître dans les moteurs physiques des spécifications dédiées aux déformations 1D. Le moteur *CryEngine2* de Crytek propose de la "physique avancée de cordes" pour modéliser la flexion de la végétation balayée par le vent, la pluie, ou les personnages, ainsi que des ponts de cordes interactifs (figure .19(b)).

Si les jeux vidéos comportent des modélisations rapides d'objets 1D déformables, leur réalisme est loin d'être à la hauteur du vrai comportement physique d'une corde ou de cheveux. Certains chercheurs s'intéressent au contraire à expliquer puis à reproduire des phénomènes naturels complexes.



(a) Half-Life 2, un simple segment oscillant ©Valve, 2005.



(b) Crysis, ©Crytek, 2007.

**Fig. .19** – *L'évolution des techniques physiques dans les jeux vidéos*

### **Autres matériaux organiques**

Certains trouvent leur explication par l'étude de la mécanique des objets déformables 1D, notamment les plantes grimpantes et la rupture d'un spaghetti.

Les plantes grimpantes ont développé de multiples stratégies pour s'enrouler en hélice autour d'autres végétaux ou d'objets sans jamais supporter leur propre poids. Goriely *et al.* (GNo6) montrent, à l'aide de la mécanique, que la croissance verticale se fait grâce à des points et des régions de contact qui créent des tensions dans la tige pour son maintien, et qu'il y a un rayon du support au-delà duquel la plante grimpante ne peut plus progresser.

Dans un autre domaine, Audoly *et al.* (ANo5a; ANo5b) se sont intéressés à la rupture du spaghetti. Lorsque l'on brise un spaghetti en le pliant, on n'obtient généralement pas deux morceaux mais souvent trois, quatre ou plus. Leur étude montre que la rupture en un point entraîne un train d'onde en flexion qui provoque à son tour des cassures en augmentant localement la flexion.

# Index

- allongement relatif, 11
- analyse modale, 129
- B-Spline uniforme cubique, 61
- chaînette, 83
- cisaillement, 20
- coefficient de Poisson, 13
- coefficients de Lamé, 13
- collision, accélération
  - méthodes eulériennes, 48
  - méthodes lagrangiennes, 48
- compression, 20
- condition d'optimalité, 44
- condition de Courant, 45
- contrainte, 10
- contrainte de von Mises, 16
- contrainte lagrangienne
  - barycentrique, 104
  - cinématique, 50
  - d'abscisse fixe, 121
  - d'abscisse glissante, 101, 113
  - d'encastrement coplanaire relatif, 103
  - d'orientation, 95
  - de courbure, 96
  - de distance, 96
  - de distance relative, 103
  - de plan, 97
  - de point glissant orienté, 99
  - de position, 95
  - de positions relatives, 102
  - glissante, 98
  - holonome, 50
  - unilatérale, 49
  - à abscisse variable, 122
- contraintes bilatérales, 49
- coordonnées généralisées, 72
- couplage, 47, 105
- couple de torsion, 23
- courbe de contrainte-déformation, 12
- courbure, 68
- critère de la contrainte tangentielle maximale, 16
- critère de Tresca, 16
- critère de von Mises-Hencky, 16, 18
- critère du plus grand cisaillement, 16
- cylindre droit, 107
- cylindre généralisé, 70, 108
- d'axe, 97
- contrainte, 102
- déformation, 10
- différences finies, 35
- dynamique directe, 32
- décalage de plasticité, 80
- défaillance, 17
- déformation, 11
- déformation relative, 81
- déformation transverse, 25
- effort normal, 23
- effort tranchant, 23, 27
- équation raide, 46
- équations de Lagrange, 39, 72
- équilibre
  - dynamique, 75, 126
  - quasi-dynamique, 127
  - quasi-statique, 43, 75, 126
- fatigue, 17
- fibre moyenne, 20
- flexion, 20
- flexion pure, 26
- fréquence propre, 128
- gradient, 44
- grandes déformations, 11
- grands déplacements, 11, 71
- géométriquement exact, 37
- hypothèse de Bernoulli, 22
- hypothèse de Kirchhoff, 72
- hypothèse des milieux continus, 10
- Intel Math Kernel Library, 123
- interpolation
  - de Hermite, 59, 64
  - de Lagrange, 64
- ligne moyenne, 20
- limite d'élasticité, 17

- loi d'équilibre, 73
- loi de Hooke, 11, 14
- LU Bande, 123
- matériau
  - cassant, 17
  - ductile, 17
- matrice de densité d'inertie, 76
- matrice de viscosité dynamique, 93
- module d'Young, 13
- module de cisaillement, 13
- module de rigidité à la torsion, 27
- modèle d'Euler-Bernoulli, 72
- modèle de Cosserat, 34
- modèle de Rayleigh, 72
- moindre action, 72
- moment fléchissant, 23
- multiplicateur de Lagrange, 49, 50, 94
- méthode de Newton, 44
- méthode de stabilisation de Baumgarte, 39
- méthode des éléments finis, 36
- méthode du gradient, 44
- méthode du gradient conjugué, 44
- méthode quasi-Newton, 44
- méthodes analytiques, 49
- méthodes continues, 48
- méthodes discrètes, 47
- méthodes à pénalités, 49
- noeuds, 36
- non-linéarités, 11
- non-linéarités géométriques, 11
- non-linéarités matérielles, 11
- optimisation, 44
- petites déformations, 11, 71
- petites perturbations, 11
- petits déplacements, 11
- placement des contraintes, 119
- placement des points de contrôle, 119
- plan osculateur, 68
- plasticité parfaite, 17, 80
- point de rupture, 17
- poutre, 20
- poutre d'Euler-Bernoulli, 22
- poutre de Rayleigh, 22
- principe de Saint-Venant, 23
- principe de superposition, 12
- problème de Saint Venant, 27
- propriété de conservation, 47
- recuit simulé, 45
- représentation eulérienne, 10
- représentation lagrangienne, 10, 72
- repère de Frenet, 68
- ressort généralisé, 30
- schéma d'Euler explicite, 45
- schéma d'Euler implicite, 46
- schéma implicite linéarisé, 77
- schéma numérique, 36
- section, 20
- section constante, 20
- section variable, 20
- simulation dynamique, 43
- simulation quasi-statique, 43
- spline, 56
  - B-Spline, 60
  - cardinale, 59, 62
  - circulaire, 62
  - d'approximation, 58
  - d'interpolation, 58
  - de Catmull-Rom, 59
  - KB-Spline, 59
- théorie des poutres, 19
- théorème de Nyquist-Shannon, 128
- torseur de cohésion, 23
- torsion, 20
- torsion de Frenet, 68
- torsion géométrique, 68
- torsion matérielle, 68
- traction, 20
- transport parallèle, 69
- variable libre, 98
- vecteur de courbure binormal, 68
- vecteur normal de courbure, 68
- vecteur normal principal, 68
- vecteur tangent, 68
- Vertex Blending, 70
- violation, 94



**Titre** Splines Dynamiques Géométriquement Exactes :simulation haute performance et interaction

**Résumé** Quel est le point commun entre une poutre, un cheveu, un brin d'ADN et un câble ? Ce sont tous des objets unidimensionnels, qui font l'objet de recherches intensives dans des domaines scientifiques divers et variés que l'architecture, l'animation, la biologie ou l'industrie automobile. Les enjeux le sont aussi : robustesse, réalisme visuel, compréhension des mécanismes de la vie, efficacité de la conception. . . Mais tous font appel à la simulation des déformations de ces objets, avec tout ou partie de trois objectifs récurrents et souvent antagonistes : précision, facilité de manipulation et performance.

Dans cette thèse, nous proposons un modèle qui satisfait ces trois exigences, à partir d'une formulation géométriquement exacte de spline matérielle.

Dans un premier temps, nous assemblons des outils géométriques et mécaniques afin de traiter les déformations réversibles et irréversibles. Un effort important est porté sur la clarté du raisonnement, pour qu'il soit accessible à des non mécaniciens, ainsi que sur la validation du modèle.

Dans un deuxième temps, nous élaborons tout un jeu de contraintes destinées à manipuler intuitivement la spline matérielle. La modélisation de clips de fixation ou de faisceaux de câbles en sont des applications directes.

Dans un troisième temps, nous précisons comment obtenir une résolution performante, de complexité linéaire. Nous présentons enfin la simulation quasi-dynamique, qui concilie les avantages des équilibres quasi-statique et dynamique. Elle consiste à passer de l'un à l'autre automatiquement en cours de manipulation, à l'aide d'heuristiques portant sur le matériau, les contraintes et le temps de calcul.

**Mots-clés** spline, poutre, géométriquement exact, quasi-dynamique, multiplicateurs de Lagrange, réalité virtuelle, mécanique, informatique graphique

**Title** Geometrically Exact Dynamic Splines :high performance and interaction

**Abstract** What do a beam, a hair strand, a DNA molecule or a cable have in common ? They are considered as one dimensional objects, that are intensively studied in such fields as architecture, animation, biology or car industry. A wide range of objectives have to be performed :robustness, virtual realism, life comprehension and efficiency. . . However, all of them consider deformations of these objects, aiming at most of the three following goals : accuracy, easiness of handling and performance.

This thesis pretends to conciliate both of them, using a material spline formulation. Firstly, we use geometrical and mechanical tools to provide a geometrically exact model of splines. We focus on a clear reasoning and the validation of the model, which can be easily understood by non mechanical engineers. Secondly, we elaborate a set of constraints to offer an intuitive handling of material splines. Direct applications are fixing clip modeling or cable beams.

Thirdly, we explain how we get an efficient linear solving, then we propose the quasi-dynamic equilibrium, which conciliates the advantages of quasi-static and dynamic ones. It consists of switching between them, using heuristics depending on material, constraints and computation time.

**Keywords** spline, beam, geometrically exact, quasi-dynamic, Lagrangian multipliers, virtual reality, mechanics, computer graphics