Deux problèmes physiques pour la synthèse d'images

François Faure

▶ **To cite this version:**

François Faure. Deux problèmes physiques pour la synthèse d'images. Modélisation et simulation. Université Joseph-Fourier - Grenoble I, 1997. Français. NNT : . tel-00528658

HAL Id: tel-00528658
https://theses.hal.science/tel-00528658

Submitted on 22 Oct 2010

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

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

Thèse présentée par :

François Faure

pour obtenir le titre de :

Docteur de l'Université Joseph Fourier
spécialité Informatique

Arrêtés ministériels du 5 Juillet 1984 et du 30 Mars 1992

Deux problèmes physiques pour la synthèse d'images

soutenue le premier Septembre 1997 devant le jury composé de :

MM.	Jacques	Voiron	Président
	Bruno	Arnaldi	Rapporteur
	Michiel	van de Panne	Rapporteur
	Claude	Puech	Directeur
	Jean-Dominique	Gascuel	
	Roger	Mohr	

Thèse préparée au sein de l'équipe *iMAGIS* du laboratoire *GRAVIR-IMAG*
iMAGIS est un projet commun entre le *CNRS*, l'*INRIA*, l'*INPG* et l'*UJF*.

Merci

Merci à tous ceux qui m'ont aidé ou encouragé au cours de ces quatre années. J'ai le sentiment d'avoir pu travailler dans des conditions particulièrement agréables et stimulantes. À Claude Puech qui m'a accueilli dans son équipe et m'a accordé son soutien et sa confiance. À Roger Mohr pour sa chaleureuse bienveillance. À Mathieu Desbrun, l'indispensable, à Frédo Durand, l'expert, à Marie-Paule Gascuel, la stratège, à Jean-Dominique Gascuel, le magicien. Et à tous ceux qui contribuent à l'alchimie vivifiante de l'équipe iMAGIS.

Table des matières

I Animation de Solides pour la Synthèse d'Images 1

Introduction 3

1 État de l'art 7
1.1 Cinématique des structures articulées . . . 8
1.1.1 Cinématique inverse par résolution linéaire . . . 9
1.1.2 Cinématique des positions . . . 9
1.2 Dynamique des structures articulées . . . 10
1.2.1 Résolution des équations dynamiques des solides . . . 11
1.2.2 Collisions et contacts . . . 13
1.3 Domaines connexes . . . 15
1.3.1 Matériaux déformables . . . 15
1.3.2 Contrôle d'animation . . . 16

2 Rappels 17
2.1 Cinématique . . . 17
2.1.1 Cinématique du point . . . 17
2.1.2 Cinématique du solide . . . 18
2.1.3 Notations cinématiques . . . 19
2.2 Dynamique du solide . . . 20
2.2.1 Loi de Newton . . . 21
2.2.2 Actions dynamiques . . . 21
2.2.3 Inertie d'un solide . . . 22
2.2.4 Repère galiléen . . . 23
2.2.5 Quantités de mouvement . . . 23
2.2.6 Quantités d'accélération . . . 24
2.2.7 Principe fondamental de la dynamique . . . 24
2.2.8 Puissance et énergie . . . 26
2.3 Structures articulées . . . 28
2.3.1 Articulations et contraintes cinématiques . . . 28
2.3.2 Graphe cinématique . . . 29
2.3.3 Systèmes de coordonnées . . . 30
2.3.4 Chaînes fermées . . . 31
2.3.5 Représentation du graphe cinématique . . . 33
2.4 Équations de mouvement . . . 33
2.4.1 Équations géométriques . . . 33

2.4.2 Équations cinématiques 35
2.4.3 Mise en oeuvre de l'animation par modèles physiques 36

3 Modélisation cinématique de scènes 37
3.1 Spécification des liaisons 37
3.2 Repère intermédiaire de liaison 38
3.3 Mise à jour des positions relatives 40
3.3.1 Calcul du nouveau repère intermédiaire 40
3.3.2 Prise en compte du temps 41
3.3.3 Le problème des rotations 41
3.3.4 Directions indépendantes 42
3.3.5 Généricité 43
3.4 Le problème du contact extérieur en coordonnées relatives 44
3.4.1 Générer la cinématique du contact 44
3.4.2 Une solution pour la cinématique du contact 45
3.5 Poser les équations de mouvement 49
3.5.1 Influences cinématiques 49
3.5.2 Influences dynamiques 50
3.5.3 Équations cinématiques 51
3.5.4 Écriture des équations cinématiques 52
3.5.5 Implémentation 54
3.6 Applications 56
3.6.1 Structures articulées 56
3.6.2 Contrôle du mouvement 56
3.7 Conclusion 56

4 Approche cinématique de l'animation 59
4.1 Contrôle du mouvement 59
4.1.1 Approches par optimisation 60
4.1.2 Pseudo-inverses droite et gauche 62
4.2 Résolution de niveaux successifs de contraintes 63
4.2.1 Décomposition par valeurs singulières de la matrice jacobienne . . . 64
4.2.2 Résoudre un niveau de contrainte 65
4.2.3 Résoudre le niveau suivant 66
4.2.4 Algorithme complet 66
4.3 De la Cinématique à la Dynamique 67
4.3.1 Principe de d' Alembert 67
4.3.2 Applications 69
4.4 Bilan 70

5 Approche dynamique de l'animation 71
5.1 Écriture des équations 71
5.1.1 Influences extérieures 72
5.1.2 Influence des articulations 73
5.1.3 Influence des solides 74
5.1.4 Équation dynamique 74
5.2 Méthodes cubiques 75

5.2.1 Résolution directe de la matrice dynamique . . . 75
5.2.2 Matrices singulières . . . 77
5.2.3 Approche locale . . . 77
5.2.4 Approche locale à propagation immédiate . . . 80
5.2.5 Approche semi-locale . . . 82
5.3 Méthode quadratique: le gradient conjugué . . . 82
5.4 Une méthode linéaire . . . 84
5.5 Méthodes mixtes . . . 86
5.5.1 Résolution mixte directe . . . 87
5.5.2 Résolution mixte itérative . . . 89
5.6 Résultats . . . 90
5.6.1 Chaîne simple . . . 91
5.6.2 Comparaison des méthodes locales itératives . . . 91
5.6.3 Méthodes rapides . . . 94
5.7 Dynamique inverse . . . 99
5.8 Conclusion . . . 102

6 Vers une approche temps réel 103
6.1 Algorithme d'animation . . . 104
6.1.1 Intégration numérique . . . 104
6.1.2 Compatibilité des vitesses . . . 104
6.1.3 Compatibilité des positions . . . 104
6.1.4 Collisions . . . 106
6.1.5 Boucle d'animation . . . 107
6.2 Résolution itérative des forces de contact . . . 108
6.2.1 Contraintes de contact . . . 109
6.2.2 Approche itérative globale des forces de contact . . . 111
6.2.3 Résolution de contraintes mixtes . . . 111
6.2.4 Amélioration itérative des solutions dynamiques . . . 112
6.3 Conclusion . . . 114

Conclusion 115

II Approche Photométrique pour la Modélisation de Surfaces à Base d'Images 117

Introduction 119

1 État de l'art 121
1.1 Rappels . . . 121
1.1.1 Définitions . . . 121
1.1.2 Réflection spéculaire et diffuse . . . 121
1.1.3 Approche physique . . . 124
1.1.4 Conclusion . . . 127
1.2 Shape from Shading . . . 127
1.2.1 Principe . . . 127

1.2.2 Image unique . . . 130
1.2.3 Stéréovision Photométrique . . . 131
1.3 Mesure de la BRDF . . . 133

2 Stéréovision photométrique rapide 137
2.1 Choix de modèle . . . 137
2.1.1 Modèles classiques . . . 137
2.1.2 Créer de nouveaux modèles pour la vision . . . 138
2.1.3 Modèle choisi . . . 139
2.2 Reconstruction des normales . . . 140
2.2.1 Hypothèses . . . 141
2.2.2 Application à un modèle bilobe . . . 143
2.2.3 Évaluation de la qualité du résultat . . . 143
2.2.4 Comparaison avec une méthode classique . . . 144
2.2.5 Retrouver les sources d'éclairage . . . 146
2.2.6 Application à une caméra mobile . . . 147
2.3 Conclusion . . . 147

3 Mesure de la BRDF 151
3.1 Reconstruction du modèle . . . 151
3.1.1 Détermination des paramètres . . . 151
3.1.2 Évaluation du résultat . . . 152
3.1.3 Expérimentation . . . 152
3.2 Détermination simultanée du modèle et de la géométrie . . . 153
3.2.1 Optimisation itérative des paramètres . . . 153
3.2.2 Expérimentation . . . 154
3.3 Essai sur des images réelles . . . 155
3.4 Bilan . . . 156

Conclusion 157

A Principaux symboles physiques utilisés 159

B Algorithmes de dynamique 161
B.1 Résolution acyclique . . . 161
B.2 Produit restreint . . . 163

C Coordonnées homogènes 165

D Centres et axes de liaisons 167
D.1 Axe instantané de rotation en 2D . . . 167
D.2 Cas tridimensionnel . . . 168

E Exemple de modélisation 169

Table des figures

1.1 Le squelette d'un personnage, composé de solides et d'articulations. 7
1.2 Une liaison pivot glissant et ses deux mouvements relatifs autorisés. 8
1.3 Deux solides 1 et 2 en contact. 13

2.1 Un point et deux repères 17
2.2 Une action dynamique entre deux solides 22
2.3 Un solide soumis à deux actions dynamiques 25
2.4 Une structure articulée, au repos. 28
2.5 Une liaison point-point. 28
2.6 Liaison pivot d'axe $\boldsymbol{i}$ entre S_1 et S_2. Le centre de liaison peut être placé n'importe où sur l'axe. 29
2.7 Un personnage et un graphe cinématique associé 30
2.8 Trois solides et un paramétrage relatif 31
2.9 Des liaisons à six degrés de liberté $(x, y, z, \alpha, \beta, \gamma)$ par rapport au repère absolu définissent un paramétrage absolu des solides. 32
2.10 Une chaîne fermée et son graphe cinématique. Les disques noirs représentent des articulations. 32
2.11 Système inconsistant. L'état du système varie avec le graphe cinématique choisi. 33
2.12 Un exemple du formalisme que nous utiliserons pour les graphes cinématiques. Les flèches simples représentent des pointeurs, les doubles des listes de pointeurs. Un solide peut avoir plusieurs mères en cas de chaînes fermées. Le solide 0, repère absolu, n'a pas de mère. 34
2.13 Une chaîne fermée simple avec ses paramètres et un graphe cinématique. . 34
2.14 Une structure à une chaîne fermée 35

3.1 Une structure cinématique 38
3.2 Principales liaisons technologiques. La liaison linéaire annulaire pourrait aussi être appelée "rotule glissante"ou "sphère-gouttière". 39
3.3 Repère intermédiaire de liaison 39
3.4 Repère intermédiaire de liaison pour un appui plan 40
3.5 Les angles d'Euler correspondent à des rotations successives 42
3.6 Une liaison linéaire rectiligne L peut être modélisée par deux contacts ponctuels L' et L''. Les lettres r et t indiquent les axes de rotations et translations relatives autorisées. 42
3.7 Le problème de l'intégration des vitesses 44

3.8 Mécanismes virtuels et contraintes nécessaires pour simuler le roulement et le glissement . . . 44
3.9 Utilisation du repère intermédiaire. . . . 45
3.10 Géométrie du contact . . . 47
3.11 Un contact entre solides est modélisé une fois la collision détectée. . . . 48
3.12 Un mécanisme à une chaîne fermée . . . 51
3.13 Algorithme d'écriture des équations de vitesse . . . 53
3.14 Algorithme d'écriture des équations d'accélération . . . 53
3.15 Champs et méthodes principaux des mobilités . . . 54
3.16 Champs et méthodes principaux des articulations . . . 55
3.17 Champs et méthodes principaux des solides . . . 55
3.18 Un personnage articulé faisant intervenir quatre sortes de liaisons . . . 56
3.19 Une liaison rotule sur S_1 et une liaison linéaire annulaire sur S_2 sont utilisées pour amener progressivement les points P_1 de S_1 et P_2 de S_2 à coïncider. . . 57

4.1 Une infinité de solutions sont admissibles pour satisfaire la contrainte symbolisée par la flèche pointillée. . . . 60
4.2 Un personnage les pieds fixés au sol a pour contrainte de toucher l'objet à droite de l'image. Les pointillés représentent les contraintes cinématiques. 60
4.3 Les méthodes d'optimisation cherchent le meilleur compromis entre les contraintes. . . . 61
4.4 La pseudo-inverse à droite permet de prendre en compte des priorités entre les contraintes. . . . 63
4.5 En pointillé: résultat avec deux niveaux de priorité. En continu: résultat avec trois niveaux. . . . 64
4.6 Résoudre plusieurs niveaux de contrainte . . . 66
4.7 Un mécanisme à un degré de liberté. . . . 68

5.1 Trois solides et deux liaisons . . . 73
5.2 Écriture de la matrice dynamique . . . 75
5.3 Un mécanisme composé de quatre solides . . . 76
5.4 Un autre mécanisme composé de quatre solides . . . 76
5.5 Système possédant une infinité de solutions. . . . 77
5.6 Système ne possédant aucune solution. . . . 77
5.7 Résolution locale itérative . . . 78
5.8 Une configuration problematique pour les méthodes locales . . . 80
5.9 (a): barre mal conditionnée. (b): barre bien conditionnée . . . 80
5.10 Résolution locale à propagation immédiate . . . 81
5.11 Oscillation infinie de la résolution locale itérative. . . . 81
5.12 Une barre soumise à deux contraintes incompatibles . . . 82
5.13 Approche semi-locale . . . 82
5.14 Approche semi-locale à propagation immédiate . . . 83
5.15 Produit matriciel décomposé réalisant $\boldsymbol{x} = \boldsymbol{D} \cdot \boldsymbol{f}$ . . . 84
5.16 Un mécanisme acyclique . . . 85
5.17 Deux chaînes fermées à résoudre. . . . 86
5.18 $ecrireMatriceFermetures(\boldsymbol{D}_c)$ . . . 87
5.19 Résolution mixte directe de $\boldsymbol{D} \cdot \boldsymbol{f} = \boldsymbol{e}$ . . . 88

5.20 Une structure cinématique variable. 89
5.21 Résolution mixte itérative de $\boldsymbol{D} \cdot \boldsymbol{f} = \boldsymbol{e}$ 90
5.22 Première configuration test. 91
5.23 Résultats de quatre approches de complexité cubique. 92
5.24 Comparaison entre méthodes de différentes complexités théoriques 92
5.25 Un mécanisme dont les contraintes interagissent fortement 93
5.26 Supériorité des méthodes à propagation immédiate parmi les méthodes locales 93
5.27 Le réseau dans son état initial. 94
5.28 Le réseau après application des forces. 94
5.29 Temps de calcul pour différentes méthodes en fonction de la taille du problème. 95
5.30 Temps de calcul pour différents facteurs de réduction d'erreur, méthode du gradient conjugué. 96
5.31 Temps de calcul pour différents facteurs de réduction d'erreur, méthode mixte itérative. 97
5.32 Temps de calcul sur une structure comportant peu de chaînes fermées . . . 97
5.33 Temps de calcul pour différents facteurs de réduction d'erreur, méthode du gradient conjugué. 98
5.34 Temps de calcul pour différents facteurs de réduction d'erreur, méthode mixte itérative. 98
5.35 Treillis à mailles triangulaires, $n = 3$. 99
5.36 Temps de calcul pour le treillis. 100
5.37 Temps de calcul à différentes précisions pour le treillis par la méthode mixte itérative. 100
5.38 Les contraintes d'une liaison passive (a) et d'une liaison active (b). 101

6.1 (a): configuration à l'instant t. (b): configuration après intégration des mouvements (c): configuration à l'instant $t + dt$ 105
6.2 Traitement classique des collisions. (a): instant t. (b): instant t+dt, détection de collision (c): retour à l'instant de collision et traitement des contraintes (d): nouvelle configuration à $t + dt$ 107
6.3 Prise en compte différée des collisions. (a): instant t. (b): instant t+dt, détection de collision (c): rétablissement de la contrainte géométrique (d): traitement des vitesses. 108
6.4 Corps de la boucle d'animation 108
6.5 Géométrie du contact entre les solides 1 et 2. 109
6.6 Calcul itératif des forces de contact 113

1.1 Intensité lumineuse 122
1.2 Réflection spéculaire: $\theta_i = \theta_r$ 122
1.3 Réflection lambertienne: redistribution uniforme 122
1.4 Le modèle de Phong 123
1.5 Vecteurs et angles 123
1.6 Phénomènes dus à la rugosité des surfaces 124
1.7 125
1.8 Répartion de l'énergie réfléchie 126
1.9 Une forme ombrée 127

1.10 Une carte de réflectance . . . 129
1.11 Pour une certaine luminosité réémise par une surface lambertienne, l'ensemble des normales possibles repose sur un cône de révolution. . . . 130
1.12 Une même surface vue sous trois éclairages différents . . . 132
1.13 Dans le cas général, trois équations sont nécessaires pour lever toute ambiguité. . . . 132
1.14 intersection de trois cartes de réflectance. La normale locale à une surface est déterminée par l'intersection des isophotes correspondant aux différentes luminosités du point pour chaque éclairage. . . . 133
1.15 Ce dispositif permet, en faisant varier les angles θ_s, ϕ_v et θ_v, de cartographier des BRDF de surfaces isotropes. . . . 134
1.16 La caméra recueille, par réflection sur l'hémissphère semi-transparent, la luminosité réfléchie dans toutes les directions. . . . 134

2.1 Une BRDF définie par la somme de deux lobes. . . . 139
2.2 Le modèle choisi permet de représenter aussi bien la réflection diffuse que la réflection quasi-spéculaire. . . . 140
2.3 Les surfaces tendent à devenir spéculaires pour les incidences rasantes. . . . 140
2.4 La fonction G permet, connaissant les valeurs L_i et $\boldsymbol{n}.\boldsymbol{v}$, de déterminer $\boldsymbol{n}.\boldsymbol{s}_i$ et $\boldsymbol{n}.\boldsymbol{h}_i$ . . . 141
2.5 La monotonie de S permet une recherche de $\boldsymbol{n}.\boldsymbol{v}$ par dichotomie . . . 142
2.6 $\|\boldsymbol{n}\| \simeq 1$, $\|\boldsymbol{n} - \boldsymbol{n}*\| \simeq \alpha$ . . . 144
2.7 Vue filaire de la surface-test . . . 145
2.8 Erreur objective en fonction de K_d . . . 146
2.9 Erreur objective en fonction de K_s . . . 146
2.10 Erreur objective en fonction de n . . . 147
2.11 Erreur évaluée par la méthode des racines en fonction de K_d . . . 147
2.12 Erreur évaluée par la méthode des racines en fonction de K_s . . . 148
2.13 Erreur évaluée par la méthode des racines en fonction de n . . . 148
2.14 Erreur évaluée par la méthode de minimisation en fonction de K_d . . . 149
2.15 Erreur évaluée par la méthode de minimisation en fonction de K_s . . . 149
2.16 Erreur évaluée par la méthode de minimisation en fonction de n . . . 149
2.17 Comparaison d'erreurs évaluées . . . 150

3.1 La bonne valeur de K_d produit la meilleure droite . . . 151
3.2 Les BRDF seront représentées à φ_s et φ_v constants . . . 152
3.3 $\sigma_{bruit} = 0.01$ . . . 153
3.4 $\sigma_{bruit} = 0.02$ . . . 153
3.5 $\sigma_{bruit} = 0.05$ . . . 154
3.6 Reconstruction simultanée de la géométrie et de la réflectance. . . . 154
3.7 Trois images utilisées pour reconstruire une surface, . . . 155
3.8 surface reconstruite . . . 156

B.1 classe noeud . . . 161
B.2 procedure linearFactor . . . 162
B.3 procedure linearSolve . . . 162
B.4 procedure numeroteFermetures . . . 163

B.5 procedures lireVecteur et ecrireVecteur 164
B.6 procedure produitRestreint 164

C.1 Deux points, un bipoint 165
C.2 Un point $\boldsymbol{M}$ dans deux repères différents 166

D.1 Une liaison sans centre fixe 167
D.2 Centre instantané de rotation 168

Première partie

Animation de Solides pour la Synthèse d'Images

Introduction

Les chercheurs en synthèse d'images rêvent de créer des scènes et mondes imaginaires dont l'ordinateur simulerait l'existence et fournirait des images. Pour cela, ils utilisent des "maquettes numériques" qui sont des représentations abstraites gérables par la machine. L'affichage des scènes ainsi définies est la branche la plus mature et la plus luxuriante de cette science qu'est la synthèse d'images. Des résultats spectaculaires ont été obtenus pour la visualisation de scènes complexes incluant quantités de formes, matières et effets de lumière.

Pour insuffler la vie dans ces mondes numériques il convient de leur ajouter une dimension supplémentaire, le temps, qui introduit variation, causalité et histoire. La synthèse d'animations consiste à générer les variations temporelles et, en tout premier lieu, le mouvement. Pour cela, le temps est découpé en durées élémentaires qui en se succédant simulent l'évolution continue de la réalité. Le mouvement naît comme au cinéma de la succession des positions propres à chaque durée élémentaire. Parmi tous les mouvements possibles, il en est une vaste classe particulièrement intéressante, celles des mouvements de solides en trois dimensions. Ils conviennent pour représenter tous les objets rigides ou dont les déformations restent imperceptibles. Reliés par des articulations, ils permettent de construire des mécanismes ou des êtres animés.

L'avantage et l'inconvénient des maquettes numériques est qu'elles nous laissent libres de spécifier à notre guise les positions au cours du temps. Cette liberté nous oblige à prendre en compte la signification des mouvements que nous désirons créer. Cette signification est étroitement reliée à celle de la maquette à animer. Prenons l'exemple d'un pendule. S'il se décroche de son point d'articulation, il perd son sens de pendule pour devenir un simple projectile. Le respect des contraintes cinématiques associées aux articulations est donc un élément nécessaire à la signification, la qualité, d'un mouvement. Elle n'est pas pour autant suffisante. La notion de pendule suppose l'existence d'un champ de force dans lequel l'objet peut osciller. Le mouvement du pendule ne sera crédible que s'il est compatible avec les influences dynamiques de son milieu extérieur. Si nous le supposons dans un champ de pesanteur, nous nous attendons à le voir osciller périodiquement autour de sa position la plus basse. C'est la prise en compte simultanée des contraintes cinématiques et des influences dynamiques qui permet la simulation de phénomènes naturels. La dynamique influence les corps, la cinématique préserve leur signification. La simulation peut être qualifiée d'animation "réactive", car elle représente la réponse de structures inertes à des sollicitations extérieures.

L'animation de créatures autonomes nécessite une autre approche. La signification réside alors dans la réalisation de scénarios censés représenter les volontés. La crédibilité d'un mouvement de marche bipède, par exemple, nécessite avant tout une progression vers l'avant et un certaine périodicité dans le mouvement. La prise en compte d'influences

dynamiques a moins d'importance que dans le cas du pendule. Le but principal est de faire exécuter des actions plus ou moins précisément spécifiées, aussi pouvons-nous qualifier cette approche d'animation "active". Disposer de "muscles" permet et nécessite de faire des choix parmi l'infinie variété des mouvements autorisés par les contraintes cinématiques.

Le mouvement inerte peut être considéré comme un simple cas particulier, correspondant à des forces musculaires nulles. C'est dans cette optique que la première partie de ce travail de thèse a été réalisé. On peut l'appeler approche cinématique, permettant l'animation active. La difficulté consiste à caractériser à chaque instant l'ensemble des mouvements autorisés par les articulations. Cette caractérisation peut être suivie de n'importe quelle procédure de décision, permettant de se rapprocher d'objectifs spécifiés ou de suivre les lois de la dynamique.

L'animation réactive nécessite moins de calculs car elle se contente de déterminer un mouvement particulier, celui qui respecte les lois de la dynamique pour des influences données. L'inconvénient est que l'animateur perd en partie le contrôle du mouvement. Il lui reste la possibilité d'appliquer des forces sur les objets, mais sans en connaître précisément d'avance le résultat. Dans bien des cas ce faible contrôle est suffisant, et les gains en rapidité de calcul permettent dans certains cas d'exécuter des simulations en "temps réel". On peut alors envisager des systèmes interactifs entre un monde virtuel dynamique et un utilisateur. Cette interactivité apporte l'étincelle de vie qui manque dans les systèmes d'animation traditionnels. Les systèmes temps réel sont astreints à évoluer à une vitesse donnée et gèrent le temps comme une ressource limitée. Ils nécessitent de pouvoir allouer un temps de calcul maximal à chacune des diverses sous-tâches à accomplir. On sait depuis longtemps mettre en équations et résoudre la dynamique du solide, mais les algorithmes classiques fournissent la solution en un nombre bien déterminé d'opérations. Ce comportement est incompatible avec l'allocation de durées de calcul limitées, car aucune alternative n'est possible si le temps autorisé est inférieur au temps requis. Il donc est préférable de mettre au point des résolutions itératives, avançant par petits pas vers la solution exacte. On peut alors moduler le temps de calcul en limitant le nombre de pas, au détriment de la précision. Ce compromis est la condition indispensable des applications temps réel.

Deux algorithmes itératifs pour l'animation de solides articulés ont été inventés et implémentés durant ce travail de thèse. L'un traite de la résolution des systèmes d'équations de la dynamique du solide, et met en oeuvre une technique efficace de résolution des systèmes linéaires, que l'auteur n'a jamais vu mentionée dans la littérature. L'autre algorithme itératif traite de la résolution de ces mêmes systèmes assortis des inéquations de comportement des contacts avec frottements de Coulomb.

Un autre problème soulevé par l'animation en temps réel est l'intégration du mouvement. Les équations de la dynamique décrivent l'état du système à un instant donné, et leur solution n'est donc valable qu'à cet instant. Or la discrétisation du temps nous oblige à considérer valide cette solution pendant toute une durée élémentaire. Les nouvelles positions déduites sont donc légèrement fausses au regard des contraintes cinématiques. En conséquence, les contraintes cinématiques ne sont pas respectées. Les schémas d'intégration à pas de temps adaptatif permettent en théorie de garder cette erreur sous un seuil arbitrairement bas, mais au prix d'un nombre imprévisible d'opérations. L'approche originale de l'animation pratiquée dans l'équipe iMAGIS depuis un certain nombre d'années est à cet égard très intéressante. Plutôt que de confier à un schéma d'intégration le soin

de maintenir la précision géométrique, une passe de rétablissement des contraintes à l'aide de petits déplacements est appliquée avant affichage. La signification du mouvement est donc préservée, au prix d'une légère dégradation de la précision dynamique. On gagne en retour la possibilité de gérer la vitesse de simulation en imposant les longueurs des durées élémentaires. Cette approche paraît donc très bien adaptée aux contraintes temps réel. Une boucle d'animation basée sur ce principe et sur l'utilisation des algorithmes de résolution itérative a donc été mise au point. Elle inclut en outre la prise en compte de collisions intervenues au cours du pas de temps, sans effectuer de retour en arrière.

Le premier chapitre de ce document contient des rappels de cinématique et de dynamique du solide. Le deuxième porte sur la notion d'articulation et en présente un modèle assez générique. Un certain nombre de notations utilisées par la suite sont présentées dans ces deux chapitres. Le troisième présente des travaux sur l'approche cinématique du mouvement. Le quatrième chapitre contient une étude comparative d'un certain nombre d'algorithmes de résolution des équations de la dynamique, ainsi que la nouvelle solution itérative proposée. Des résultats de tests, présentés sous forme de graphiques, montrent une évaluation des performances relatives pour différents types de scènes. Le dernier chapitre traite de l'approche temps réel et présente la résolution itérative des forces de contact ainsi que la boucle d'animation à pas de temps imposé.

Chapitre 1

État de l'art

La synthèse d'animation par ordinateur fait depuis longtemps appel aux modèles solides dans de nombreuses applications. Pour l'animation de personnages, par exemple, on définit la morphologie générale par un squelette composé de solides reliés par des articulations, comme illustré sur la figure (1.1). La chair, la peau et les vêtements peuvent être ajoutés autour de ces objets de base pour obtenir une meilleure esthétique.

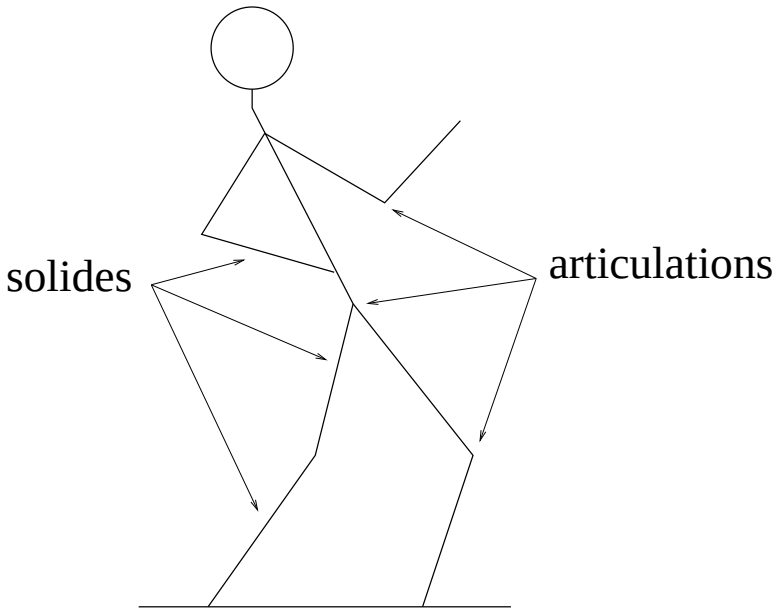


FIG. 1.1 - *Le squelette d'un personnage, composé de solides et d'articulations.*

L'animation de solides peut être considérée au premier abord comme un problème purement géométrique. Étant donné un ensemble de solides reliés par des articulations, la question est de savoir comment bouger ces solides. Bien entendu, la réponse dépend largement du type de mouvement que l'on veut obtenir. Pourtant, dans tous les cas, les articulations entre les solides vont jouer un rôle de tout premier plan. Les articulations, ou liaisons, ont deux aspects duaux. On peut les voir comme un ensemble de mouvements relatifs possibles entre les solides (fig. 1.2), mais aussi comme un ensemble de contraintes, c'est-à-dire de mouvements relatifs impossibles entre les solides. La liaison pivot glissant de la figure (1.2) peut être spécifiée comme ayant deux mouvements relatifs possibles et

combinables: une translation et une rotation selon un axe donné; ou encore deux translations et deux rotations interdites, selon des directions indépendantes orthogonales à l'axe de la liaison.

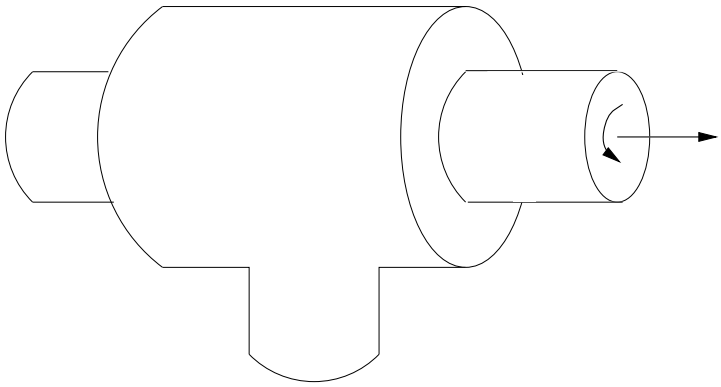


FIG. 1.2 - *Une liaison pivot glissant et ses deux mouvements relatifs autorisés.*

L'animation de solides n'est pourtant pas uniquement un problème géométrique. Le souci de réalisme dans les mouvements a poussé les chercheurs à utiliser des *modèles générateurs* de mouvement, dans lesquels la prise en compte de forces est le premier facteur responsable des déplacements. On peut ainsi générer des mouvements en spécifiant simplement des conditions initiales (positions, vitesses), des attributs (masses, inerties) et des forces agissant sur les objets. Cela permet d'obtenir des animations réalistes en évitant beaucoup de travail fastidieux à l'utilisateur. En revanche, il est malaisé de contrôler le résultat produit, si bien que le contrôle d'animation dynamique est devenu une branche à part entière de la recherche en synthèse d'animations.

Nous passerons tout d'abord en revue les approches cinématiques de l'animation de structures articulées. Nous verrons ensuite les approches dynamiques. Nous terminerons par quelques références sur des sujets connexes non abordés dans ce document.

1.1 Cinématique des structures articulées

La cinématique étudie les relations géométriques entre les objets, et leurs variations au cours du temps. La cinématique inverse consiste à obtenir un mouvement partiellement spécifié. Par exemple, que le personnage de la figure (1.1) pose sa main gauche sur sa tête. Parmi les applications de la cinématique inverse, on trouve:

- la manipulation interactive, où un animateur positionne un personnage par rapport à son environnement en "attrapant"et déplaçant des parties de son corps, à la manière dont on positionnerait un mannequin dans une vitrine;
- la spécification de trajectoires pour permettre la réalisation de scénarios. On veut dans ce cas spécifier une partie du mouvement, par exemple le trajet de la main du personnage au cours du temps, le reste du corps devant suivre;
- la détermination du domaine spatial susceptible d'être atteint par une partie du personnage. Ce problème intéresse particulièrement les roboticiens pour positionner et adapter des robots manipulateurs, généralement fixés au sol;

– l'évitement de collisions afin d'obtenir des comportements réalistes (pour l'animation) ou sans danger (pour la robotique).

Nous nous intéresserons plus particulièrement aux deux premiers thèmes cités. On peut consulter Alameldin et al.[APRB90] sur la question du domaine spatial atteignable, Badler et Zhao[BBG+94, ZB94b] sur la question de l'évitement des collision.

Les contraintes cinématiques sur les vitesses et accélérations se traduisent par des équations linéaires. Les contraintes sur les positions se traduisent par des équations non linéaires. Nous passerons en revue les techniques associées à ces deux catégories de problèmes.

1.1.1 Cinématique inverse par résolution linéaire

La résolution de contraintes cinématiques a été étudiée de longue date par les roboticiens[Pau81]. Beaucoup utilisent l'approche par résolution de la matrice pseudo-inverse des contraintes cinématiques. On peut représenter l'état d'une structure articulée en décrivant la configuration des ses articulations par un ensemble de valeurs appelées variables cinématiques. La liaison pivot glissant de la figure (1.2) possède deux variables indépendantes qui représentent la translation et la rotation relatives des deux solides. Les valeurs de ces variables indépendantes sont souvent notées individuellement q_i, et l'ensemble de ces valeurs est rassemblé dans un vecteur $\boldsymbol{q}$ qui possède autant de composantes qu'il y a de variables indépendantes dans l'ensemble de la structure cinématique. Les variations de ces valeurs sont représentées par leurs dérivées première $\dot{\boldsymbol{q}}$ et seconde $\ddot{\boldsymbol{q}}$ qui représentent respectivement des vitesses et des accélérations. Les valeurs des contraintes cinématiques sont rassemblées dans un vecteur $\boldsymbol{c}$, et l'influence des variables cinématiques sur les contraintes est exprimée par la relation $\boldsymbol{J} \cdot \boldsymbol{\delta q} = \boldsymbol{\delta c}$. La matrice $\boldsymbol{J}$ est appelée le jacobien des contraintes. La résolution des contraintes $\boldsymbol{c}$ par pseudo-inverse consiste à résoudre le problème sous la forme

$$\boldsymbol{\delta q} = \boldsymbol{J}^{+} \cdot \boldsymbol{c}$$

où $\boldsymbol{J}^{+}$ est la pseudo-inverse de la matrice $\boldsymbol{J}$. En effet, la matrice $\boldsymbol{J}$ est rarement carrée, et ne possède donc pas d'inverse strictement définie. Girard et Maciejewski[GM85] en adaptent une approche pour faire suivre à l'extrêmité d'un bras une courbe dans l'espace. Korein et Badler[KB82] explorent plus spécifiquement la question des butées articulaires et des articulations redondantes. Boulic[BTT90] utilise des contraintes de cinématique inverse pour animer la marche d'un personnage, ou optimiser des postures en contrôlant la position du centre de gravité du personnage[BMT95]. Nakamura[Nak91] aborde le problème du calcul robuste de matrices pseudo-inverses pour les systèmes sur-contraints.

1.1.2 Cinématique des positions

La cinématique des positions consiste à chercher des postures satisfaisant certaines contraintes, sans accorder d'importance à la manière dont on peut atteindre ces positions. La cinématique inverse exprimée en termes de positions peut être traitée par minimisation d'une fonction d'erreur. Cette fonction d'erreur doit être croissante en fonction des écarts entre les positions désirées et les positions actuelles. Par exemple, pour une contrainte

point-point, l'erreur peut être définie comme le carré de la distance qui sépare les points. Il est alors facile de pondérer les importances relatives de ces contraintes, ce qui est particulièrement utile dans les cas où l'ensemble des contraintes ne peut être satisfait (systèmes surcontraints). Witkin et al.[WFB87] procèdent itérativement en appliquant des vitesses articulaires colinéaires au gradient de l'erreur cinématique. Le résultat est un mouvement continu à partir d'une configuration fausse vers une configuration qui satisfait les contraintes.

Dans certains cas, seule la position finale présente un intérêt pour l'animateur, par exemple pour la création de postures. Zhao et Badler[ZB94a] optimisent directement la fonction d'erreur dans l'espace des positions articulaires tout en tenant compte de limitations articulaires en combinant l'algorithme de minimisation itérative BFGS[PTVF92] et la méthode de projection de Rosen[Ros60]. À la différence des méthodes précédemment citées, il trouve par essais successifs une configuration satisfaisant les contraintes plutôt que de s'en rapprocher de façon continue en jouant sur les vitesses.

1.2 Dynamique des structures articulées

Les lois de la dynamique s'appliquent simplement aux solides pourvu qu'on connaisse les forces appliquées:

$$m\boldsymbol{a} = \boldsymbol{f}$$
$$\boldsymbol{I}_g \cdot \dot{\boldsymbol{\omega}} = \boldsymbol{c}$$

où m est la masse du solide, $\boldsymbol{I}_g$ sa matrice d'inertie au centre de masse, $\boldsymbol{a}$ son accélération au centre de masse, $\dot{\boldsymbol{\omega}}$ son accélération angulaire, $\boldsymbol{f}$ et $\boldsymbol{c}$ la somme des forces et des moments exprimés au centre de masse. Les corps déformables ont des équations plus complexes, mais qui relient aussi leurs accélérations aux forces appliquées. Les modèles générateurs de mouvement utilisent ces équations pour calculer la trajectoire d'un corps d'après les conditions initiales et les forces appliquées au cours du temps. Le temps est discrétisé en intervalles. Les *schémas d'intégration* sont les algorithmes utilisés pour réévaluer les vitesses et positions à l'issue de chacun de ces intervalles de temps dt en fonction de l'accélération a. Les plus classiques sont celui d'Euler:

$$\begin{aligned} v(t+dt) &= v(t) + a.dt \\ x(t+dt) &= x(t) + v(t+dt).dt \end{aligned}$$

et celui de Newton, aussi appelé schéma d'Euler modifié:

$$\begin{aligned} v(t+dt) &= v(t) + a.dt \\ x(t+dt) &= x(t) + \tfrac{1}{2}(v(t) + v(t+dt)).dt \end{aligned}$$

Le schéma d'Euler sous-entend que les vitesses changent instantanément au début de l'intervalle de temps et restent constantes au cours de celui-ci, alors que le schéma de Newton correspond à l'hypothèse d'accélération constante tout au long du pas de temps. On peut montrer que l'erreur d'intégration est proportionnelle à dt^2 au lieu de dt pour le schéma d'Euler. De plus il donne des résultats exacts dans le cas d'un corps en chute libre, aussi est-il le plus communément utilisé.

Les équations instantanées du mouvement sont posées en appliquant à l'instant considéré les équations dynamiques et cinématiques. Des approches consistant à précalculer une partie des équations ont été proposées, basées sur des calculs symboliques, et donc exacts, du système au cours du temps. Le système DREAM[Coz96] de Rémi Cozot effectue avant l'animation proprement dite une passe de calcul symbolique permettant le développement et la simplification des équations de mouvement. Pour cela, les corps à simuler doivent être décrits dans un formalisme permettant la création d'expressions formelles représentatives du système. Cette méthode permet la génération de code en langage C++ dédié spécialement à chaque simulation[CA95]. La méthode peut être appliquée aux corps déformables[CV96] et au calcul en parallèle[CC96].

1.2.1 Résolution des équations dynamiques des solides

Le problème avec les structures articulées est que les forces internes aux articulations sont généralement inconnues *a priori*. Il existe deux manières de traiter cette difficulté. La première consiste à restreindre la recherche de mouvements à l'espace des mouvements admissibles par les articulations. Cette approche est appelée méthode du solide composé (Composite Rigid-Body Method,CRBM) et peut s'expliciter en suivant le formalisme très générale du principe des travaux virtuels[Wit77, MD82, WB85, IC87, IC88, ADH89]. Les inconnues sont alors les accélérations articulaires. La résolution s'effectue en temps $O(n^3)$ où n est le nombre de variables articulaires. Cette méthode ne résout pas le problème des contraintes complexes comme les chaînes fermées.

L'autre approche, bien connue des roboticiens, est basée sur le formalisme de Newton-Euler et consiste à inclure les contraintes de mouvements interdits dans les équations de la dynamique, dont les inconnues sont alors les accélérations des solides par rapport au repère absolu, au nombre de 6 par solide. On peut alors, pour les structures sans chaîne fermées, effectuer une résolution récursive en parcourant le graphe cinématique dans un sens puis dans l'autre. Cette approche est appelée méthode des corps articulés (Articulated Body Method, ABM)[Ver74, Arm79, Fea83, Fea87, AG85]. La résolution s'effectue en temps $O(n)$ mais ce type de méthode ne traite pas les chaînes fermées. Lathrop[Lat86] propose de traiter les chaînes fermées comme des contraintes entre structures acycliques. Cette idée sera reprise par Baraff[Bar96].

Les travaux et formulations des roboticiens ont mis un certain temps à diffuser dans la communanuté scientifique de synthèse d'animation. Wilhelms et Barsky[WB85] utilisent en 1985 la méthode CRBM pour animer des structures munies de degrés de liberté en translation et rotation. Wilhelms adoptera par la suite[Wil87, MW88, WMS88] l'approche ABM.

Armstrong et Green[AG85] introduisent la même année la méthode ABM pour la synthèse d'animation, limitée aux degrés de liberté en rotation entre les solides.

Barzel et Barr[BVBS87, BB88] proposent une approche originale pour gérer des solides soumis à des forces et des contraintes cinématiques. Elle applique une réduction au cours du temps de la déviation D par rapport aux contraintes articulaires en traitant l'équation différentielle $\ddot{D} + \frac{2}{h}\dot{D} + \frac{1}{h^2}D = 0$. Une erreur initiale D_0 se réduit alors au cours du temps suivant la loi $D(t) = D_0 e^{-\frac{t}{h}}$. Cette approche permet un auto-assemblage élégant de structures cinématiques. La solution instantanée repose sur la résolution d'un système linéaire non carré et parfois singulier, que les auteurs pratiquent en utilisant la décom-

position par valeurs singulières (SVD)[PTVF92]. Cette résolution est en pratique assez longue et le maintien des contraintes peut, de l'aveu des auteurs, être plus efficacement géré en utilisant une méthode ABM ou CRBM.

Isaacs et Cohen s'intéressent à la prise en compte de contraintes cinématiques dans les résolutions dynamiques. Ils montrent[IC87] en 1987 que la méthode CRBM s'adapte très simplement au traitement de contraintes cinématiques simples du genre $q_i = f_i(t)$. La résolution calcule alors la force nécessaire pour réaliser le mouvement imposé. Ils présentent l'année suivante une méthode utilisant les multiplicateurs de Lagrange qui permet la prise en compte de contraintes cinématiques plus complexes comme les fermetures de chaînes[IC88].

Schröder et Zeltzer[SZ90] intègrent la méthode de Featherstone[Fea83] étendue aux chaînes fermées par Lathrop[Lat86] dans leur système d'animation. La méthode de résolution est très (trop) brièvement décrite. D'après les auteurs, les mesures montrent que les temps de calcul sont linéaires en fonction du nombre de degrés de liberté. Toutefois, les chaînes fermées traitées ne sont pas des boucles internes mais des contraintes par rapport au repère absolu.

Gascuel et Gascuel[GG94] proposent une résolution itérative de contraintes de petits déplacements. Les solides sont laissés libres de toute contrainte lors de l'intégration du mouvement. Avant affichage, les contraintes géométriques sont résolues par déplacements des solides. Ces mouvements résolvent les contraintes une par une, de sorte que plusieurs passes peuvent s'avérer nécessaires quand plusieurs contraintes affectent un même solide. Cette approche permet la modélisation de contraintes classiques comme point-point ou très originales comme boîte-boîte. Elle prend en compte les butées articulaires ou les chaînes fermées sans aucune modification de l'algorithme. Elle s'avère expérimentalement de complexité cubique.

Gleicher[Gle94] propose d'exploiter le fait que le système d'équations dynamiques peut être exprimé en fonction de matrices creuses. La résolution par gradient conjugué, qui met uniquement en oeuvre des produits matrice-vecteur, se prête bien aux matrices creuses car les produits peuvent être effectués en temps $O(n)$ où n est le nombre de contraintes cinématiques indépendantes. La résolution nécessitant en théorie n produits pour converger vers la solution exacte, des temps de calcul en $O(n^2)$ peuvent être obtenus. Cette méthode est la plus rapide pour traiter les chaînes fermées. Son inconvénient est d'être considérablement plus lente que la méthode ABM sur les structures acycliques. Un de ses atouts est d'être une méthode itérative de minimisation d'erreur, dont le temps de calcul peut donc être modulé en échange d'une moindre précision sur le résultat.

Dernièrement, Baraff[Bar96] a proposé une méthode dont l'idée de base est de traiter les contraintes de fermeture de boucles indépendamment des contraintes acycliques. On peut alors utiliser une méthode linéaire sur la partie acyclique des contraintes. L'originalité de sa résolution linéaire est de reposer sur une formulation matricielle équivalente à celle d'Isaacs et Cohen[IC88] mais exprimée en coordonnées absolues. Il en résulte une équation matricielle creuse, qui peut être résolue en temps linéaire si la structure est acyclique. Les fermetures de chaînes sont résolues en deux étapes. La première consiste à écrire une matrice dont les coefficients représentent les mouvements aux points de fermetures dans les directions des contraintes en fonction d'efforts appliqués dans les directions de contraintes. La deuxième phase consiste à résoudre un système d'équations concernant uniquement les forces et déplacements aux points de fermetures. Cette méthode est parti-

culièrement efficace quand le nombre de chaînes fermées est faible devant le nombre total de contraintes.

Pour finir sur ces équations et méthodes de résolution, signalons que Cloutier et al.[CPA95] ont comparé les méthodes ABM et CRBM d'un point de vue théorique et pratique. Ils montrent qu'elles ne sont que deux stratégies différentes d'élimination et substitution d'un même système d'équations mécaniques. Ils estiment à $n = 7$ le nombre de variables cinématiques à partir duquel la formulation ABM est plus rapide que la formulation CRBM. Ils constatent expérimentalement une plus grande stabilité numérique de la méthode ABM et en proposent une explication théorique. Cette stabilité permet aux intégrateurs à pas de temps adaptatifs de moins réduire les pas de temps, et donc de moins multiplier les calculs. Dans certains cas, on peut ainsi gagner en temps d'exécution même avec une méthode théoriquement moins rapide.

1.2.2 Collisions et contacts

Des collisions entre les objets d'une scène peuvent se produire au cours d'une animation. Dans ce cas, de nouvelles contraintes apparaissent. Nous ne traiterons pas ici de la détection des collisions et du retour à l'instant de collision, mais de leur prise en compte dans la dynamique du système. La figure (1.3) représente deux contacts au point P. Leur normale commune en ce point est le vecteur $\boldsymbol{n}$. Baraff[Bar89] propose différentes stratégies pour choisir une normale arbitraire dans le cas où celle ci n'est pas clairement définie (surfaces polyédriques).

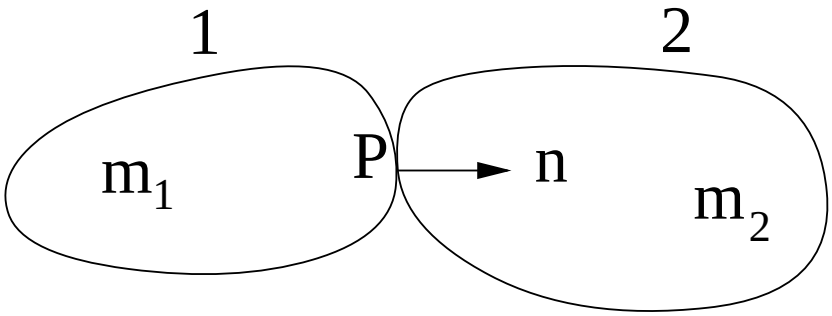


FIG. 1.3 - *Deux solides* 1 *et* 2 *en contact.*

Les contacts surfaciques posent un problème spécifique car les contraintes cinématiques associées s'expriment sous forme d'inéquations. Deux solides peuvent s'éloigner mais pas s'interpénétrer. À l'instant de collision, les vitesses relatives au point de contact tendent à les faire interpénétrer, on est donc obligé de poser des contraintes de vitesses nulles, qui nécessitent le calcul d'impulsions. Les impulsions sont aux vitesses ce que les forces sont aux accélérations. Moore et Wilhelms[MW88] proposent plusieurs modèles dont le plus simple consiste à annuler la composante de la vitesse tangentielle orientée suivant la normale au contact. Hahn[Hah88] modélise le frottement de Coulomb en imposant une vitesse relative nulle y compris selon les directions tangentielles. Si l'impulsion tangentielle f_t est excessive, autrement dit $f_t > \mu f_n$, la contrainte sur les vitesses tangentielles est remplacée par une contrainte sur les forces tangentielles: $f_t = \mu f_n$, où f_n est la force tangentielle et μ le coefficient de frottement. Chaque couple de matériaux a son propre

coefficient de frottement. Une valeur typique est 0.2 (acier-acier). Ces deux approches sont limitées à la prise en compte d'un seul contact simultanément.

Une fois la collision traitée, la question est de savoir si les solides restent en contact. Dans le cas d'une collision élastique, l'impulsion est partiellement emmagasinée et restituée, ce qui décolle à nouveau les solides. Dans le cas de collisions inélastiques ou de contacts préalablement établis, les contraintes se traduisent par des inéquations. Les forces et les accélérations doivent rester répulsives. Baraff[Bar89] propose une approche heuristique pour contacts multiples sans frottement entre surfaces polyédriques. Il l'étendra par la suite à diverses classes de surfaces courbes[Bar90].

Les roboticiens ont une approche du contact plus axée sur la manipulation d'objets, à l'aide de "mains"artificielles ou autres outils, que sur la simulations de corps inertes en collision. Montana[Mon88] étudie la cinématique du contact dans des buts d'analyse des surfaces (calcul des courbures "au toucher") ou de suivi de contours. Chong et al.[CCS93] abordent le problème du calcul des forces de contact avec friction par une approche d'optimisation, utilisée sur des problèmes comportant un nombre faible de contacts (< 10).

Lötstedt[Löt84] aborde le problème du contact sans frottement. Pour simplifier, nous ne considérons dans l'équation que les liaisons de contact surfacique. La prise en compte des autres liaisons est triviale et ne change en rien les algorithmes que nous décrirons. Dans ce problème, seules les équations portant sur la direction normale du contact sont prises en compte. Ceci correspond aux équations:

$$\begin{array}{c} \boldsymbol{a} = \boldsymbol{D} \cdot \boldsymbol{f} + \boldsymbol{b} \\ \forall i \; a_i \geq 0 \quad f_i \geq 0 \quad f_i a_i = 0 \end{array} \tag{1.1}$$

Dans tout ce document, le point représente indifféremment le produit matriciel ou le produit scalaire. Le vecteur $\boldsymbol{f}$ représente les forces au contact, uniquement dans la direction normale pour cette équation, le vecteur $\boldsymbol{a}$ rassemble les accélérations relatives dans les directions des forces de contacts, et la matrice $\boldsymbol{D}$ code les relations entre forces et accélérations. Le vecteur $\boldsymbol{b}$ rassemble les influences extérieures et inertielles. On peut exprimer le problème (1.1) par

$$\boldsymbol{D} \cdot \boldsymbol{f} + \boldsymbol{b} \geq \boldsymbol{0}, \;\; \boldsymbol{f} \geq \boldsymbol{0}, \quad et \quad \boldsymbol{f}^t \cdot (\boldsymbol{D} \cdot \boldsymbol{f} + \boldsymbol{b}) = 0 \tag{1.2}$$

Nous notons $\boldsymbol{u} \geq 0$ le fait que toutes les composantes de $\boldsymbol{u}$ sont positives. L'équation (1.2) définit ce qu'on appelle un problème de complémentarité linéaire (LCP). Le même problème peut aussi s'exprimer sous la forme:

$$trouver \; \boldsymbol{f} \; qui \; minimise \; \boldsymbol{f}^t \cdot (\boldsymbol{D} \cdot \boldsymbol{f} + \boldsymbol{b}) \quad en \; respectant \quad \left\{ \begin{array}{c} \boldsymbol{D} \cdot \boldsymbol{f} + \boldsymbol{b} \geq \boldsymbol{0} \\ \boldsymbol{f} \geq \boldsymbol{0} \end{array} \right\} \tag{1.3}$$

L'équation (1.3) définit ce qu'on appelle un problème de programmation quadratique (QP). La résolution des problèmes de programmation quadratique a été plus étudiée que la résolution des problèmes de complémentarité linéaire, bien que tous deux soient NP-durs. Aussi Lötstedt a-t-il choisi de résoudre le problème sous cette forme. Malheureusement, le problème du calcul des forces de contact ainsi ramené à un problème plus général perd de sa spécificité et devient par là même plus difficile. En pratique, les conditions $af = 0$ sont particulièrement difficiles à remplir, et l'approche par programmation quadratique s'est révélée infructueuse pour traiter les problèmes du frottement.

L'approche de Baraff[Bar94] consiste à traiter un à un les contacts en les incorporant progressivement dans un ensemble de contacts pris en compte. Cet ensemble est divisé en sous-ensembles C, C^+ et NC qui correspondent respectivement au frottement statique, frottement dynamique et décollement. À chaque nouvelle prise en compte de contact, celui-ci est incorporé dans les ensembles C ou NC suivant l'état courant de son accélération normale. Un nouveau contact dans NC ne génère aucune contrainte supplémentaire et on peut passer au suivant. Un contact dans C entraîne la prise en compte d'une contrainte d'accélération nulle selon la normale. Une résolution est effectuée, en tenant compte des équations relatives à tous les contacts précédemment pris en compte. Cette résolution mène à une solution globale différente, et la différence entre la solution nouvelle et l'ancienne est définie comme la correction induite par la prise en compte du nouveau contact. Cette correction peut amener certains contacts à franchir les limites de leur domaine, c'est à dire ne plus respecter les inégalités associées à leur état courant. On n'applique donc qu'une fraction de cette correction, la plus grande possible qui respecte les inégalités associées aux contacts. Si cette fraction vaut 1, le nouveau contact est entièrement pris en compte et on peut passer au suivant. Sinon, on a atteint les limites des hypothèses de certains contacts. Ces contacts changent alors d'hypothèses, et sont transférés dans un des trois ensembles C, C^+ ou NC. La correction n'ayant pas été appliquée intégralement, on en calcule une nouvelle pour terminer la prise en compte du nouveau contact en tenant compte des nouvelles hypothèses sur l'ensemble des contacts. Cette correction peut à nouveau être appliquée entièrement ou partiellement, suivant les comportements des contacts de C, C^+ et NC. Ce processus est répété jusqu'à ce que le nouveau contact pris en compte vérifie entièrement les conditions de contact. On peut alors prendre en compte un nouveau contact, précédemment ignoré, et ainsi de suite jusqu'à ce que l'ensemble des contacts soit pris en compte et satisfasse les conditions de contact. Une preuve de convergence est fournie, uniquement dans le cas des contacts sans frottement.

Récemment, une approche purement géométrique restreinte aux objets convexes et sans frottements a été présentée[Mil96].

En conclusion, l'approche de Lötstedt effectue une résolution globale par raffinements successifs, mais ne résoud pas le problème du frottement, et la résolution des contraintes $a_i f_i = 0$ est difficile à obtenir par optimisation. À l'inverse, la méthode de Baraff résoud les contraintes de frottement mais prend en compte les contacts un à un. Cette méthode oblige donc à attendre que l'ensemble des contacts aient été traités pour obtenir un résultat. Le dernier contact pris en compte étant susceptible de modifier radicalement la solution globale, cette approche n'est pas satisfaisante dans l'optique d'une application avec contraintes de temps réel.

1.3 Domaines connexes

Nous fournissons ici quelques références à des domaines de la synthèse d'animation non directement étudiés dans ce document.

1.3.1 Matériaux déformables

Un intérêt s'est développé pour des systèmes capables d'animer des matériaux déformables, élastiques ou inélastiques, de topologie fixe ou variable. On distingue principale-

ment les approches reposant sur des modèles de matière continue et les approches reposant sur des modèles de matière partitionnée en éléments discrets. Les modèles continus sont beaucoup utilisés pour animer des "solides déformables", c'est à dire des objets dont la forme peut légèrement varier. Une série d'approches repose sur la mécanique des milieux continus[TPBF87, TW88, TF88]. Les relations entre géométrie et comportements mécaniques ont aussi été abordés par le biais des surfaces implicites[Gas93], les fonctions de potentiels étant corrélées à des modèles simples de comportement mécanique. L'approche des matières déformables par potentiels permet en outre les changements de topologie au cours du temps.

Les sytèmes discrets sont composés d'éléments repérés individuellement exerçant des influences les uns sur les autres. Ils permettent dans certains cas de faire cohabiter et interagir divers modèles de comportement[Luc90, CD95]. Les systèmes masses-ressorts sont utilisés pour les matières très déformables à topologie fixes[Mil88, HW90, Jim93, HA95, MYYT95]. Les matières à topologie non fixe (capables de se séparer en morceaux ou de se regrouper) sont le plus souvent modélisés par des systèmes de particules[MP89, LHVD95, DG95, Lom95, Lom96, DG96, HLV96].

1.3.2 Contrôle d'animation

La synthèse d'animation ne se limite pas toujours à la simulation crédible de phénomènes. Il existe de nombreux cas où l'utilisateur veut pouvoir définir un scénario ou fournir des règles de comportement aux objets animés. Ceci nécessite un contrôle de l'évolution des évènements au cours du temps. La notion de marche, par exemple, implique l'enchaînement d'actions et une certaine périodicité des évémenements. Witkin et Kass[WK88] ainsi que Brotman et Netravali[BN88] étudient la possibilité de spécifier simultanément une configuration de départ et d'arrivée séparées par un certain intervalle de temps. Une solution particulière est trouvée en minimisant une fonction de coût, par exemple l'énergie fournie. Bruderlin et Calvert[BC89] proposent une approche spécifique à la marche. Cohen s'attachera à réduire le nombre d'inconnues du problème en limitant le nombre de paramètres de contrôle[Coh92, GRBC96], hiérarchisant ces paramètres[LGC94] ou en découpant le temps en intervalles définis interactivement par l'utilisateur[Coh92].

La dynamique des personnages peut aussi être contrôlée de manière instantanée par des contrôleurs soigneusement mis au point[MZ90, RH91, HWBO95, LvF96]. Van de Panne et Fiume[vF93, vKF94] étudient la génération automatique de contrôleurs par essais successifs et optimisation de fonction d'évaluation comme la distance parcourue pour la marche, la hauteur atteinte pour le saut, etc. Lamouret[LvdP96] explore la réutilisabilité, dans différentes conditions, de mouvements obtenus par de tels contrôleurs.

Citons enfin quelques approches purement cinématiques de contrôle de mouvement. L'interpolation entre positions-clefs[Stu84, FRMTT86, Las87] reste une technique efficace mais fastidieuse à utiliser. Des méthodes procédurales permettent de définir des mouvements en utilisant un plus haut degré d'abstraction[BTT90, MTT91, KB93, BPW93, BTC94], voire de déformer ou mélanger des mouvements prédéfinis[WP95, UAT95, BW95]. La limitation essentielle de toutes ces méthodes est l'absence d'interaction entre les objets animés et leur environnement, qui pénalise le réalisme des animations.

Chapitre 2

Rappels

Ce chapitre présente quelques rappels de cinématique et de dynamique utiles pour la synthèse d'animation. Ces notions de base sont introduites de manière traditionnelle puis une notation unifiée en dimension six proposée par l'auteur est présentée. La notation en dimension six présente l'avantage d'une relative simplicité dans l'écriture des équations tant cinématiques que dynamiques. Elle sera beaucoup utilisée dans tout ce document. L'annexe A contient un résumé des principaux symboles mécaniques utilisés. Ce chapitre introduit d'autre part les notions de contrainte cinématique et d'articulation, puis le graphe cinématique et les chaînes fermées.

2.1 Cinématique

2.1.1 Cinématique du point

Soit un point $\boldsymbol{M}$ défini par le vecteur $\boldsymbol{O_1 M}$ dans le repère $\mathcal{R}_1$. Dans tout ce document, les caractères gras dénotent des vecteurs ou matrices. Les valeurs scalaires sont représentées par des caractères ordinaires. La figure 2.1 représente ce point, ainsi que deux repères $\mathcal{R}_1(\boldsymbol{O}_1, \boldsymbol{i}_1, \boldsymbol{j}_1, \boldsymbol{k}_1)$ et $\mathcal{R}_2(\boldsymbol{O}_2, \boldsymbol{i}_2, \boldsymbol{j}_2, \boldsymbol{k}_2)$. On note $\boldsymbol{V_1^M}$ la vitesse du point M par rapport au

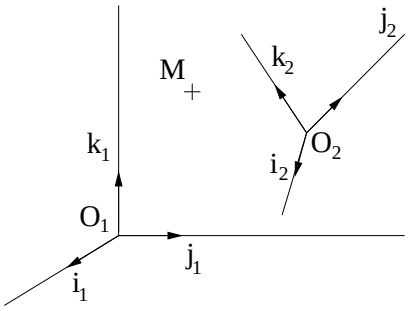


FIG. 2.1 - *Un point et deux repères*

repère R_1, définie par la formule:

$$\boldsymbol{V_1^M} = \left(\frac{d\boldsymbol{O_1 M}}{dt}\right)_{\mathcal{R}_1} \tag{2.1}$$

Les vitesses s'expriment en $m.s^{-1}$. L'indice $\mathcal{R}_1$ indique que la dérivation s'effectue dans le repère $\mathcal{R}_1$, ses vecteurs de base sont donc considérés fixes. L'accélération $\boldsymbol{\Gamma_1^M}$ n'est autre que la dérivée de la vitesse:

$$\boldsymbol{\Gamma_1^M} = \left(\frac{\boldsymbol{V_1^M}}{dt}\right)_{\mathcal{R}_1} = \left(\frac{d^2\boldsymbol{O_1M}}{dt^2}\right)_{\mathcal{R}_1} \tag{2.2}$$

Les accélérations s'expriment en $m.s^{-2}$. La dérivée dans $\mathcal{R}_1$ d'un vecteur $\boldsymbol{u}$ défini dans $\mathcal{R}_2$ vaut:

$$\left(\frac{d\boldsymbol{u}}{dt}\right)_{\mathcal{R}_1} = \left(\frac{d\boldsymbol{u}}{dt}\right)_{\mathcal{R}_2} + \boldsymbol{\Omega_{21}} \wedge \boldsymbol{u} \tag{2.3}$$

Le vecteur $\boldsymbol{\Omega_{21}}$ est le vecteur rotation de $\mathcal{R}_2$ par rapport à $\mathcal{R}_1$. Il est colinéaire à l'axe de rotation relative à l'instant considéré, sa norme vaut la vitesse angulaire relative instantanée, et son sens suit la règle du trièdre direct. L'unité de vitesse angulaire est le radian par seconde qui s'exprime en s^{-1}.

La composition des vitesses relie le mouvement d'un point par rapport à deux repères différents. En utilisant la décomposition $\boldsymbol{O_1M} = \boldsymbol{O_1O_2}+\boldsymbol{O_2M}$ où le point $\boldsymbol{O_2}$ est l'origine de $\mathcal{R}_2$, et les relations (2.1) et (2.3), la vitesse composée est donnée par la relation:

$$\boldsymbol{V_1^M} = \boldsymbol{V_1^{O_2}} + \boldsymbol{\Omega_{21}} \wedge \boldsymbol{O_2M} + \boldsymbol{V_2^M} \tag{2.4}$$

La composition des accélérations s'obtient en dérivant dans $\mathcal{R}_1$ la relation (2.4):

$$\boldsymbol{\Gamma_1^M} = \boldsymbol{\Gamma_{21}^M} + 2\boldsymbol{\Omega_{21}} \wedge \boldsymbol{V_2^M} + \boldsymbol{\Gamma_2^M} \tag{2.5}$$

Les trois termes de la somme sont respectivement l'accélération d'entraînement, l'accélération de Coriolis et l'accélération relative. L'accélération d'entraînement est définie par:

$$\boldsymbol{\Gamma_{21}^M} = \boldsymbol{\Gamma_1^{O_2}} + \left(\frac{d\boldsymbol{\Omega_{21}}}{dt}\right)_{\mathcal{R}_1} \wedge \boldsymbol{O_2M} + \boldsymbol{\Omega_{21}} \wedge (\boldsymbol{\Omega_{21}} \wedge \boldsymbol{O_2M})$$

Elle correspond à l'accélération dans 1 d'un point $\boldsymbol{M}$ fixe dans 2. Le vecteur $\left(\frac{d\boldsymbol{\Omega_{21}}}{dt}\right)_{\mathcal{R}_1}$ est l'accélération angulaire du repère 2 par rapport à 1, dont l'unité est le radian par seconde au carré, i.e. le s^{-2}.

2.1.2 Cinématique du solide

Si $\mathcal{R}_2$ est le repère local d'un solide et M un point quelconque du solide, alors $\boldsymbol{V_2^M} = \boldsymbol{0}$. La vitesse du point M fixe dans le repère $\mathcal{R}_2$ par rapport au repère $\mathcal{R}_1$ est notée V_{21}^M. Ce formalisme est celui de la cinématique du solide. Toute vitesse ou accélération est exprimée en un point considéré fixe dans un repère, et concerne le mouvement de ce repère par rapport à un autre. Le passage de la cinématique du point à la cinématique du solide correspond à adopter la convention:

$$\boldsymbol{V_{21}^A} = \boldsymbol{V_1^A} - \boldsymbol{V_2^A} \tag{2.6}$$

Il s'en suit que $\boldsymbol{V_{21}^A} = -\boldsymbol{V_{12}^A}$

La cinématique du solide est bien adaptée à l'étude des vitesses de solides articulés, car elle fournit des relations simples mettant en jeu les vitesses articulaires. Le champ des vitesses relatives d'un solide S_1 par rapport à un solide S_2 est ainsi totalement défini par la donnée de la vitesse relative $\boldsymbol{V_{21}^A}$ en un point $\boldsymbol{A}$ et de la vitesse angulaire $\boldsymbol{\Omega_{12}}$ de 2 par rapport à 1. En effet, les relations (2.6) et (2.4) entraînent:

$$\boldsymbol{V_{12}^B} = \boldsymbol{V_{12}^A} + \boldsymbol{\Omega_{12}} \wedge \boldsymbol{AB} \tag{2.7}$$

La composition des vitesses angulaires s'exprime par une simple relation de Chasles:

$$\boldsymbol{\Omega}_{31} = \boldsymbol{\Omega}_{32} + \boldsymbol{\Omega}_{21} \tag{2.8}$$

La relation de Chasles est également valable pour les vitesses des points, que nous appellerons parfois vitesse de translation:

$$\boldsymbol{V_{31}^A} = \boldsymbol{V_{32}^A} + \boldsymbol{V_{21}^A} \tag{2.9}$$

Le champ des accélérations d'entraînement est hélas moins simple que le champ des vitesses:

$$\boldsymbol{\Gamma_{21}^N} = \boldsymbol{\Gamma_{21}^M} + \left(\frac{d\boldsymbol{\Omega_{21}}}{dt}\right)_{\mathcal{R}_1} \wedge \boldsymbol{MN} + \boldsymbol{\Omega_{21}} \wedge (\boldsymbol{\Omega_{21}} \wedge \boldsymbol{MN}) \tag{2.10}$$

D'autre part, la relation de Chasles sur les vitesses (eq. 2.9) ne peut s'appliquer aux accélérations d'entraînement, car un point n'est en général fixe que dans un repère donné. La composition des accélérations s'exprime par:

$$\boldsymbol{\Gamma_{i1}^M} = \boldsymbol{\Gamma_{i2}^M} + \boldsymbol{\Gamma_{21}^M} + 2\boldsymbol{\Omega_{21}} \wedge \boldsymbol{V_{i2}^M}$$

Dans la suite de ce document, nous utiliserons parfois indifféremment les termes repère et solide car tout solide peut être doté d'un repère local dans lequel tous ses points matériels sont fixes.

2.1.3 Notations cinématiques

Vitesse généralisée

Soit $\boldsymbol{v}_{ij}^p$ la vitesse généralisée du repère i par rapport à j, exprimée au point $\boldsymbol{p}$.

$$\boldsymbol{v}_{ij}^p = \begin{pmatrix} \boldsymbol{V}_{ij}^p \\ \boldsymbol{\Omega}_{ij} \end{pmatrix} \tag{2.11}$$

Nous utiliserons souvent les lettres minuscules pour représenter les points afin de ne pas trop encombrer graphiquement des notations suffisamment complexes. Ce vecteur à six composantes, défini en un point $\boldsymbol{p}$, définit entièrement le mouvement du solide. En effet, conformément à l'équation (2.7), on peut exprimer la vitesse généralisée en un autre point $\boldsymbol{m}$ par la relation:

$$\boldsymbol{v}_{ij}^m = \boldsymbol{\nabla}_{mp} \cdot \boldsymbol{v}_{ij}^p \tag{2.12}$$

avec

$$\boldsymbol{\nabla}_u = \begin{pmatrix} \mathbf{1} & \boldsymbol{u}^\wedge \\ \mathbf{0} & \mathbf{1} \end{pmatrix} \quad , \quad \boldsymbol{u}^\wedge = \begin{pmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{pmatrix} \quad et \quad \mathbf{1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.13}$$

Nous représenterons par un "$\cdot$" le produit matriciel. Le produit produit scalaire de deux vecteurs $\boldsymbol{u}$ et $\boldsymbol{v}$ sera noté $\boldsymbol{u}^t \cdot \boldsymbol{v}$. L'exposant t représentera la transposition de matrices ou de vecteurs.

Les équations (2.9) et (2.8) de composition des vitesses se traduisent par:

$$\boldsymbol{v}^p_{ik} = \boldsymbol{v}^p_{ij} + \boldsymbol{v}^p_{jk} \tag{2.14}$$

Accélération généralisée

On appellera par la suite accélération généralisée du repère i par rapport à j exprimée au point $\boldsymbol{p}$, *fixe dans* i, le vecteur à six composantes:

$$\boldsymbol{\gamma}^p_{ij} = \begin{pmatrix} \boldsymbol{\Gamma}^p_{ij} \\ (\dot{\boldsymbol{\Omega}}_{ij})_j \end{pmatrix} \tag{2.15}$$

avec

$$(\dot{\boldsymbol{u}})_i = \left(\frac{d\boldsymbol{u}}{dt}\right)_{\mathcal{R}_i}$$

On peut changer de point d'expression en utilisant la relation:

$$\boldsymbol{\gamma}^m_{ij} = \boldsymbol{\nabla}_{mp} \cdot \boldsymbol{\gamma}^p_{ij} + \boldsymbol{\delta}^{mp}_{ij} \tag{2.16}$$

avec

$$\boldsymbol{\delta}^{mp}_{ij} = \begin{pmatrix} -\boldsymbol{\Omega}_{ij} \wedge (\boldsymbol{\Omega}_{ij} \wedge \boldsymbol{mp}) \\ \mathbf{0} \end{pmatrix}$$

Les équations (2.5), (2.8) et (2.3) de composition des accélérations, composition des vitesses de rotation et dérivation vectorielle se traduisent par:

$$\boldsymbol{\gamma}^p_{ik} = \boldsymbol{\gamma}^p_{ij} + \boldsymbol{\gamma}^p_{jk} + \tilde{\boldsymbol{\gamma}}^p_{ijk} \tag{2.17}$$

avec:

$$\tilde{\boldsymbol{\gamma}}^p_{ijk} = \begin{pmatrix} 2.\boldsymbol{\Omega}_{jk} \wedge \boldsymbol{V}^p_{ij} \\ \boldsymbol{\Omega}_{jk} \wedge \boldsymbol{\Omega}_{ij} \end{pmatrix}$$

2.2 Dynamique du solide

La dynamique du solide a pour but de calculer les mouvements de solides soumis à des forces.

2.2.1 Loi de Newton

La célèbre loi de Newton concerne l'accélération $\boldsymbol{a}$ d'un point muni d'une certaine masse m et soumis à une certaine force $\boldsymbol{f}$, et stipule que:

$$\boldsymbol{f} = m \cdot \boldsymbol{a} \tag{2.18}$$

Les vecteurs $\boldsymbol{a}$ et $\boldsymbol{f}$ appartiennent à $\mathbb{R}^3$. Cette relation résume ce qu'il est convenu d'appeler la dynamique du point. Conformément à la relation (2.18), les forces s'expriment en $kg.m.s^{-2}$, ou $Newton(N)$.

Soit un ensemble de points matériels tous fixes dans un certain repère. Ils constituent un solide. La dynamique du solide relie le mouvement de ce solide à l'ensemble des forces extérieures appliquées sur ce solide. En combinant la loi de Newton et la relation (2.10) sur les accélérations d'un solide, on peut montrer qu'il existe un unique point dans ce solide satisfaisant la loi de Newton, en prenant $\boldsymbol{f}$ égal à la somme de toutes les forces extérieures sur le solide. Ce point est le centre de masse du solide, et il est souvent noté $\boldsymbol{G}$. Afin de définir plus précisément les relations entre forces et accélérations, il est bon d'introduire quelques rappels sur la notion d'action dynamique, d'inertie, de quantité de mouvement et d'accélération, avant de présenter le principe fondamental de la dynamique du solide qui rassemble la loi de Newton et la loi sur les rotations.

2.2.2 Actions dynamiques

On regroupe sous le terme d'*actions dynamiques* les forces et moments susceptibles de s'exercer sur un solide. Tout comme les vitesses des solides, les actions dynamiques ont deux composantes et un point d'expression. Nous noterons $\boldsymbol{f}^p$ le vecteur à six composantes représentant une action dynamique exprimée au point $\boldsymbol{p}$:

$$\boldsymbol{f}^p = \begin{pmatrix} \boldsymbol{f} \\ \boldsymbol{c}(\boldsymbol{p}) \end{pmatrix}$$

Le terme $\boldsymbol{c}(\boldsymbol{p})$, parfois improprement appelé couple, désigne le moment de l'action dynamique au point $\boldsymbol{p}$ et s'exprime en $kg.m^2.s^{-2}$, ou $m.N$. Le moment est associé à la notion de rotation. On change le point d'expression d'une action dynamique au moyen de la relation:

$$\boldsymbol{f}^p = \Delta_{pm} \cdot \boldsymbol{f}^m \quad avec \quad \Delta_u = \begin{pmatrix} \mathbf{1} & \mathbf{0} \\ \boldsymbol{u}^\wedge & \mathbf{1} \end{pmatrix} \tag{2.19}$$

où les matrices $\mathbf{1}$ et $\boldsymbol{u}_\wedge$ sont définies comme au chapitre 2.1.3. Nous noterons $\boldsymbol{f}^p_{ij}$ l'action dynamique exercée par le solide i sur le solide j, exprimée en $\boldsymbol{p}$. Le principe de réciprocité des forces stipule qu'en tout point $\boldsymbol{p}$:

$$\boldsymbol{f}^p_{ij} = -\boldsymbol{f}^p_{ji}$$

Sommer des actions dynamiques nécessite de les exprimer au même point:

$$\boldsymbol{f}^p_{(j \cup k)i} = \boldsymbol{f}^p_{ji} + \boldsymbol{f}^p_{ki}$$

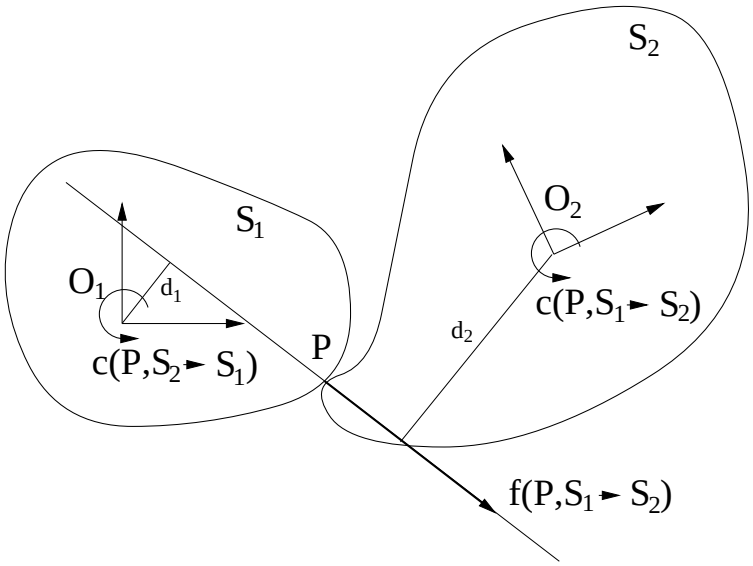


FIG. 2.2 - *Une action dynamique entre deux solides*

La figure (2.2) présente l'exemple d'une action dynamique entre les solides 1 et 2. Nous nous plaçons dans un problème plan où les solides et l'action dynamique appartiennent à un même plan orthogonal à une direction $\boldsymbol{z}$. Cette action se compose d'une force appliquée au point $\boldsymbol{p}$. Exprimée au point $\boldsymbol{p}$, l'action vaut:

$$\boldsymbol{f}_{12}^{p} = \begin{pmatrix} \boldsymbol{f} \\ 0 \end{pmatrix}$$

Nous pouvons exprimer l'action dynamique aux points $\boldsymbol{O}_1$ et $\boldsymbol{O}_2$:

$$\boldsymbol{f}_{21}^{\boldsymbol{O}_1} = \begin{pmatrix} \boldsymbol{f} \\ c(\boldsymbol{O}_1) = d_1 f \boldsymbol{z} \end{pmatrix} \quad et \quad \boldsymbol{f}_{12}^{\boldsymbol{O}_2} \begin{pmatrix} \boldsymbol{f} \\ c(\boldsymbol{O}_2) = d_2 f \boldsymbol{z} \end{pmatrix}$$

où $\boldsymbol{z}$ est la direction normale au plan de la figure.

2.2.3 Inertie d'un solide

L'inertie d'un solide caractérise sa quantité de masse m et la répartition de celle-ci. Le centre d'inertie $\boldsymbol{g}$ d'un solide, aussi appelé centre de masse ou barycentre, est le seul point vérifiant la propriété:

$$\int_{\boldsymbol{p} \in S} \boldsymbol{gp}\ dm = \boldsymbol{0}$$

où dm est la masse du volume élémentaire centré en $\boldsymbol{p}$. Il en découle que pour tout point $\boldsymbol{a}$:

$$\int_{\boldsymbol{p} \in S} \boldsymbol{ap}\ dm = m\boldsymbol{ag}$$

La matrice d'inertie en $\boldsymbol{p}$ d'un solide est une matrice symétrique définie par:

$$\boldsymbol{I}^{\boldsymbol{p}} = \begin{pmatrix} A & -F & -E \\ -F & B & -D \\ -E & -D & C \end{pmatrix}$$

avec:

$$\begin{cases} A = \int_S [(y-y_P)^2 + (z-z_P)^2] dm \\ B = \int_S [(x-x_P)^2 + (z-z_P)^2] dm \\ C = \int_S [(x-x_P)^2 + (y-y_P)^2] dm \end{cases} et \begin{cases} D = \int_S [(y-y_P).(z-z_P)] dm \\ E = \int_S [(x-x_P).(z-z_P)] dm \\ F = \int_S [(x-x_P).(y-y_P)] dm \end{cases}$$

Les termes A, B et C sont appelés les *moments d'inertie*, D, E et F sont les *produits d'inertie*. Leur unité est le $kg.m^2$. Si $\boldsymbol{p}$ est le centre d'inertie, il existe un repère dans lequel seuls les termes diagonaux sont non nuls. Ce repère est appelé repère principal d'inertie du solide. Il est aligné sur les plans de symétrie, s'il en existe. Les moments d'inertie sont d'autant plus faibles que l'ensemble de la masse est proche du centre d'inertie.

Soit $\boldsymbol{g}$ le centre de masse du solide i, de masse m et de matrice d'inertie $\boldsymbol{I}^g$ en $\boldsymbol{g}$. La matrice de masse généralisée du solide i en $\boldsymbol{g}$ est notée $\boldsymbol{M}_i^g$ et est définie par:

$$\boldsymbol{M}_i^g = \begin{pmatrix} m\mathbf{1} & 0 \\ 0 & I^g \end{pmatrix} \quad et \quad \boldsymbol{M}_i^{g^{-1}} = \begin{pmatrix} \frac{1}{m}\mathbf{1} & 0 \\ 0 & I^{g^{-1}} \end{pmatrix} \tag{2.20}$$

La matrice de masse est symétrique, définie, positive. Ceci implique en particulier qu'elle est inversible, et que son inverse possède aussi ces propriétés.

2.2.4 Repère galiléen

Un repère est dit *galiléen* s'il est en translation rectiligne uniforme par rapport au repère absolu. En pratique, on considèrera le repère global de la scène comme galiléen. Les repères galiléens sont les repères où l'expression des lois de la dynamique est la plus aisée car ils ne nécessitent pas l'utilisation de forces d'inertie.

Pour les plus curieux, on peut rappeler au passage que l'hypothèse d'existence d'un repère absolu, référence d'immobilité, est le talon d'Achille de la mécanique newtonienne. Cette hypothèse est absente de la théorie de la relativité. Il va sans dire la mécanique newtonienne, beaucoup plus simple, suffit largement aux applications actuelles d'animation par modèles physiques.

2.2.5 Quantités de mouvement

Nous appellerons quantité de mouvement d'un solide i , de masse m et de matrice d'inertie $\boldsymbol{I}_g$ en son centre de masse $\boldsymbol{g}$, dans son mouvement par rapport à j, le vecteur à six composantes défini en $\boldsymbol{g}$:

$$\dot{\boldsymbol{d}}_{ij}^g = \begin{pmatrix} m\boldsymbol{V}_{ij}^g \\ \boldsymbol{I}^g \cdot \boldsymbol{\Omega}_{ij} \end{pmatrix} = \boldsymbol{M}_i^g \cdot \boldsymbol{v}_{ij}^g \tag{2.21}$$

où le vecteur $\boldsymbol{v}_{ij}^g$ est la vitesse généralisée de i par rapport à j. En pratique, on exprimera les quantités de mouvement et d'accélération (paragraphe suivant) par rapport au repère absolu. Le point sur le vecteur $\dot{\boldsymbol{d}}$ rappelle que cette quantité est liée aux vitesses.

Le parti pris de représenter les entités en dimension six nous conduit à un léger abus de langage. Dans la terminologie standard, la quantité de mouvement est le vecteur $m\boldsymbol{V}_{ij}^g$ (unité: le $kg.m.s^{-1}$), alors que $\boldsymbol{I}^g \cdot \boldsymbol{\Omega}_{ij}$ est appelé *moment cinétique* dont l'unité est le $kg.m^2.s^{-1}$).

On peut changer le point d'expression de la quantité de mouvement au moyen de la relation:

$$\dot{\boldsymbol{d}}^p_{ij} = \Delta_{pm} \cdot \dot{\boldsymbol{d}}^m_{ij}$$

La quantité de mouvement d'un ensemble (ou *système*) de solides est la somme des quantités de mouvement de chacun des solides. Pour être ajoutées, ces quantités de mouvement doivent impérativement être exprimées au même point:

$$\dot{\boldsymbol{d}}^p_{(i\cup j)k} = \dot{\boldsymbol{d}}^p_{ik} + \dot{\boldsymbol{d}}^p_{jk}$$

Les quantités de mouvement sont utilisées pour calculer la réponse des solides à des collisions. Le principe fondamental de la cinétique est que la quantité de mouvement d'un système isolé se conserve.

2.2.6 Quantités d'accélération

Nous appellerons quantité d'accélération d'un solide i, de masse m et de matrice d'inertie $\boldsymbol{I}_g$ en son centre de masse $\boldsymbol{g}$, dans son mouvement par rapport à j, le vecteur à six composantes défini en $\boldsymbol{g}$ par:

$$\ddot{\boldsymbol{d}}^g_{ij} = \begin{pmatrix} m\boldsymbol{\Gamma}^g_j \\ \boldsymbol{I}^g_{\boldsymbol{i}} \cdot \left(\dot{\boldsymbol{\Omega}}_{ij}\right)_j \end{pmatrix} = \boldsymbol{M}^g_i \cdot \boldsymbol{\gamma}^g_{ij} \tag{2.22}$$

Le double point correspond à la dérivée seconde par rapport au temps. Pour simplifier les notations, nous noterons $\dot{\boldsymbol{\Omega}}_{ij} = \left(\dot{\boldsymbol{\Omega}}_{ij}\right)_j$. Comme en cinétique, le repère habituel de référence est le repère absolu, noté 0, et la terminologie utilisée est légèrement abusive. En terminologie standard, la quantité d'accélération est le vecteur $m\boldsymbol{\Gamma}^g_{ij}$ (unité: le $kg.m.s^{-2}$), alors que $\boldsymbol{I}^g \cdot \dot{\boldsymbol{\Omega}}_{ij}$ est appelé *moment dynamique* dont l'unité est le $kg.m^2.s^{-2}$).

On peut changer le point d'expression de la quantité d'accélération par la relation:

$$\ddot{\boldsymbol{d}}^p_{ij} = \Delta_{pm} \cdot \ddot{\boldsymbol{d}}^m_{ij}$$

La quantité d'accélération d'un système de solides est égale à la somme des quantités d'accélération de chaque solide, toutes exprimées en un même point de l'espace.

2.2.7 Principe fondamental de la dynamique

Le principe fondamental de la dynamique appliqué à un solide stipule que la quantité d'accélération du solide par rapport à un référentiel galiléen est égale à la somme des actions dynamiques extérieures exercées sur le solide. Autrement dit, appliqué au solide i, en tout point $\boldsymbol{p}$:

$$\ddot{\boldsymbol{d}}^p_{i0} = \boldsymbol{f}^p_{\bar{i}i} \tag{2.23}$$

où $\bar{i}$ représente l'extérieur du solide i. L'indice 0 correspond à un repère galiléen, généralement le repère absolu de la scène. On en déduit les relations:

$$\boldsymbol{f}^g_{\bar{i}i} = \boldsymbol{M}^g_i \cdot \boldsymbol{\gamma}^g_{i0} \tag{2.24}$$

et

$$\boldsymbol{\gamma}_{i0}^{g} = \boldsymbol{M}_{i}^{g\,-1} \cdot \boldsymbol{f}_{\bar{i}i}^{g} \tag{2.25}$$

L'équation (2.25) est dite de "dynamique directe" car elle consiste à appliquer des forces pour en déduire un mouvement. L'équation (2.24) est dite de "dynamique inverse" car elle consiste à calculer les forces nécessaires pour obtenir un mouvement désiré.

Considérons l'exemple de dynamique directe illustré par la figure (2.3). Un solide S en forme de disque de masse m et de matrice d'inertie $\boldsymbol{I}^{\boldsymbol{G}}$ est en rotation autour de son barycentre $\boldsymbol{G}$. On suppose que $\boldsymbol{V}_{S0}^{\boldsymbol{G}} = \boldsymbol{0}$ à l'instant considéré. La vitesse de son origine $\boldsymbol{O}$ peut être calculée par la relation (2.7). Deux actions dynamiques $(\boldsymbol{f}_1, \boldsymbol{0})$ en $\boldsymbol{A}$ et $(\boldsymbol{f}_2, \boldsymbol{c}_2)$ en $\boldsymbol{B}$ sont appliquées par l'univers extérieur sur ce solide. Par souci de simplicité, on considère que les forces et les axes $\boldsymbol{i}$ et $\boldsymbol{j}$ du repère local de S sont dans le plan de la figure et que le moment $\boldsymbol{c}_2$ est normal à celle-ci. Les axes $\boldsymbol{k}$ du solide et de l'univers sont colinéaires.

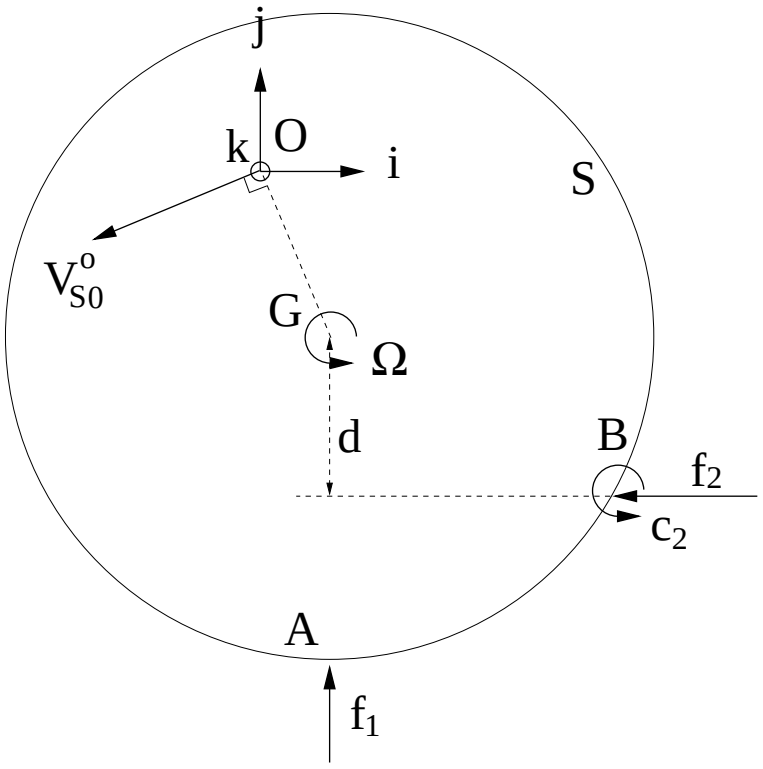


FIG. 2.3 - *Un solide soumis à deux actions dynamiques*

Voyons maintenant comment calculer l'accélération du repère local du solide dans son mouvement par rapport à l'univers. On commence par sommer en $\boldsymbol{G}$ les actions dynamiques extérieures:

$$\boldsymbol{f}_{\bar{S}S}^{G} = \left(\begin{array}{c} \boldsymbol{f}_1 + \boldsymbol{f}_2 \\ \boldsymbol{c}(\boldsymbol{G}) = \boldsymbol{GA} \wedge \boldsymbol{f}_1 + \boldsymbol{c}_2 + \boldsymbol{BG} \wedge \boldsymbol{f}_2 = \boldsymbol{c}_2 - d.f_2 \cdot \boldsymbol{k} \end{array} \right)$$

On peut remarquer que la force $\boldsymbol{f}_1$ ne génère aucun moment en $\boldsymbol{G}$, car sa ligne d'action passe par ce point. Le scalaire f_2 est la norme de la force $\boldsymbol{f}_2$. D'après le principe fondamental (eq. 2.22 et 2.23), nous pouvons maintenant calculer l'accélération du centre de

masse, ainsi que l'accélération angulaire du solide:

$$\boldsymbol{\gamma}_{s0}^{G} = \begin{pmatrix} \boldsymbol{\Gamma}_{S0}^{G} = \frac{1}{m}(\boldsymbol{f}_1 + \boldsymbol{f}_2) \\ \dot{\boldsymbol{\Omega}}_{S0} = \boldsymbol{I}^{\boldsymbol{G}^{-1}} \cdot \boldsymbol{c}(\boldsymbol{G}) \end{pmatrix}$$

La relation (2.10) nous permet de déterminer l'accélération de l'origine du repère:

$$\boldsymbol{\Gamma}_{S0}^{O} = \boldsymbol{\Gamma}_{S0}^{G} + \dot{\boldsymbol{\Omega}}_{S0} \wedge \boldsymbol{GO} + \boldsymbol{\Omega}_{S0} \wedge (\boldsymbol{\Omega}_{S0} \wedge \boldsymbol{GO})$$

Nous venons de traiter un exemple où l'origine du solide, autrement dit le point où est défini et mis à jour son mouvement, diffère du centre de masse. L'exemple a été traité "à la main", voyons maintenant les relations à mettre en oeuvre pour traiter ce problème de façon plus formelle. La relation (2.16) nous donne:

$$\boldsymbol{\gamma}_{S0}^{O} = \boldsymbol{\nabla}_{OG} \cdot \boldsymbol{\gamma}_{S0}^{G} + \boldsymbol{\delta}_{S0}^{OG}$$

On peut alors exprimer le principe fondamental de la dynamique sous la forme:

$$\boldsymbol{\gamma}_{S0}^{O} = \boldsymbol{\nabla}_{OG} \cdot \boldsymbol{M}_{S}^{-1} \cdot \boldsymbol{f}_{\bar{S}S}^{G} + \boldsymbol{\delta}_{S0}^{OG}$$

Pour tout ramener au point $\boldsymbol{O}$, on peut y exprimer les forces extérieures au moyen de la relation (2.19):

$$\boldsymbol{\gamma}_{S0}^{O} = \boldsymbol{\nabla}_{OG} \cdot \boldsymbol{M}_{S}^{-1} \cdot \boldsymbol{\Delta}_{GO} \cdot \boldsymbol{f}_{\bar{S}S}^{O} + \boldsymbol{\delta}_{S0}^{OG}$$

Définissons la matrice de masse $\boldsymbol{M}_{S}^{P}$ au point $\boldsymbol{P}$, du solide S de centre de masse $\boldsymbol{G}$, telle que:

$$\boldsymbol{M}_{S}^{P^{-1}} = \boldsymbol{\nabla}_{PG} \cdot \boldsymbol{M}_{S}^{-1} \cdot \boldsymbol{\Delta}_{GP}$$

En remarquant que par construction, $\Delta_{GP} = \nabla_{PG}{}^{t}$, la matrice $\boldsymbol{M}_{S}^{P}$ est symétrique, positive, définie, comme toute matrice de masse. Ceci nous mène à une formulation du principe fondamental de la dynamique dans le cas où le point d'expression des quantités n'est pas le centre de masse:

$$\boldsymbol{f}_{\bar{S}S}^{P} = \boldsymbol{M}_{S}^{P} \cdot (\boldsymbol{\gamma}_{S0}^{P} - \boldsymbol{\delta}_{S0}^{PG}) \tag{2.26}$$

et

$$\boldsymbol{\gamma}_{S0}^{P} = \boldsymbol{M}_{S}^{P^{-1}} \cdot \boldsymbol{f}_{\bar{S}S}^{P} + \boldsymbol{\delta}_{S0}^{PG} \tag{2.27}$$

2.2.8 Puissance et énergie

Puissance d'une action dynamique

La puissance d'une action dynamique d'un solide 1 sur un solide 2 est égale à son produit par la vitesse relative des solides au même point:

$$P_{12}^{f} = \boldsymbol{v}_{21}^{p\ t} \cdot \boldsymbol{f}_{12}^{p} \tag{2.28}$$

On retrouve le résultat bien connu, que la puissance fait intervenir des produits de vitesses par forces et de vitesses angulaires par moments de forces. L'unité de puissance est le

$kg.m^2.s^{-3}$ aussi appelé le *Watt*(W). Ce produit scalaire (2.28) est indépendant du point d'expression. En effet,

$$\begin{aligned} \boldsymbol{v}_{21}^{p\ t} \cdot \boldsymbol{f}_{12}^{p} &= \boldsymbol{v}_{21}^{a\ t} \cdot \boldsymbol{\nabla}_{ap}^{t} \cdot \boldsymbol{\Delta}_{ap} \cdot \boldsymbol{f}_{12}^{a} \\ &= \boldsymbol{v}_{21}^{a\ t} \cdot \boldsymbol{\Delta}_{pa} \cdot \boldsymbol{\Delta}_{ap} \cdot \boldsymbol{f}_{12}^{a} \\ &= \boldsymbol{v}_{21}^{a\ t} \cdot \boldsymbol{f}_{12}^{a} \end{aligned}$$

D'autre part, notons que la puissance développée par une force dont le point d'application ne varie pas (vitesse relative nulle au point d'application) développe une puissance nulle.

Puissance des quantités d'accélérations

La puissance des quantités d'accélérations d'un solide S est égale au produit de sa quantité d'accélération par sa vitesse absolue:

$$P_S^{acc} = \boldsymbol{v}_{S0}^{p\ t} \cdot \ddot{\boldsymbol{d}}_{S0}^{p} \tag{2.29}$$

Ici aussi, la puissance est indépendante du point d'expression.

Energie

L'énergie E développée par une action dynamique ou par une quantité d'accélération pendant un intervalle de temps $t_2 - t_1$ est simplement l'intégrale de sa puissance P sur cet intervalle de temps:

$$E(t_2 - t_1) = \int_{t_1}^{t_2} P dt \tag{2.30}$$

L'unité d'énergie est le *Joule*(J), ou $kg.m^2.s^{-2}$. La plupart des principes physiques couramment utilisés qui font intervenir la puissance sont vrais à tout instant. On peut donc les intégrer au cours du temps et obtenir des relations équivalentes sur l'énergie. C'est pourquoi ces deux termes, puissance et énergie, sont souvent utilisés indifféremment.

Théorème de la puissance (ou de l'énergie)

Pour tout système matériel, la puissance des quantités d'accélérations est égale à la somme des puissances des efforts extérieurs plus la somme des puissances des efforts intérieurs.

$$P^{int} + P^{ext} = P^{acc} \tag{2.31}$$

Si le système matériel est constitué d'un solide unique, la puissance des efforts intérieurs est nulle. Si le système est constitué d'un ensemble de solides, la puissance des efforts intérieurs rassemble la puissance des interactions entre les solides.

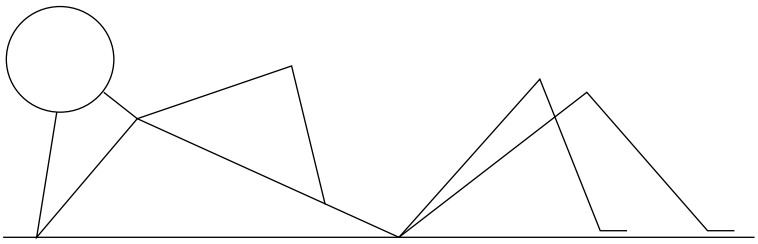


FIG. 2.4 - *Une structure articulée, au repos.*

2.3 Structures articulées

Les structures articulées sont des assemblages de solides possédant certaines mobilités. La figure (2.4) en motre un exemple.

Les structures articulées sont composées de solides reliés par des articulations (ou liaisons). Ces liaisons contraignent les mouvements relatifs des solides qu'elles relient. Par exemple, seule une rotation est possible entre le bras et l'avant-bras du personnage de la figure (2.4). De plus, les extrêmités de ces solides doivent rester confondus au cours du temps.

2.3.1 Articulations et contraintes cinématiques

Les contraintes cinématiques restreignent les mouvements possibles d'un repère par rapport à un autre repère. Nous utilisons neuf valeurs pour définir le mouvement d'un repère par rapport à un autre (une vitesse angulaire, une vitesse de translation, et le point d'expression de cette dernière). Parmi ces neuf, six seulement sont indépendantes de par la forme du champ des vitesses d'un solide (eq. 2.7 et 2.12), correspondant à trois translations et trois rotations indépendantes. On dit que le mouvement relatif de deux solides comprend au maximum six *degrés de liberté*. Une *liaison* ou *articulation* entre deux solides est constituée d'une ou plusieurs contraintes cinématiques entre les deux repères. La figure 2.5 représente deux solides reliés par une articulation. Ici, la contrainte est que les points $\boldsymbol{L}$ de $\mathcal{R}_1$ et $\boldsymbol{L'}$ de $\mathcal{R}_2$ restent confondus au cours du temps.

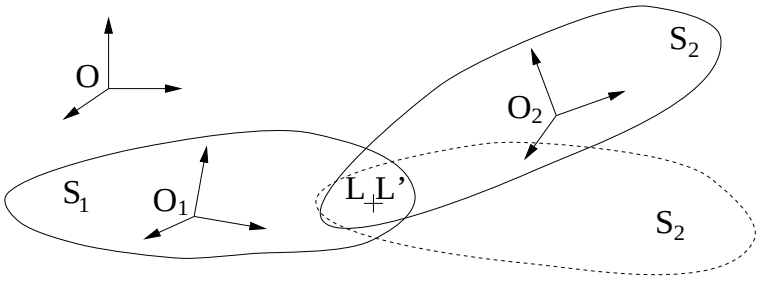


FIG. 2.5 - *Une liaison point-point.*

En trois dimensions, ce type d'articulation est communément appelé *liaison rotule*. En deux dimensions, c'est une *liaison pivot*. Sa contrainte cinématique peut s'exprimer d'un point de vue géométrique par la relation: $\boldsymbol{OO_1 + O_1L = OO_2 + O_2L'}$, où le point O est un point quelconque. Ceci implique que les point $\boldsymbol{L}$ et $\boldsymbol{L'}$ sont confondus et que

cette relation devant rester satisfaite au cours du temps, on peut la dériver pour obtenir l'équation aux vitesses: $\boldsymbol{V_{21}^L} = \mathbf{0}$ et aux accélérations: $\boldsymbol{\Gamma_{21}^L} = \mathbf{0}$ Chacune de ces équations vectorielles se décompose en trois équations scalaires, qui enlèvent donc trois degrés de liberté dans le mouvement relatif des deux solides.

Les contraintes peuvent aussi porter sur les rotations relatives, ou sur une combinaison de rotations et translations. La liaison pivot, représentée sur la figure 2.6, n'autorise en tout qu'une rotation relative, selon une direction appelée axe du pivot. Les relations

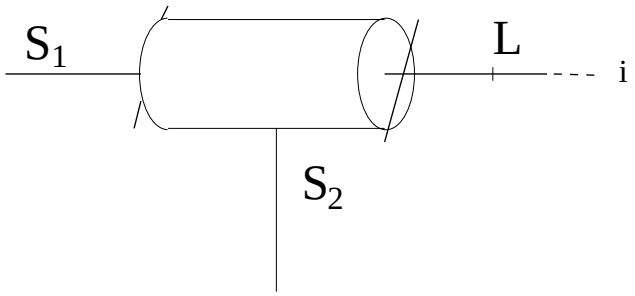


FIG. 2.6 - *Liaison pivot d'axe $\boldsymbol{i}$ entre S_1 et S_2. Le centre de liaison peut être placé n'importe où sur l'axe.*

cinématiques correspondantes sont:

$$\boldsymbol{V_{21}^L} = \mathbf{0}$$
$$\boldsymbol{\Omega_{21}} = \omega \boldsymbol{i}$$

où ω est un scalaire quelconque. Une formulation scalaire permet d'isoler cinq contraintes cinématiques, ne laissant qu'un seul degré de liberté pour la liaison pivot:

$$\boldsymbol{V_{21}^L} \cdot \boldsymbol{i} = 0$$
$$\boldsymbol{V_{21}^L} \cdot \boldsymbol{j} = 0$$
$$\boldsymbol{V_{21}^L} \cdot \boldsymbol{k} = 0$$
$$\boldsymbol{\Omega_{21}} \cdot \boldsymbol{j} = 0$$
$$\boldsymbol{\Omega_{21}} \cdot \boldsymbol{k} = 0$$

où les vecteurs $\boldsymbol{j}$ et $\boldsymbol{k}$ sont deux vecteurs indépendants orthogonaux à la direction de l'axe.

Tout mouvement relatif de deux solides peut se représenter à l'aide d'une ou plusieurs liaisons. Nous reviendrons sur les différents types de liaisons au chapitre 3.

2.3.2 Graphe cinématique

Le *graphe cinématique* est un outil utile pour représenter un système articulé (figure 2.7).

Les noeuds sont les solides, notés par des entiers, et les arcs sont les articulations, notées par des couples d'entiers correspondant aux solides concernés. Le personnage de la figure (2.7) est représenté par un graphe acyclique. On remarque l'existence d'une articulation entre le repère absolu et le tronc du personnage. Elle est nécessaire pour connaître la position absolue des solides du graphe. Cette articulation ne réduit pas nécessairement les

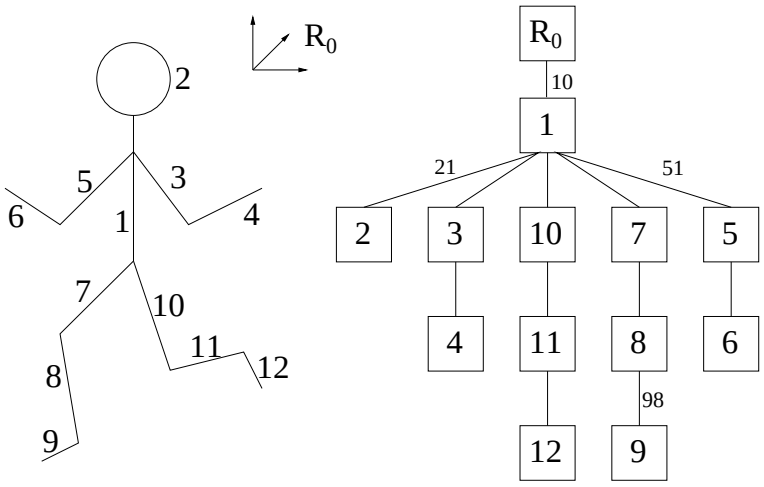


FIG. 2.7 - *Un personnage et un graphe cinématique associé*

mouvements possibles du bonhomme car elle peut comporter six degrés de liberté. Notons que cinématiquement, le repère absolu n'est rien de plus qu'un solide particulier dont on conservera une référence pour accéder à volonté au repère absolu. On peut utiliser ce graphe pour calculer récursivement toutes les positions absolues des solides. En parcourant le graphe cinématique à partir du repère absolu, on peut déterminer récursivement les positions absolues de chaque solide en fonction de son supérieur (son "père") dans la hiérarchie par simple changement de repère. Ceci impose des contraintes:

- tout solide doit avoir un père, excepté celui qui tient le rôle de repère absolu. Dans le cas où un solide possède plusieurs pères, on peut calculer sa position par rapport à n'importe lequel d'entre eux. Il faut alors pouvoir garantir l'unicité de sa position en fonction du père choisi pour référence. Ceci est le problème de toutes les chaînes fermées, qui correspondent à des graphes cinématiques non arborescents, et qui sont étudiés plus en détail dans le chapitre 2.4.

- le graphe doit être acyclique afin d'éviter tout bouclage dans la récursion. Cela est possible pour toute structure, même comprenant des chaînes fermées, si l'on utilise des arcs orientés. Ceci justifie la distinction entre le père et le fils d'une liaison.

On peut noter que différents graphes cinématiques peuvent être utilisés pour représenter la même structure. Dans le cas où la liaison entre le personnage de la figure 2.7 et l'univers serait totalement libre, on pourrait tout aussi bien mettre la tête ou tout autre membre à la racine du graphe, pourvu que cette racine soit à son tour reliée à l'univers. Cette variabilité du graphe cinématique suggère qu'il n'est qu'une convention de parcours de la structure et non un paramètre du système.

2.3.3 Systèmes de coordonnées

Nous avons vu que le mouvement, et donc la position, d'un solide par rapport à un autre est composé de six degrés de liberté. Parmi tous les repères de référence, on peut en choisir un considéré immobile et qu'on appelle donc repère absolu.

Utiliser les coordonnées absolues consiste à décrire le mouvement de chaque solide par six fonctions du temps. Pour n solides, on a donc $6n$ fonctions du temps. Ce paramétrage a l'avantage de la simplicité.

Toutefois, nous avons aussi vu que les articulations entre les solides réduisent le nombre de degrés de liberté dans le mouvement relatif des repères liés par l'articulation, et il peut être intéressant de tenir compte de cette réduction pour le paramétrage global du système. En effet, un nombre réduit de paramètres conduit à des systèmes d'équations plus petits et, espérons-le, à une résolution accélérée. La figure 2.8 montre un exemple de coordonnées relatives. On s'est placé dans un cas bidimensionnel, avec des liaisons pivot entre les solides.

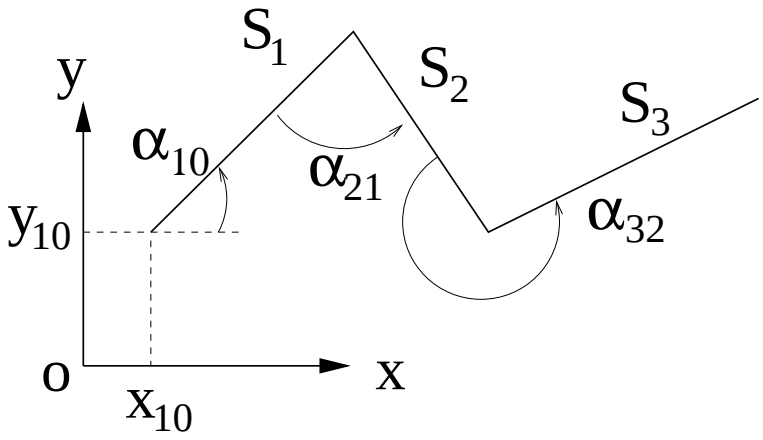


FIG. 2.8 - *Trois solides et un paramétrage relatif*

Le jeu de paramètres nécessaire pour décrire la position des trois solides se réduit à $x_{10}, y_{10}, \alpha_{10}, \alpha_{21}, \alpha_{32}$, ce qui représente plusieurs avantages:

- Une économie de paramètres par rapport aux neuf nécessaires (trois par solide dans le plan);
- Une prise en compte implicite des contraintes.

Notons que le paramétrage en coordonnées absolues peut être considéré comme un cas particulier du paramétrage en coordonnées relatives. Dans ce cas, les solides comportent tous une articulation à six degrés de liberté par rapport au repère absolu. Sur la figure (2.9), les vitesses absolues des solides sont les vitesses articulaires de leurs liaisons avec le repère absolu.

2.3.4 Chaînes fermées

Chaque articulation peut être paramétrée par un nombre de variables égal au nombre de degrés de liberté autorisés par la liaison.

Dans le cas d'un graphe acyclique tel que celui représenté sur la figure 2.7, le nombre de degrés de liberté indépendants est égal à la somme des degrés de liberté de chaque articulation. Par exemple, trois pour le cou, deux pour les hanches et les épaules, un pour les genoux et les coudes. Le bonhomme est de plus positionné par rapport à son monde environnant, une liaison supplémentaire entre un repère extérieur et le tronc (solide 1)

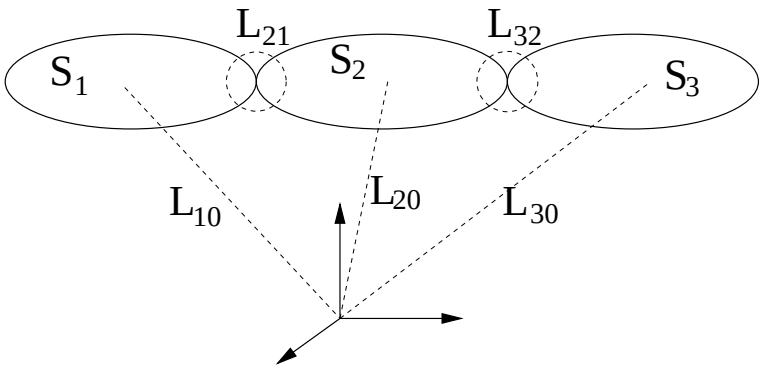


FIG. 2.9 - *Des liaisons à six degrés de liberté* $(x, y, z, \alpha, \beta, \gamma)$ *par rapport au repère absolu définissent un paramétrage absolu des solides.*

est donc nécessaire, comportant six degrés de liberté. Chaque paramètre d'articulation peut avoir une valeur quelconque, sans violer aucune contrainte cinématique. Bien sûr, ceci n'est vrai que si l'on ne tient pas compte d'éventuelles collisions.

Dans le cas d'une chaîne fermée, le graphe n'est plus acyclique, comme l'illustre la figure 2.10.

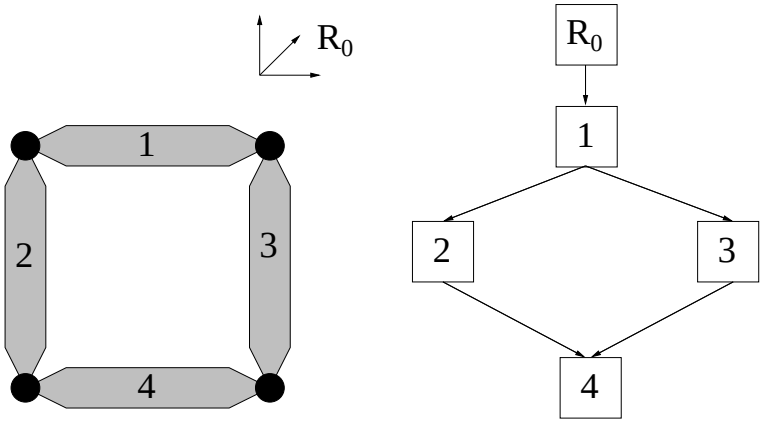


FIG. 2.10 - *Une chaîne fermée et son graphe cinématique. Les disques noirs représentent des articulations.*

Plusieurs graphes cinématiques peuvent convenir pour représenter une même structure articulée, mais tous ont forcément le même nombre de chaînes fermées.

L'exemple de la figure 2.10 soulève une question: la position absolue du solide 4 doit-elle être calculée par rapport à celle de 2 ou de 3? L'état du système sera dit *consistant* si cela ne fait pas de différence. Dans le cas contraire, la chaîne fermée est brisée, et une au moins de ses branches nécessite une correction, comme illustré sur la figure 2.11.

Le problème des vitesses et accélérations est similaire à celui des positions: les vitesses et accélérations des solides calculées d'après les données articulaires ne doivent pas dépendre du graphe cinématique choisi.

Nous dirons qu'un système est *résolu en position* si la consistance des positions est obtenue, *résolu en vitesse* si la consistance des vitesses est obtenue, et *résolu en accélération* si la consistance des accélérations est obtenue. Le chapitre suivant présente les équations permettant de représenter toutes ces contraintes, ainsi que des méthodes de résolution.

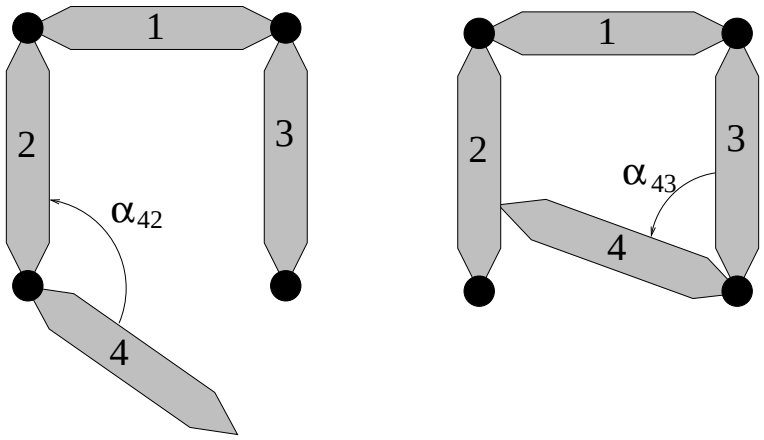


FIG. 2.11 - *Système inconsistant. L'état du système varie avec le graphe cinématique choisi.*

2.3.5 Représentation du graphe cinématique

Nous avons vu les graphe cinématique sous forme d'un ensemble de noeuds, les solides, reliés par un ensemble d'arcs, les liaisons. Or les liaisons contiennent un certain nombre d'informations et ne se réduisent donc pas à de simples pointeurs de solides. La représentation des graphes cinématiques présentée sur la figure (2.12) comporte deux types de noeuds, solides et articulations, qui contiennent toutes les données cinématiques. Dans ce formalisme, les arcs sont bien réduits à de simples pointeurs. Toutes ce qui a été dit sur les graphes cinématiques reste vrai dans ce nouveau formalisme, que nous adoptons pour la suite. Le graphe cinématique d'une structure sera représenté par les notions de *père, fils, mère, fille*. On fait le choix arbitraire de placer systématiquement le repère absolu en haut de la figure représentant le graphe.

Notons que les liaisons ont toujours un père et un fils, et que les solides ont toujours au moins une mère. Les ramifications du graphe cinématique se concrétisent par des solides ayant plusieurs filles, et les chaînes fermées par des solides ayant plus d'une mère.

2.4 Équations de mouvement

2.4.1 Équations géométriques

Soit une chaîne fermée simple, composée de trois solides devant former un triangle. Plaçons-nous dans le cas bidimensionnel pour simplifier. La figure 2.13 nous montre cette chaîne fermée ainsi qu'un jeu de paramètres possible.

La flèche pointillée sur le graphe cinématique indique que le solide 3 est mis en position par rapport au solide 2 et non par rapport au solide 1. La contrainte à exprimer est donc que les solides 1 et 3 doivent coïncider au point C. Considérons que le point C fait partie du solide 1, et qu'il doit se confondre avec le point C' du solide 3. La relation d'égalité entre C et C' est:

$$\boldsymbol{AB} + \boldsymbol{BC'} = \boldsymbol{AC} \tag{2.32}$$

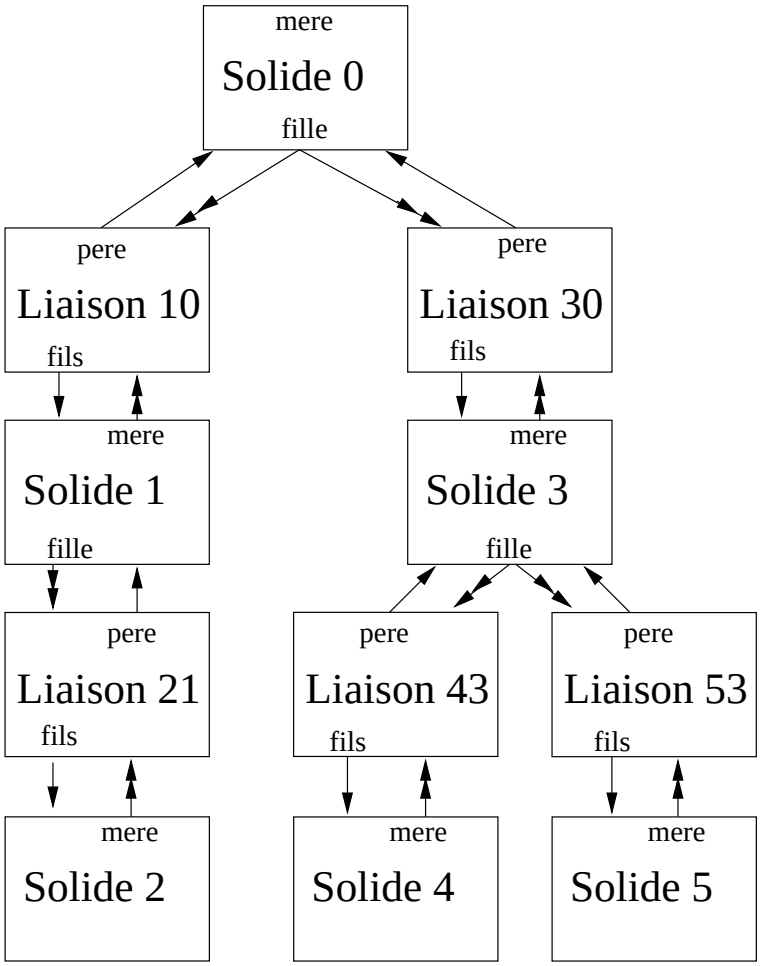


FIG. 2.12 - *Un exemple du formalisme que nous utiliserons pour les graphes cinématiques. Les flèches simples représentent des pointeurs, les doubles des listes de pointeurs. Un solide peut avoir plusieurs mères en cas de chaînes fermées. Le solide* 0*, repère absolu, n'a pas de mère.*

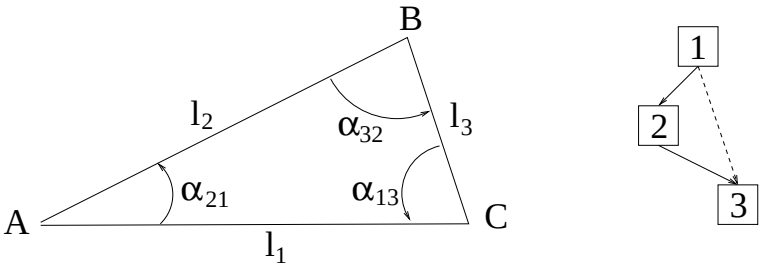


FIG. 2.13 - *Une chaîne fermée simple avec ses paramètres et un graphe cinématique.*

Cette relation peut se traduire par les équations suivantes:

$$l_2 . \cos \alpha_{21} + l_3 \cos(\pi - \alpha_{21} - \alpha_{32}) = l_1$$
$$l_2 . \sin \alpha_{21} + l_3 \sin(\pi - \alpha_{21} - \alpha_{32}) = 0$$

Notons que l'angle α_{13} n'intervient pas dans les équations de fermeture de chaîne. Il peut être déduit *a posteriori* des deux autres. Ces équations ne sont pas linéaires ni même polynômiales. Il est possible dans certains cas d'effectuer une résolution analytique de ce genre d'équations au moyen de calculs formels hélas coûteux[Coz96]. Des solutions consistant à précalculer ces solutions ont été proposées, qui aboutissent à des programmes efficaces d'animation de structures articulées. Leur inconvénient est de ne s'appliquer qu'à des liaisons permanentes et bien définies, sans pouvoir prendre en compte de nouvelles contraintes susceptibles d'apparaître en cours d'animation, comme par exemple les contacts surfaciques qui se créent lors des collisions entre objets. Toutefois, nous verrons qu'il est possible de trouver une solution satisfaisant les équations de position par des résolutions linéaires itératives.

2.4.2 Équations cinématiques

Les équations cinématiques interviennent pour satisfaire les contraintes de vitesses ou d'accélération aux fermetures de boucles. Ces contraintes sont d'une part la nullité des mouvements relatifs interdits, d'autre part la valeur correcte des éventuels mouvements relatifs imposés. La figure (2.14) montre un mécanisme à une chaîne fermée.

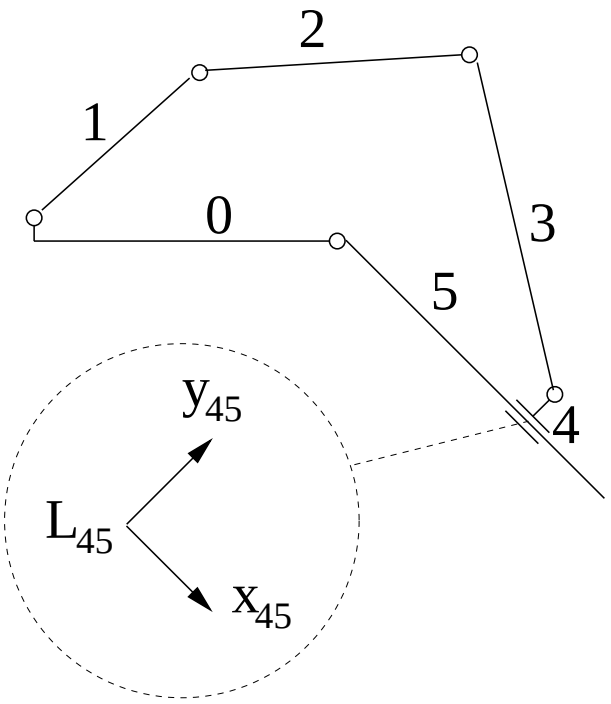


FIG. 2.14 - *Une structure à une chaîne fermée*

On choisit de placer la fermeture en L_{45}. La contrainte de mouvement interdit est $\dot{y}_{45} = 0$. On pourrait vouloir imposer une contrainte de mouvement relatif à la fermeture de boucle, de type $\dot{x}_{45} = v$.

Une contrainte de type $\dot{\alpha}_{21} = c$ ne génère pas d'équation spécifique. On la prend en compte en retirant $\dot{\alpha}_{21}$ de l'ensemble des inconnues et en prenant en compte son influence lors de l'écriture des équations. Ceci est le principal avantage des coordonnées relatives.

Les équations de vitesse sont donc:

$$\boldsymbol{V}_{45}^{L_{45}} \cdot \boldsymbol{y}_{45} = 0$$
$$\boldsymbol{V}_{45}^{L_{45}} \cdot \boldsymbol{x}_{45} = v$$

Les équations d'accélération s'obtiennent en dérivant les équations de vitesse. Le repère de dérivation est celui dans lequel sont définies les directions de mouvement contraint, ici, 4 ou 5. Cette dérivation sera explicitée plus en détail dans le chapitre 3. On y verra aussi que les équations de vitesse et d'accélération sont linéaires en fonction des inconnues cinématiques (vitesses ou accélérations articulaires).

2.4.3 Mise en oeuvre de l'animation par modèles physiques

Nous avons vu dans ce chapitre que l'état d'un système de solides peut se représenter par un ensemble de matrices et vecteurs à six dimensions. Deux catégories de contraintes doivent être satisfaites:

- les contraintes cinématiques, qui se traduisent par des équations linéaires sur les vitesses et accélérations, mais non linéaires sur les positions.

- Les contraintes dynamiques, résumées par le principe fondamental de la dynamique, qui constituent la spécificité de l'animation par modèles physiques. Elles lient les dérivées seconde et première des positions avec les forces, et sont linéaires selon les dérivées secondes (accélérations).

Le fait que la dynamique concerne les dérivées des positions, et qu'on dispose de méthodes de résolution d'équations linéaires relativement souples et efficaces incite à traiter les contraintes de mouvement au niveau des vitesses et accélérations. L'intégration au cours du temps des solutions trouvées est alors nécessaire pour connaître l'évolution des solides.

Chapitre 3

Modélisation cinématique de scènes

La modélisation cinématique de scènes consiste à définir les positions relatives des solides et comment ces positions sont susceptibles d'évoluer. Les outils utilisés pour cela sont les liaisons, ou articulations. La fonction des articulations est triple:

- permettre à un utilisateur de définir des structures articulées représentant les personnages et autres éléments de la scène à animer;
- évoluer au cours du temps pour permettre aux objets de bouger; prendre en compte des contraintes survenant au cours de l'animation comme l'apparition ou la disparition de contacts entre objets;
- fournir des outils nécessaires à l'écriture des équations de mouvement.

Nous allons dans ce chapitre présenter le modèle d'articulations que nous avons implémenté et utilisé pour tout le travail présenté dans ce document. Nous nous sommes initialement intéressés à une formulation en coordonnées relatives. C'est pourquoi seront présentés des algorithmes permettant l'écriture des équations cinématiques et dynamiques en coordonnées relatives, ainsi qu'une formulation originale des contacts extérieurs compatible avec le paramétrage en coordonnées relatives. Les coordonnées absolues n'étant qu'un cas particulier des coordonnées relatives, les travaux présentés s'y appliquent aussi.

3.1 Spécification des liaisons

La figure (3.1) représente un balancier articulé à son support par une liaison pivot, ainsi que certains éléments de la représentation géométrique de la scène.

La spécification de l'articulation passe par la donnée de plusieurs éléments:

- le type d'articulation (ici une liaison pivot);
- le père et le fils de l'articulation (ici S_1 et S_2);
- le centre de liaison (ici $\boldsymbol{L}_{12}$);
- le centre de liaison par rapport au fils;
- l'axe de la liaison par rapport au père;

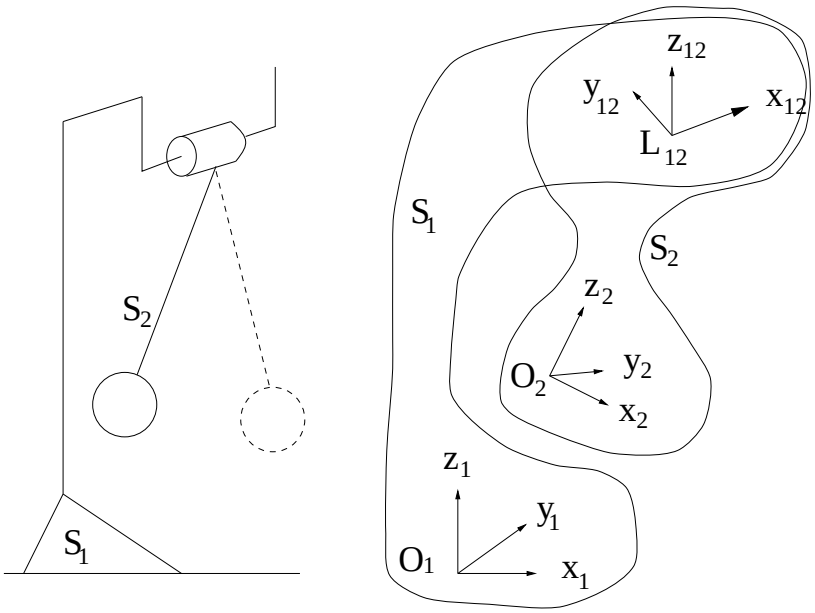


FIG. 3.1 - *Une structure cinématique*

– l'axe de liaison par rapport au fils;

– la position initiale de l'articulation (ici, la rotation initale).

Notons que la donnée d'un axe n'est pas toujours nécessaire car certaines articulations n'en comportent pas. Le tableau (3.2) présente un certain nombre de liaisons classiques. Les liaisons possèdent au plus une direction privilégiée, que nous appellerons son axe. On peut trouver plus de précisions sur la notion d'axe de liaison dans l'annexe (D).

Un repère intermédiaire a été construit autour de l'axe de la liaison. Il sera utile pour représenter les directions orthogonales à l'axe lors de l'écriture des équations.

3.2 Repère intermédiaire de liaison

Nous avons vu qu'un mouvement relatif entre solides est soit une translation, auquel cas le mouvement est le même en tout point de l'espace, soit un vissage autour d'une droite appelée axe instantané de rotation. Cet axe peut varier au cours du temps, comme dans le cas d'un appui plan, d'une rotule ou d'un mouvement libre, ou bien rester fixe comme dans le cas d'une liaison pivot ou pivot glissant. La figure 3.3 illustre l'utilisation d'un repère intermédiaire de liaison.

Ce repère sera orienté selon les directions privilégiées de la liaison. La position du solide i par rapport au repère absolu sera définie par la matrice $\boldsymbol{P}_{i0}$. Nous utilisons ici le formalisme des coordonnées homogènes pour représenter les passages d'un repère à l'autre par des matrices de taille $4,4$. Un rappel sur les coordonnées homogènes est fourni en annexe (C). La matrice de position du repère j par rapport au repère local de i sera notée $\boldsymbol{P}_{ji}$. La matrice de passage entre le repère j et le repère absolu pourra donc être calculée par la relation: $\boldsymbol{P}_{j0} = \boldsymbol{P}_{ji} \cdot \boldsymbol{P}_{i0}$, où le point représente le produit matriciel. On peut noter la relation de Chasles sur les indices des matrices.

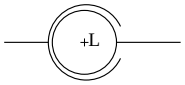
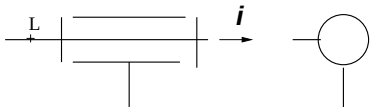
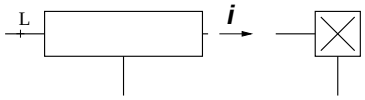
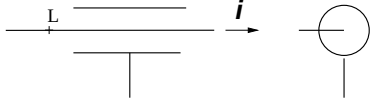
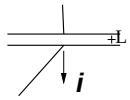
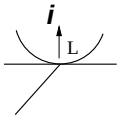
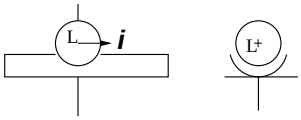
Liaison	Schematisation	Cinematique
Rotule		Ω $V_L = 0$
Pivot		$\Omega = \dot{\alpha}.i$ $V_L = 0$
Glissiere		$\Omega = 0$ $V_L = \dot{x}.i$
Pivot glissant		$\Omega = \dot{\alpha}.i$ $V_L = \dot{x}.i$
Appui plan		$\Omega = \dot{\alpha}.i$ $V_L = \dot{y}.j + \dot{z}.k$
Contact ponctuel		Ω $V_L = \dot{y}.j + \dot{z}.k$
Lineaire annulaire		Ω $V_L = \dot{x}.i$

FIG. 3.2 - *Principales liaisons technologiques. La liaison linéaire annulaire pourrait aussi être appelée "rotule glissante"ou "sphère-gouttière".*

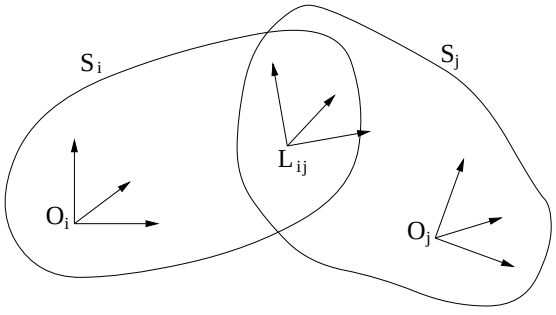


FIG. 3.3 - *Repère intermédiaire de liaison*

Soit $\boldsymbol{P}_{li}$ la matrice définissant le repère intermédiaire de liaison dans le solide i. De même,$\boldsymbol{P}_{lj}$ définit la position du repère intermédiaire de liaison dans le repère j. Le calcul de la matrice de passage de S_j à S_i s'effectue par un produit matriciel:

$$\boldsymbol{P}_{ji} = \boldsymbol{P}_{lj}^{-1} \cdot \boldsymbol{P}_{li} \tag{3.1}$$

Le repère intermédiaire de liaison permet de mettre en relation vitesse articulaire et vitesse relative des solides. On peut utiliser les axes du repère intermédiaire de liaison comme base orthonormale pour exprimer les trois translations et les trois rotations indépendantes. Si de plus, certains de ces axes correspondent à des directions privilégiées de l'articulation, les vitesses indépendantes correspondront aux vitesses articulaires, ou bien seront nulles. Utilisons l'exemple de l'appui plan illustré sur la figure 3.4. Nous pouvons utiliser un repère intermédiaire de liaison dont un des axes est orienté orthogonalement au plan. La figure 3.4 représente un repère intermédiaire possible pour une liaison appui plan.

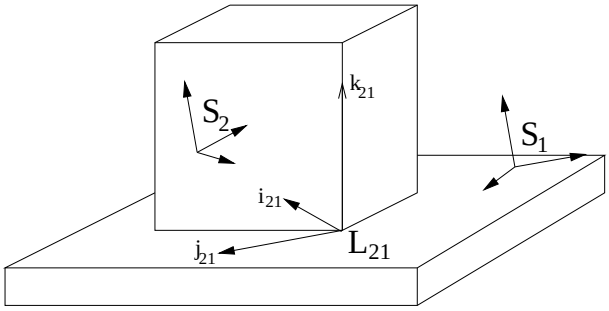


FIG. 3.4 - *Repère intermédiaire de liaison pour un appui plan*

Nous avons choisi d'orienter l'axe $\boldsymbol{k}$ du repère intermédiaire de liaison selon la normale au plan. La vitesse relative $\boldsymbol{v}_{21}$ de S_2 par rapport à S_1 peut donc s'exprimer en L_{21} par:

$$\boldsymbol{v}_{21}^{L_{21}} = \begin{pmatrix} vi_{21}.\boldsymbol{i}_{21} + vj_{21}.\boldsymbol{j}_{21} \\ \omega_{21}.\boldsymbol{k}_{21} \end{pmatrix}$$

Les valeurs ω_{21}, vi_{21} et vj_{21} sont les vitesses articulaires de la liaison.

3.3 Mise à jour des positions relatives

3.3.1 Calcul du nouveau repère intermédiaire

La mise à jour des positions relatives s'effectue en modifiant les matrices $\boldsymbol{P}_{li}$ et/ou $\boldsymbol{P}_{lj}$, ce qui laisse généralement le choix entre plusieurs possibilités. Pour l'exemple de la figure 3.4, on peut indifféremment modifier $\boldsymbol{P}_{li}$, $\boldsymbol{P}_{lj}$, ou les deux en appliquant par exemple la translation sur $\boldsymbol{P}_{li}$ et la rotation sur $\boldsymbol{P}_{lj}$. Concrètement, le choix a été fait de modifier uniquement $\boldsymbol{P}_{li}$ dans la mesure du possible. Ceci évite de recalculer $\boldsymbol{P}_{lj}^{-1}$ (eq. 3.1). On verra plus loin que le contact extérieur nécessite quand à lui de modifier conjointement $\boldsymbol{P}_{li}$ et $\boldsymbol{P}_{lj}$.

3.3.2 Prise en compte du temps

Suivant les cas, on pourra calculer les nouvelles matrices à partir d'une position de référence initiale correspondant à $t = 0$ par exemple, où t représente le temps, ou alors d'un instant sur l'autre entre t et $t + \delta t$. Dans le premier cas, il est nécessaire de conserver en mémoire les matrices de passage initiales. Ce choix correspond, dans l'exemple de l'appui plan où nous choisirions de porter toute la modification sur $\boldsymbol{P}_{li}$, à utiliser l'une des deux relations suivantes:

$$\boldsymbol{P}_{li}(t) = \boldsymbol{P}_{li}(t_0) \cdot \boldsymbol{C}(x(t), \alpha(t))$$
$$\boldsymbol{P}_{li}(t) = \boldsymbol{P}_{li}(t - \delta t) \cdot \boldsymbol{C}(\dot{x}, \dot{\alpha}, dt)$$

où x et α désignent la translation et la rotation, et $\boldsymbol{C}$ est une matrice homogène de passage comme expliqué en annexe C.

3.3.3 Le problème des rotations

S'il est trivial d'intégrer et combiner des mouvements de translation, il n'en va pas de même pour les rotations. Les vitesses de rotation, représentées par des vecteurs, possèdent leurs caractéristiques sympathiques, notamment de linéarité:

$$\left.\begin{array}{l} \boldsymbol{\Omega}_1 \cdot \boldsymbol{u} = 0 \\ \boldsymbol{\Omega}_2 \cdot \boldsymbol{u} = 0 \end{array}\right\} \Rightarrow (\boldsymbol{\Omega}_1 + \boldsymbol{\Omega}_2) \cdot \boldsymbol{u} = 0 \tag{3.2}$$

Cette propriété se perd au niveau des déplacements de rotation. Posons un livre sur une table rectangulaire. Appliquons-lui une rotation d'angle π (un demi-tour) selon un axe parallèle à un bord de la table. Recommençons l'opération en tournant cette fois selon l'autre bord (perpendiculaire au premier) de la table. On constate que pour retrouver la position initiale du livre, il suffit maintenant d'effectuer un demi-tour selon un axe *perpendiculaire à la table*. La propriété (3.2) n'a pas son équivalent au niveau des déplacements. Les rotations ne peuvent se composer par addition. En particulier, l'ordre des rotations successives joue un rôle.

Quel sens a donc une liaison n'autorisant que deux rotations relatives indépendantes? Considérons à nouveau l'exemple du livre, et décrétons que la rotation autour de l'axe perpendiculaire à la table est la seule interdite. Nous avons vu que dans ce cas, la composition de deux rotations autorisées peut générer une rotation résultante interdite. D'autre part, imposer un ordre sur les rotations à effectuer relève de l'arbitraire et n'est donc pas satisfaisant d'un point de vue physique. Les rotations ordonnées, commes celles qui entrent en jeu dans les angles d'Euler (fig. 3.5) devront donc, dans notre modèle, être décomposées en utilisant plusieurs articulations faisant intervenir des solides intermédiaires.

La décomposition en rotations successives est d'ailleurs la solution utilisée dans les mécanismes réels à plusieurs rotations (cardans, bras manipulateurs...). Le seul cas de liaison simple à deux rotations connu de l'auteur est la liaison linéaire rectiligne, dite aussi liaison couteau, que l'on obtient par exemple en posant la tranche du livre sur la table. Dans ce cas, deux contacts ponctuels suffisent à modéliser le même comportement, comme illustré sur la figure 3.6.

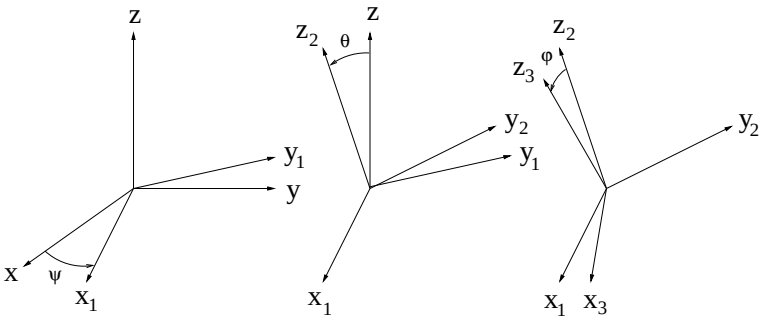


FIG. 3.5 - *Les angles d'Euler correspondent à des rotations successives*

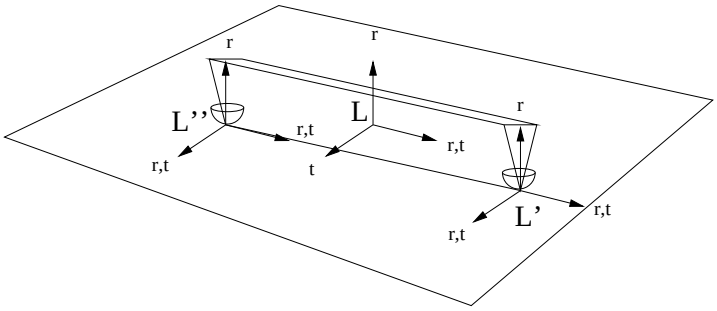


FIG. 3.6 - *Une liaison linéaire rectiligne* L *peut être modélisée par deux contacts ponctuels* L' *et* L"*. Les lettres* r *et* t *indiquent les axes de rotations et translations relatives autorisées.*

3.3.4 Directions indépendantes

Les directions indépendantes d'une liaison représentent les six mouvements relatifs indépendants (trois translations, trois rotations) au centre de la liaison. Une direction indépendante contient principalement un vecteur à six dimensions qu'on appelle son axe, une force notée f et des variables de mouvement notées q, $\dot{q}$ et $\ddot{q}$. Parmi les directions indépendantes nous distinguerons les mobilités, qui sont les directions indépendantes selon lesquelles le mouvement relatif est possible, et les blocages, selon lesquels les mouvements sont interdits. Les blocages et les mobilités forment deux ensembles complémentaires, la somme du nombre de blocages et de mobilités vaut six.

Par convention, le mouvement selon une direction indépendante représente le mouvement du fils de la liaison par rapport au père. Inversement, l'effort exercé représente l'effort du père sur le fils.

Par souci d'efficacité, on choisit d'orienter les axes des directions indépendantes selon les axes du repère intermédiaire de liaison. On dispose donc de six mobilités possibles ti, tj, tk, ri, rj, rk correspondant aux translations et rotations relatives selon les axes $\boldsymbol{i}$, $\boldsymbol{j}$ et $\boldsymbol{k}$ du repère de liaison. L'axe privilégié de la liaison, s'il existe, est par convention l'axe $\boldsymbol{i}$ du repère intermédiaire. La matrice ci-dessous est composée de six vecteurs représentant

les axes des directions indépendantes:

$$\begin{pmatrix} ti \\ tj \\ tk \\ ri \\ rj \\ rk \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Par la suite, nous utiliserons les notations suivantes:

- A_{jk} représente l'articulation entre le solide j (le fils) et le solide k (le père). Quand plusieurs articulations existeront entre deux mêmes solides, une notation *ad hoc* sera utilisée.

- L_{jk} est le centre de la liaison A_{jk}.

- $\boldsymbol{a}_l$ est l'axe de la mobilité l. C'est un vecteur à six composantes de type $(\boldsymbol{u}^t, \mathbf{0}^t)^t$ ou $(\mathbf{0}^t, \boldsymbol{u}^t)^t$ selon qu'il s'agit, respectivement, d'une mobilité de translation ou de rotation. Le vecteur $\boldsymbol{u}$ est un des axes du repère intermédiaire de liaison.

Selon ce formalisme, une liaison pivot est une articulation possédant uniquement ri dans sa liste de mobilités. Une rotule contient ri, rj, rk, un appui plan contient tj, tk, ri, un contact extérieur contient tj, tk, ri, rj, rk, etc. On peut ainsi définir toutes les liaisons technologiques, et aussi d'autres impossibles à réaliser directement en réalité comme par exemple une translation 3D qui contiendrait ti, tj, tk. L'intérêt de ce genre de liaisons abstraites est de rassembler dans un même formalisme articulations et contraintes cinématiques diverses.

3.3.5 Généricité

On désire pouvoir modéliser un grand nombre d'articulations différentes. La figure 3.2 représente les liaisons simples les plus couramment utilisées en technologie. Les liaisons différant principalement par les mouvements relatifs autorisés, on peut définir une classe générique d'articulations dont les sous-classes comportent différentes combinaisons de mobilités et de blocages.

La classe de base doit posséder un pointeur sur chacun des solides reliés par l'articulation. Tout solide a au moins une mère. Toutes les mères supplémentaires sont des fermetures de chaînes. Le point d'expression des contraintes doit être précisément défini dans le père et dans le fils afin de positionner le repère intermédiaire de liaison. De plus, un certain nombre de liaisons comportent une direction privilégiée que nous appellerons *axe*. Cette notion est plus développée dans l'annexe D. Cette direction privilégiée, définie dans chacun des repères, doit alors aussi faire partie des champs de l'articulation. Ainsi la spécification d'une liaison se fera par la donnée de son père et son fils, le centre de liaison et, si besoin, l'axe dans chacun des deux solides. Les sous-classes se distinguent par leurs ensembles de mobilités et de blocages ainsi que par les procédures de mise à jour des matrices définissant le repère intermédiaire de liaison.

3.4 Le problème du contact extérieur en coordonnées relatives

Nous désirons dans une scène complexe représenter aussi bien des objets en mouvement libre, par exemple la chute d'un corps, que des mouvements fortement contraints comme celui du squelette d'un personnage, mais aussi des surfaces qui roulent et glissent les unes sur les autres. En coordonnées relatives, la mise à jour des matrices de passage nécessite dans ce cas un soin tout particulier sans quoi des décollements ou pénétrations peuvent se produire, comme l'illustre la figure 3.7.

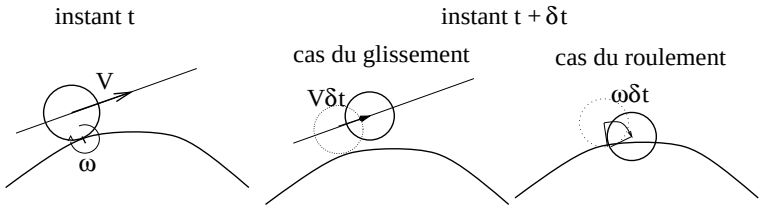


FIG. 3.7 - *Le problème de l'intégration des vitesses*

Dans un modèle simpliste de coordonnées articulaires, les axes sont considérés fixes dans les repères, alors que dans le cas du roulement relatif de deux solides, ou du glissement de deux surfaces non planes, les axes varient au cours du temps : deux solides dans cette configuration ne constituent pas un mécanisme. Il est important de réaliser qu'il ne s'agit pas là d'une simple erreur d'intégration numérique mais d'une véritable lacune du modèle présenté.

3.4.1 Générer la cinématique du contact

On pourrait générer la cinématique du contact extérieur par combinaison de mécanismes virtuels comme sur la figure 3.8. Le roulement sans glissement s'obtient par un engrenage, le glissement pur est une translation qu'on peut générer par un parallélogramme déformable.

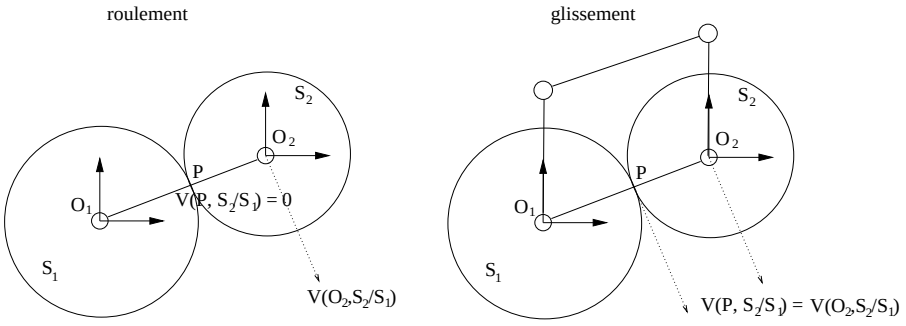


FIG. 3.8 - *Mécanismes virtuels et contraintes nécessaires pour simuler le roulement et le glissement*

La mise en place de mécanismes virtuels a l'inconvénient de rajouter au système des inconnues et des équations de fermetures cinématiques de chaînes (équations de vitesse).

Cela a pour effet d'alourdir le temps de calcul de la scène. De plus, ces mécanismes sont définis par rapport aux centres de courbures des surfaces, qui ne sont pas constants dans le cas général de surfaces quelconques. D'où des frais de remise à jour de la structure qui doit à tout instant vérifier la fermeture géométrique (équations de position). Il est dommage de rajouter les inconnues et équations correspondantes alors que les comportements de ces mécanismes sont connus et ne devraient pas avoir à être résolus à chaque instant. C'est pourquoi on se propose d'introduire un nouveau type d'articulation qui rassemble toute la connaissance de la cinématique du contact nécessaire pour gérer la géométrie du système, sans rajouter de traitements inutiles.

3.4.2 Une solution pour la cinématique du contact

Cas de deux disques dans un plan

La position du disque 1 de courbure C_1 par rapport au disque 2 de courbure C_2 peut être paramétrée par les angles θ_1 et θ_2 avec un repère intermédiaire $(\boldsymbol{L}_{21}, \boldsymbol{i}, \boldsymbol{j})$ où $\boldsymbol{i}$ et $\boldsymbol{j}$ sont les vecteurs normal et tangent (figure 3.9). L'évolution de ces angles, en fonction de la vitesse de roulement relatif $\boldsymbol{\Omega}_{21} = \omega.\boldsymbol{k}$ et de la vitesse de glissement tangentiel $\boldsymbol{V}_{21} = V.\boldsymbol{l}_t$, est décrite par les vitesses angulaires:

$$\dot{\theta}_1 = \frac{d\theta_1}{dt} = \frac{C_1.\omega}{C_1 + C_2} - \frac{C_1.C_2.V}{C_1 + C_2} \tag{3.3}$$

et

$$\dot{\theta}_2 = \frac{d\theta_2}{dt} = \frac{-C_2.\omega}{C_1 + C_2} - \frac{C_1.C_2.V}{C_1 + C_2} \tag{3.4}$$

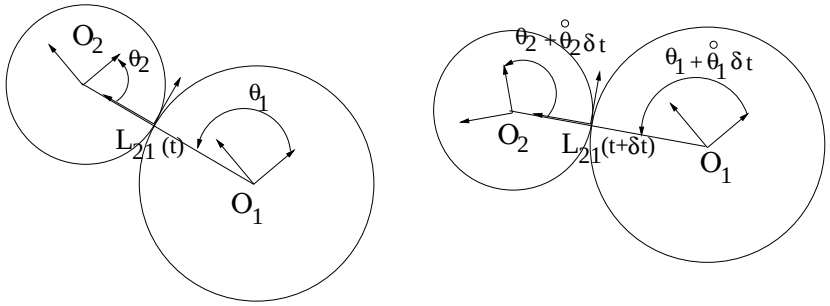


FIG. 3.9 - *Utilisation du repère intermédiaire.*

L'idée est d'utiliser le repère intermédiaire centré au point de contact pour définir la position d'un solide par rapport à l'autre. Ce repère est purement géométrique car il n'est lié à aucun point matériel des solides. Les équations de vitesses instantanées se posent de la manière habituelle, mais l'intégration du mouvement porte sur la position du repère intermédiaire dans chacun des solides en contact, et assure ainsi que les contraintes géométriques associées au contact (non-pénétration, non-décollement) restent satisfaites au cours du temps.

Cas général en deux dimensions

Dans le cas de deux contours quelconques, on peut utiliser l'approximation circulaire locale. Soit $\boldsymbol{P}_{l0}$ la matrice définissant la position absolue du repère intermédiaire de liaison. Le repère intermédiaire de liaison caractérisé par sa position absolue $\boldsymbol{P}_{l0}$ et la matrice de passage $\boldsymbol{P}_{21}$ de R_2 à R_1 sont définis par:

$$\begin{array}{c} \boldsymbol{P}_{l0} = \boldsymbol{P}_{li} \cdot \boldsymbol{P}_{i0} = \boldsymbol{P}_{lj} \cdot \boldsymbol{P}_{j0} \\ \boldsymbol{P}_{21} = \boldsymbol{P}_{l1}.\boldsymbol{P}_{l2}^{-1} \end{array} \tag{3.5}$$

A chaque instant, on détermine les points $\boldsymbol{C_{c1}}$ et $\boldsymbol{C_{c2}}$, qui sont les centres de courbure locale des contours et on les repères en utilisant les vecteurs $\boldsymbol{r}_1 = \boldsymbol{LC}_{c1}$ et $\boldsymbol{r}_2 = \boldsymbol{LC}_{c2}$. Ces vecteurs vont servir à calculer les matrices de rotation nécessaires pour mettre à jour les matrices de passages $\boldsymbol{P}_{l1}$ et $\boldsymbol{P}_{l2}$:

$$\begin{aligned} \boldsymbol{P}_{l1}(t+\delta t) &= \boldsymbol{\delta P}_1 \cdot \boldsymbol{P}_{l1}(t) \\ \boldsymbol{P}_{l2}(t+\delta t) &= \boldsymbol{\delta P}_2 \cdot \boldsymbol{P}_{l2}(t) \end{aligned}$$

avec

$$\boldsymbol{\delta P}_1 = \mathcal{P}(\boldsymbol{r}_1, C_2, V_{21}, \omega_{21})\boldsymbol{\delta P}_1 = \mathcal{P}(\boldsymbol{r}_2, C_1, V_{21}, \omega_{21})$$

où $\mathcal{P}(\boldsymbol{r}_i, C_j, V_{ji}, \omega_{ji})$ désigne la transformation à appliquer à la matrice $\boldsymbol{P}_{li}$ dans le mouvement de j par rapport à i, $\boldsymbol{r}_i$ désigne le centre de courbure local de la surface i, C_j la courbure de la surface de j, V_{ji} et ω_{ji} les vitesses relatives au contact. Cette transformation consiste en une rotation de centre $\boldsymbol{r}_i$ dont l'angle est déterminé en intégrant la vitesse angulaire définie dans la relation (3.3). Si la surface i est localement plane, sa courbure est nulle et la transformation est une translation:

$$\boldsymbol{\delta P}_1 = \boldsymbol{T}(\dot{x_1}\boldsymbol{j}\delta t) \quad avec \quad \dot{x}_1 = V_{21} - \frac{\omega_{21}}{C_2}$$

où $\boldsymbol{T}(\boldsymbol{u})$ représente une matrice de translation suivant un vecteur $\boldsymbol{u}$, tandis que $\boldsymbol{P}_2$ demeure inchangée. Si les deux surfaces sont planes, la rotation relative est nulle et les transformations sont de simples translations.

Approximation en trois dimensions

En trois dimensions, les surfaces en contact ont deux courbures principales et on utilise l'approximation torique. En effet, une surface possédant deux courbures différentes et considérées constantes peut être vue comme le voisinage du cercle extérieur d'un tore. Cette configuration est représentée sur la figure 3.10.

Les tores 1 et 2, en contact au point L_{21} appartenant à leurs cercles extérieurs, ont respectivement pour plans principaux P_1 et P_2 de normales $\boldsymbol{k}_1$ et $\boldsymbol{k}_2$, qui sont inclinés d'un angle β l'un par rapport à l'autre. Leurs repères ont en commun le vecteur $\boldsymbol{i}$. Les vecteurs glissement $\boldsymbol{V}_{21}$ et roulement $\boldsymbol{\omega}_{21}$ appartiennent au plan tangent commun $(L_{21}, \boldsymbol{i})$, et le vecteur pivotement $\boldsymbol{\rho}_{21}$ est normal à celui-ci. Les deux tores sont respectivement

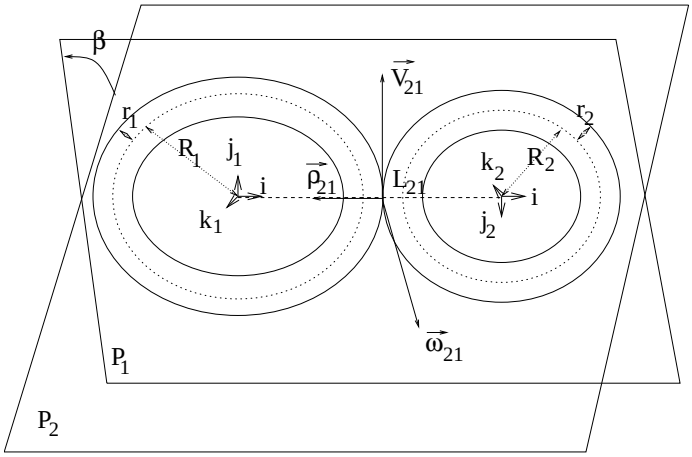


FIG. 3.10 - *Géométrie du contact*

caractérisés par leurs courbures principale et secondaire C_i, c_i dont les axes correspondants sont $\boldsymbol{j}_i$ et $\boldsymbol{k}_i$. Les centres de courbures principale et secondaire des deux surfaces en contact seront repérés par les vecteurs $\boldsymbol{r}_{p1}$, $\boldsymbol{r}_{s1}$, $\boldsymbol{r}_{p2}$ et $\boldsymbol{r}_{s2}$.

En trois dimensions, l'évolution du repère intermédiaire est la combinaison des transformations dans les plans de courbure principale et secondaire. Pour chaque tore, on est ramené à étudier le cas bidimensionnel dans chaque plan de courbure, la transformation tridimensionnelle étant le produit des deux. L'ordre du produit étant arbitraire mais non équivalent, le résultat est une approximation de la solution réelle.

A la différence du cas bidimensionnel, les plans de courbures des deux solides ne sont pas confondus, et on aura à considérer les projections des vecteurs glissement et roulement dans chacun d'eux. Les courbures de chaque surface dans les plans de courbure de l'autre se calculent suivant des formules bien connues [LFA75]. Les formules suivantes donnent les courbures du tore 2 dans les plans de courbure principale et secondaire du tore 1:

$$\begin{aligned} c_2(\boldsymbol{j}_1) &= C_2 \cos^2\beta + c_2 \sin^2\beta \\ c_2(\boldsymbol{k}_1) &= C_2 \sin^2\beta + c_2 \cos^2\beta \end{aligned}$$

On obtient les formules réciproques en permutant simplement les indices. Il faut de plus tenir compte du pivotement relatif, qui s'effectue selon la normale commune aux surfaces en contact, et qui s'intègre comme une liaison pivot classique. On peut arbitrairement choisir d'appliquer cette dernière transformation sur l'une ou l'autre des matrices de passage, ou de la répartir sur les deux. Le changement de repère intermédiaire dans chacun des solides en contact est donc finalement le produit de deux ou trois transformations intermédiaires, qui sont des rotations ou des translations suivant des axes orthogonaux.

Les matrices de passage analogues à celles de l'équation (3.5) sont définies ci-dessous, ainsi que les valeurs des vitesses et des courbure à prendre en compte pour les transformations intermédiaires. Les transformations dans les différents plans de courbure se ramènent ainsi à des cas bidimensionnels. Les matrices $\boldsymbol{\delta P}_n$, $\boldsymbol{\delta P}_p$ et $\boldsymbol{\delta P}_s$ représentent, respectivement, le pivotement, le roulement dans le plan de courbure principale, et le

roulement dans le plan de courbure secondaire.

$$\begin{aligned}
\boldsymbol{\delta P}_{l1} &= \boldsymbol{\delta P}_{s1}.\boldsymbol{\delta P}_{p1} \\
\boldsymbol{\delta P}_{p1} &= \mathcal{P}(\boldsymbol{r}_{p1}, c_2(\boldsymbol{j}_1), \boldsymbol{V}_{21}^L \cdot \boldsymbol{k}_1, \boldsymbol{\Omega}_{21} \cdot \boldsymbol{j}_1 \\
\boldsymbol{\delta P}_{s1} &= \mathcal{P}(\boldsymbol{r}_{s1}, c_2(\boldsymbol{k}_1), \boldsymbol{V}_{21}^L \cdot \boldsymbol{j}_1, \boldsymbol{\Omega}_{21} \cdot \boldsymbol{k}_1
\end{aligned}$$

$$\begin{aligned}
\boldsymbol{\delta P}_{l2} &= \boldsymbol{\delta P}_{s2}.\boldsymbol{\delta P}_{p2}.\boldsymbol{\delta P}_{n} \\
\boldsymbol{\delta P}_{p2} &= \mathcal{P}(\boldsymbol{r}_{p2}, c_1(\boldsymbol{j}_2), \boldsymbol{V}_{21}^L \cdot \boldsymbol{k}_2, \boldsymbol{\Omega}_{21} \cdot \boldsymbol{j}_2 \\
\boldsymbol{\delta P}_{s2} &= \mathcal{P}(\boldsymbol{r}_{s2}, c_1(\boldsymbol{k}_2), \boldsymbol{V}_{21}^L \cdot \boldsymbol{j}_2, \boldsymbol{\Omega}_{21} \cdot \boldsymbol{k}_2 \\
\boldsymbol{\delta P}_{n} &= \boldsymbol{R}((\boldsymbol{\Omega}_{21} \cdot \boldsymbol{i}).\boldsymbol{i})
\end{aligned}$$

où $\boldsymbol{R}(\boldsymbol{u})$ est la rotation d'axe $\boldsymbol{u}$ et d'angle $\|\boldsymbol{u}\|$. L'ordre arbitraire utilisé pour combiner les transformations induit une erreur équivalente à intégrer des angles d'Euler à partir d'une vitesse angulaire en trois dimensions. L'erreur est d'autant plus faible que les mouvements sont petits. De plus, ce modèle de contact ne résoud pas tous les problèmes car les contacts en fermeture de boucle cinématique ne sont pas concernés par les coordonnées articulaires. De plus il n'est pas simple à implémenter et des formulations dynamiques de petits déplacements, abordées plus loin dans ce document, offrent aussi la possibilité de maintenir les contacts au cours du temps.

Modélisation du contact

Un contact est modélisé quand deux objets entrent en collision, comme illustré sur la figure (3.11).

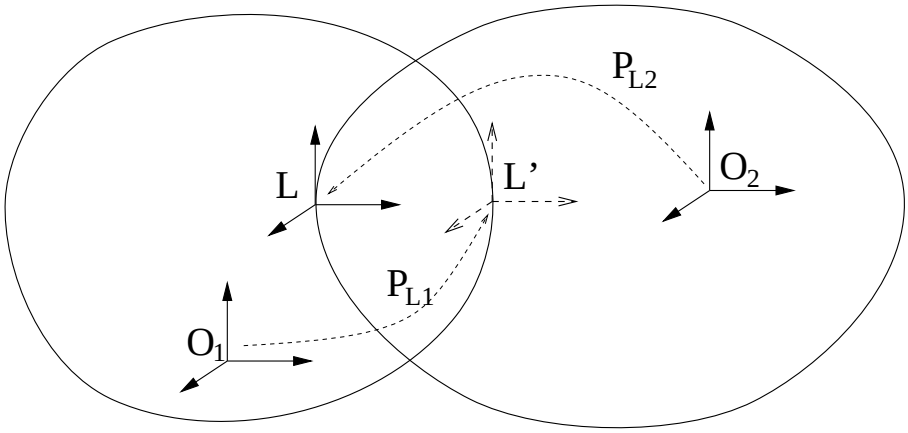


FIG. 3.11 - *Un contact entre solides est modélisé une fois la collision détectée.*

Les points L et L' sont les points où la pénétration est maximale. Ils sont distincts, à moins de revenir dans le temps à l'instant précis de la collision. Dans le cas contraire, on peut utiliser des contraintes cinématiques pour les ramener l'un vers l'autre *a posteriori*.

Application

Les surfaces élémentaires couramment utilisées en modélisation géométrique ont des propriétés de régularité intéressantes. On peut ainsi définir des cas spéciaux pour les contacts entre surfaces de type plan, cylindre, cône, sphère et autres géométries régulières

qui permettent des calculs rapides et exacts. Ces "contacts typés" doivent contenir des données exprimant les positions des axes de courbure par rapport aux surfaces.

Si les surfaces ne sont pas de révolution dans la direction de leur axe de roulement commun, il en découlera une erreur d'intégration proportionnelle à la variation de courbure. L'erreur de position peut être mesurée, et annulée lors d'une passe de correction des positions relatives comme expliqué plus loin dans ce document. A l'instant suivant, on déterminera les nouveaux centres de courbure locale.

Le repère local isolant explicitement les directions normale et tangentes, on peut aisément introduire des contraintes exprimées dans ce repère afin d'autoriser ou interdire à volonté glissement, roulement et pivotement. Des exemples ont été présentés sur une vidéo de démonstration faite pour la présentation de ce travail[Fau95] aux journées de l'AFIG'95.

Application aux surfaces facettisées

Le modèle de contact présenté utilise la notion de courbure, ce qui suppose que les surfaces en contact sont deux fois dérivables. Ce n'est pas le cas de surfaces facettisées. La modélisation d'arêtes pourrait se faire par des cylindres de rayon quasi-nul représentant les arêtes, et des sphères pour les sommets.

Toutefois, pour des surfaces suffisamment facettisées, on aura probablement intérêt à définir un modèle semi-local autour de chaque zone de contact qui approxime les facettes du voisinage par une surface lisse.

3.5 Poser les équations de mouvement

3.5.1 Influences cinématiques

Le centre de liaison est le point où le mouvement relatif est connu à tout instant, on l'utilise à la place du point $\boldsymbol{O_2}$ dans les relations (2.12) et (2.16). On peut ainsi, en parcourant le graphe cinématique, calculer des vitesses ou accélérations, relatives ou absolues. Pour toute articulation A_{jk} et pour tout point $\boldsymbol{p}$, on a:

$$\boldsymbol{v}_{jk}^{L_{jk}} = \sum_l \dot{q}_{jkl}.\boldsymbol{a}_{jkl}$$
$$\boldsymbol{\gamma}_{jk}^{L_{jk}} = \sum_l \ddot{q}_{jkl}.\boldsymbol{a}_{jkl}$$

Le changement de point d'expression s'effectue en utilisant les relations (2.12) et (2.16) et se traduit par:

$$\boldsymbol{v}_{jk}^{P} = \sum_l \dot{q}_{jkl}.\nabla_{PL_{jk}} \cdot \boldsymbol{a}_{jkl}$$
$$\boldsymbol{\gamma}_{jk}^{P} = \sum_l \ddot{q}_{jkl}.\boldsymbol{\nabla}_{PL_{jk}} \cdot \boldsymbol{a}_{jkl} \; + \; \boldsymbol{\delta}_{PL_{jk}}$$

Pour tout solide i, nous noterons $\mathcal{A}(i)$ l'ensemble des articulations rencontrées en remontant le graphe cinématique du solide i jusqu'à la racine. Par exemple, sur la figure

(2.7), $\mathcal{A}(6)$ contient les arcs $56, 15, 01$. Seules les mobilités appartenant aux liaisons ancêtres d'un solide influent sur son mouvement absolu.

L'ensemble des mobilités de toute la structure articulée sera noté $\mathcal{M}$. L'ensemble des mobilités de l'articulation A_{jk} sera noté $\mathcal{M}(jk)$, et l'ensemble des blocages de cette même liaison sera noté $\mathcal{B}(jk)$. Parmi $\mathcal{M}(jk)$, l'ensemble des mobilités dont le mouvement est imposé, et donc connu, sera noté $\mathcal{M}^+(jk)$. L'ensemble des mobilités de A_{jk} dont le mouvement est inconnu sera noté $\mathcal{M}^-(jk)$. On notera $\mathcal{M}(i)$ l'ensemble des mobilités des articulations de $\mathcal{A}(i)$.

La composition des vitesses s'effectue en utilisant la relation (2.14) et se traduit par:

$$\begin{aligned} \boldsymbol{v}_{i0}^p &= \sum_{A_{jk}\in\mathcal{A}(i)} \boldsymbol{v}_{jk}^p \\ &= \sum_{A_{jk}\in\mathcal{A}(i)} \nabla_{PL_{jk}} \sum_{l\in\mathcal{M}(A_jk)} \dot{q}_{jkl}.\boldsymbol{a}_{jkl} \end{aligned}$$

La vitesse absolue du solide i peut donc se représenter par le produit

$$\boldsymbol{v}_{i0}^p = \boldsymbol{J}_{i0}^p \cdot \dot{\boldsymbol{q}} \tag{3.6}$$

où $\dot{\boldsymbol{q}}$ est un vecteur à n dimensions rassemblant l'ensemble des vitesses articulaires de toutes les mobilités, et $\boldsymbol{J}_{i0}^p$ une matrice de taille $(6, n)$ composée de vecteurs-colonnes $\boldsymbol{J}_{i0}^p[*][l]$ tels que

$$\boldsymbol{J}_{i0}^p[*][l] = l \in \mathcal{M}(i) ? \boldsymbol{\nabla}_{PL_l} \cdot \boldsymbol{a}_l : 0$$

Dans cette dernière équation, le point $\boldsymbol{L}_l$ représente le centre de la liaison à laquelle appartient la mobilité l, et $\mathbf{0}$ le vecteur nul en dimension six.

L'accélération absolue du solide i est alors représentable par:

$$\boldsymbol{\gamma}_{i0}^p = \boldsymbol{J}_{i0}^p \cdot \ddot{\boldsymbol{q}} + \hat{\boldsymbol{\gamma}}_{i0}^p \tag{3.7}$$

avec

$$\hat{\boldsymbol{\gamma}}_{i0}^p = \sum_{A_{jk}\in\mathcal{A}(i)} \begin{pmatrix} \boldsymbol{\Omega}_{jk} \wedge (\boldsymbol{\Omega}_{jk} \wedge \boldsymbol{L}_{jk}\boldsymbol{P}) + 2\boldsymbol{\Omega}_{jk} \wedge \boldsymbol{V}_{ij}^p \\ \boldsymbol{\Omega}_{k0} \wedge \boldsymbol{\Omega}_{jk} \end{pmatrix} = \sum_{A_{jk}\in\mathcal{A}(i)} \hat{\boldsymbol{\gamma}}_{ijk}^p \tag{3.8}$$

3.5.2 Influences dynamiques

L'action dynamique en $\boldsymbol{p}$ d'une direction indépendante l vaut $\boldsymbol{f}_l^p = f_l.\boldsymbol{\Delta}_{PL_l} \cdot \boldsymbol{a}_l$. Les forces motrices sont celles exercées par les mobilités, les blocages n'exerçant que des forces résistantes pour empêcher le mouvement relatif. Nous allons considérer uniquement l'action des forces motrices pour l'instant, et reviendrons sur l'action des forces de blocage dans le chapitre consacré spécifiquement à la dynamique. La convention adoptée ici est que l'action dynamique exercée selon une direction indépendante représente l'action du père sur le fils. En vertu du principe de réciprocité, l'action du fils sur le père est l'opposée.

Soit $\mathcal{M}^+(i)$ l'ensemble des mobilités appartenant aux articulations mères de i (celles dont i est le fils). Soit $\mathcal{M}^-(i)$ l'ensemble des mobilités appartenant aux articulations filles de i. Soit $\boldsymbol{f}$ un vecteur à n composantes regroupant l'ensemble des actions dynamiques exercées par les mobilités. La somme des actions dynamiques exercées par les mobilités sur un solide i peut être représentée par:

$$\boldsymbol{f}_i^p = \sum_{l\in\mathcal{M}^+(i)} f_l.\Delta_{PL_l} \cdot \boldsymbol{a}_l - \sum_{l\in\mathcal{M}^-(i)} f_l.\Delta_{PL_l} \cdot \boldsymbol{a}_l$$

Le point $\boldsymbol{L}_l$ est le centre de la liaison à laquelle appartient l. Nous pouvons réécrire cette équation sous forme matricielle:

$$\boldsymbol{f}_i^p = \boldsymbol{F}_i^p \cdot \boldsymbol{f}$$

où $\boldsymbol{F}_i^p$ est une matrice de taille $(6, n)$ composée de vecteurs-colonnes $\boldsymbol{F}_i^p[*][l]$ dont l'expression vaut:

$$\boldsymbol{F}_i^p[*][l] = l \in \mathcal{M}^+(i) ? \boldsymbol{\Delta}_{PL_l} \cdot \boldsymbol{a}_l : l \in \mathcal{A}^-(i) ? - \boldsymbol{\Delta}_{PL_l} \cdot \boldsymbol{a}_l : 0$$

Les colonnes non nulles de cette matrice correspondent aux mobilités des articulations filles ou mères de ce solide.

3.5.3 Équations cinématiques

A chaque direction contrainte l (blocage ou mobilité à mouvement imposé) appartenant à une liaison A_{ij} de fermeture de boucle correspond une équation scalaire du type:

$$\boldsymbol{a}_l^t \cdot \boldsymbol{v}_{ij}^{L_{ij}} = c_i \tag{3.9}$$

où c_i est la valeur du mouvement désiré (nulle pour les blocages). C'est une équation scalaire que l'on peut écrire sous la forme:

$$\boldsymbol{a}_l^t \cdot (\boldsymbol{J}_{i0}^{L_{ij}} - \boldsymbol{J}_{j0}^{L_{ij}}) \cdot \dot{\boldsymbol{q}} = c_i \tag{3.10}$$

Par exemple, la figure (3.12) représente un moteur à explosion et son graphe cinématique. Ce mécanisme bidimensionnel comporte une chaîne fermée et la fermeture de chaîne est constituée par la liaison 30.

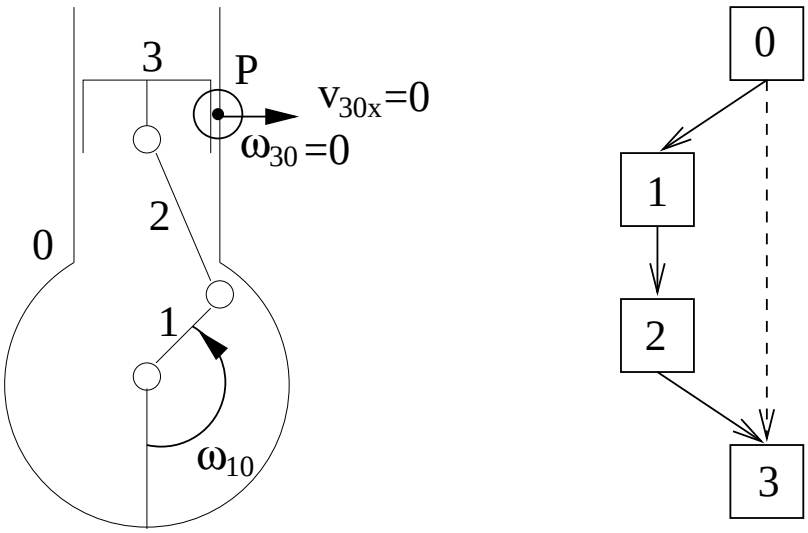


FIG. 3.12 - *Un mécanisme à une chaîne fermée*

Les équations de fermeture de chaîne concernent donc les blocages de cette liaison. Le blocage en translation concerne la direction $\boldsymbol{a}_1 = (1, 0, 0, 0, 0, 0)^t$ et le blocage en rotation

la direction $\boldsymbol{a}_2 = (0,0,0,0,0,1)^t$. Les relations à assurer sont:

$$\begin{aligned}
\boldsymbol{a}_1^t \cdot \boldsymbol{v}_{30}^{L_{30}} &= 0 \\
\boldsymbol{a}_2^t \cdot \boldsymbol{v}_{30}^{L_{30}} &= 0 \\
\boldsymbol{a}_1^t \cdot \boldsymbol{\gamma}_{30}^{L_{30}} &= 0 \\
\boldsymbol{a}_2^t \cdot \boldsymbol{\gamma}_{30}^{L_{30}} &= 0
\end{aligned}$$

Les équations d'accélération sont obtenues en dérivant les équations de vitesse dans le repère intermédiaire de liaison. Soit (ij) le repère intermédiaire de la liaison ij. Selon les types d'articulations, il peut être fixe par rapport à i, j, ou aucun des deux dans le cas d'un contact surfacique. Si nous définissons:

$$\boldsymbol{u}^{\wedge 6} = \begin{pmatrix} \boldsymbol{u}^{\wedge} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{u}^{\wedge} \end{pmatrix}$$

Nous pouvons alors exprimer:

$$\begin{aligned}
\left(\dot{\boldsymbol{v}}_{ij}^{L_{ij}}\right)_{(ij)} &= \left(\dot{\boldsymbol{v}}_{ij}^{L_{ij}}\right)_0 - \boldsymbol{\Omega}_{(ij)0}^{\wedge 6} \cdot \boldsymbol{v}_{ij}^{L_{ij}} \\
&= \boldsymbol{\gamma}_{i0}^{L_{ij}} - \boldsymbol{\gamma}_{j0}^{L_{ij}} - \hat{\boldsymbol{\omega}}_{ij0}
\end{aligned} \tag{3.11}$$

et:

$$\hat{\boldsymbol{\omega}}_{ij0} = \boldsymbol{\Omega}_{(ij)0}^{\wedge 6} \cdot \boldsymbol{v}_{ij}^{L_{ij}} \tag{3.12}$$

Une contrainte d'accélération s'exprime donc sous la forme:

$$\boldsymbol{a}_l^t \cdot (\boldsymbol{J}_{i0}^{L_{ij}} - \boldsymbol{J}_{j0}^{L_{ij}}) \cdot \ddot{\boldsymbol{q}} = c_i + \boldsymbol{a}_l^t \cdot (\hat{\boldsymbol{\gamma}}_{j0}^{L_{ij}} - \hat{\boldsymbol{\gamma}}_{i0}^{L_{ij}} + \hat{\boldsymbol{\omega}}_{ij0}) \tag{3.13}$$

3.5.4 Écriture des équations cinématiques

On commence par numéroter de 0 à $n-1$ toutes les mobilités dont le mouvement est inconnu. On numérote d'autre part de 0 à $m-1$ tous les blocages et mobilités à mouvement imposés appartenant à des articulations de fermeture de boucle. Cette dernière numérotation fournit le nombre d'équations à satisfaire. Dans les algorithmes de ce paragraphe, les directions indépendantes sont désignées par leur numéro. On crée une matrice $\boldsymbol{J}$ de taille (m,n) et un vecteur $\boldsymbol{c}$ de taille m initialisés à zéro. Le pseudocode de la figure (3.13) présente l'algorithme d'écriture des équations de vitesse en traduisant l'équation (3.10).

Les indices f et p représentent le fils et le père de la liaison. On remonte le graphe cinématique jusqu'à la racine pour chacun d'eux. On pourrait encore raffiner l'algorithme car les parties communes aux deux chemins $(\mathcal{A}(f) \cap \mathcal{A}(p))$ s'annulent. Les mobilités dont les mouvements sont imposés $(\mathcal{M}^+(A_{ij}))$ ne font pas partie des inconnues mais leur influence est prise en compte en mettant à jour la valeur de la contrainte.

Le pseudocode de la figure (3.14) présente l'algorithme d'écriture des équations d'accélération conformément à l'équation (3.13). Les termes $\hat{\boldsymbol{\gamma}}$ et $\hat{\boldsymbol{\omega}}$ sont pris en compte pour la mise à jour de la valeur de la contrainte.

```
pour chaque fermeture de chaîne A_fp
    pour chaque l ∈ M+(A_fp) ∪ B(A_fp)                    /* contraintes cinematiques */
        c[l] = c_l                                         /* 0 pour les blocages */
        pour chaque A_ij ∈ A(f)                            /* mouvement du fils */
            pour chaque k ∈ M−(A_ij)                       /* mobilités libres */
                J[l][k] = a_l^t · ∇_{L_ij L_fp} · a_k
            pour chaque k ∈ M+(A_ij)                       /*mobilités contraintes */
                c[l] −= q̇_k.a_l^t · ∇_{L_ij L_fp} · a_k
        pour chaque A_ij ∈ A(p)                            /* mouvement du père */
            pour chaque k ∈ M−(A_ij)
                J[l][k] = −a_l^t · ∇_{L_ij L_fp} · a_k
            pour chaque k ∈ M+(A_ij)
                c[l] += q̇_k.a_l^t · ∇_{L_ij L_fp} · a_k
```

FIG. 3.13 - *Algorithme d'écriture des équations de vitesse*

```
pour chaque fermeture de chaîne A_fp
    pour chaque l ∈ M+(A_fp) ∪ B(A_fp)
        c[l] = c_l + a_l^t · ω̂_fp0
        pour chaque A_ij ∈ A(f)
            c[l] −= a_l^t · γ̂_ijf
            pour chaque k ∈ M−(A_ij)
                J[l][k] = a_l^t · ∇_{L_ij L_fp} · a_k
            pour chaque k ∈ M+(A_ij)
                c[l] −= q̈_k.a_l^t · ∇_{L_ij L_fp} · a_k
        pour chaque A_ij ∈ A(p)
            c[l] += a_l^t · γ̂_ijp
            pour chaque k ∈ M−(A_ij)
                J[l][k] = −a_l^t · ∇_{L_ij L_fp} · a_k
            pour chaque k ∈ M+(A_ij)
                c[l] += q̈_k.a_l^t · ∇_{L_ij L_fp} · a_k
```

FIG. 3.14 - *Algorithme d'écriture des équations d'accélération*

Les pseudocodes fournis dans ce chapitre sont adaptés au traitement de structures cinématiques en coordonnées relatives. On peut tout aussi bien les utiliser pour des structures en coordonnées absolues, en définissant la structure de manière analogue à la figure (2.9). Les ensembles $\mathcal{A}(fils)$ et $\mathcal{A}(pere)$ comportent alors un unique élément qui est une liaison à six degrés de liberté par rapport à l'univers.

3.5.5 Implémentation

Les algorithmes d'écriture des équations suggèrent des champs et méthodes pour les divers objets. Une opération fréquemment utilisée est l'évaluation de produits du type $\boldsymbol{a}_i^t \cdot \boldsymbol{\nabla_u} \cdot \boldsymbol{a}_j$. Par construction, les axes des directions indépendantes et la matrice $\boldsymbol{\nabla_u}$ contiennent beaucoup de zéros, il parait donc judicieux de créer une méthode de calcul efficace *prodNabla* de ce type de produit. Les directions de translation et rotation nécessitent un sous-typage de la classe de base. La figure (3.15) représente les champs et méthodes des directions indépendantes regroupees sous le nom de DirectionContrainte.

DirectionContrainte
- $\boldsymbol{d}$ 3-vecteur direction dans le repère absolu
- q variable d'état (distance ou angle)
- $\dot{q}$ vitesse
- $\ddot{q}$ acceleration
- f force exercée
- *num* numéro d'élément dans l'ensemble de toutes les mobilités de la structure
- $prodNabla(vecteur3\ \boldsymbol{u}, DirectionContrainteb) = \boldsymbol{a}^t \cdot \boldsymbol{\nabla_u} \cdot \boldsymbol{b}$

DirectionContrainteTranslation
- $prodNabla(vecteur\ \boldsymbol{u}, DirectionContrainteTranslation\ b) = \boldsymbol{d} \cdot b.\boldsymbol{d}$
- $prodNabla(vecteur\ \boldsymbol{u}, DirectionContrainteRotation\ b) = \boldsymbol{d} \cdot (\boldsymbol{u} \wedge b.\boldsymbol{d})$

DirectionContrainteRotation
- $prodNabla(vecteur3\ \boldsymbol{u}, DirectionContrainteTranslation\ b) = 0$
- $prodNabla(vecteur3\ \boldsymbol{u}, DirectionContrainteRotation\ b) = \boldsymbol{d} \cdot b.\boldsymbol{d}$

FIG. 3.15 - *Champs et méthodes principaux des mobilités*

La figure (3.16) présente les principaux champs et méthodes de la classe *Articulation*. Les différents groupes de données concernent respectivement la définition géométrique, l'évolution au cours du temps, les mouvements autorisés et les outils nécessaires à l'écriture des équations. Les méthodes ***omegaChap*** et ***gammaChap*** renvoient des vecteurs à six dimensions conformément aux définitions (3.12) et (3.8).

Les champs et méthodes principaux des solides sur la figure (3.17) rassemblent des données sur la position, la vitesse et la place dans le graphe cinématique.

Articulation
pere Solide
fils Solide
pointPere centre de la liaison dans le Solide pere
pointFils centre de la liaison dans le Solide fils
axePere axe de la liaison dans le Solide pere
axeFils axe de la liaison dans le Solide fils

$\boldsymbol{L}$ centre de la liaison dans le repère absolu
$\boldsymbol{P_0}$ position du repere de liaison dans le repere absolu
$\boldsymbol{P_i}$ position du repere de liaison dans le repere pere
$\boldsymbol{P_j}$ position du repere de liaison dans le repere fils

ti, tj, tk, ri, rj, rk six mobilités indépendantes possibles
mobilites liste de mobilites parmi les six possibles, caractéristique du type de liaison
blocages liste complementaire de *mobilites*

omegaChap$(Solide\ i, Solide\ j, Solide\ k) = \boldsymbol{\Omega}^{\wedge 6}_{(ij)0} \cdot \boldsymbol{v}_{ij}^{L_{ij}}$
gammaChap$(Solide\ i, Solide\ j, Solide\ k, vecteur3\ \boldsymbol{p}) = \hat{\boldsymbol{\gamma}}^{p}_{ijk}$

FIG. 3.16 - *Champs et méthodes principaux des articulations*

Solide
$\boldsymbol{P_0}$ /*matrice 44, position absolue*/
$\boldsymbol{O}_i$ /*position de l'origine dans le repère absolu*/

$\boldsymbol{v}$ /* $= \boldsymbol{v}^{\boldsymbol{o_i}}_{\boldsymbol{i0}}$ vitesse généralisée à l'origine*/
vPij$(point\ \boldsymbol{p}, Solide\ j) = \boldsymbol{\nabla}_{PO_i} \cdot \boldsymbol{v} - \boldsymbol{\nabla}_{PO_j} \cdot j.\boldsymbol{v}$

Meres /*liste d'articulations*/
Filles /*liste d'articulations*/

FIG. 3.17 - *Champs et méthodes principaux des solides*

3.6 Applications

3.6.1 Structures articulées

La généricité du modèle d'articulation présenté a permis d'en créer un grand nombre de variétés facilement paramétrables. Une interface en langage tcl a été réalisée par Marc Vidal et a servi dans un grand nombre d'applications. La définition du bonhomme de la figure 3.18 met en jeu plusieurs types d'articulations.

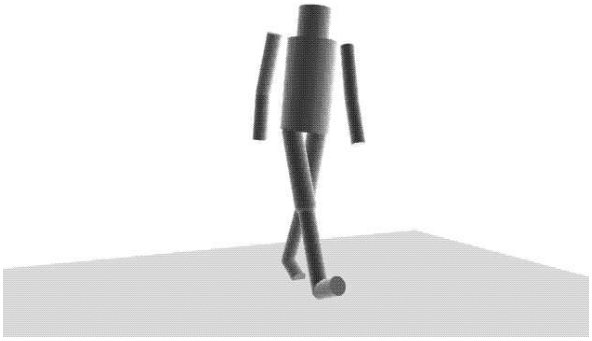


FIG. 3.18 - *Un personnage articulé faisant intervenir quatre sortes de liaisons*

Ce personnage, bien que tridimensionnel, est conçu pour l'analyse de mouvements de marche bidimensionnels. On a donc eu recours à des pivots (FaArtPivotDriven) pour les articulations. Le corps est soumis à un mouvement plan, et le basculement vers l'avant ou l'arrière a été empêché en définissant une liaison de translation dans le plan de la marche (FaArtPlaneTranslation). En l'absence de données sur le mouvement du cou, celui-ci a été rigidifié (FaArtStill). D'un point de vue purement cinématique, cette dernière articulation n'a aucune utilité, mais en dynamique inverse elle permet de calculer des efforts intérieurs à un assemblage rigide. Un contact extérieur (FaArtSurfaceContact) a été initialisé entre le talon droit et le sol. Une partie du fichier de définition du personnage est fournie en annexe E.

3.6.2 Contrôle du mouvement

Les articulations peuvent aussi être utilisées pour appliquer certaines contraintes de mouvement. Prenons comme exemple le point $\boldsymbol{p}_1$ du solide S_1 qui doit rejoindre le point $\boldsymbol{p}_2$ du solide S_2 (fig. 3.19) en un temps Δt. La distance au départ est de d_0. Nous pouvons utiliser un solide intermédiaire articulé au moyen d'une rotule sur S_1 et d'une linéaire annulaire sur S_2 (liaisons définies sur la figure 3.2). Il ne reste qu'à appliquer la contrainte $ti.\dot{q} = d_0/\Delta t$ pour voir les deux points se rejoindre au moment voulu.

3.7 Conclusion

Le modèle d'articulation présenté permet de représenter toutes les liaisons conventionnelles entre deux solides. L'approche orientée-objet et l'utilisation d'une liste de directions

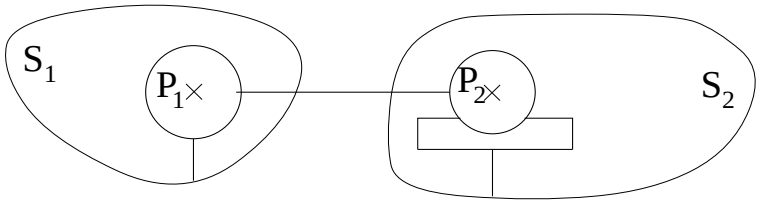


FIG. 3.19 - *Une liaison rotule sur S_1 et une liaison linéaire annulaire sur S_2 sont utilisées pour amener progressivement les points P_1 de S_1 et P_2 de S_2 à coïncider.*

indépendantes de mouvements permet de définir rapidement de nouveaux types d'articulations par sous-typage de la classe générique et de traiter les contraintes correspondantes.

Ces articulations peuvent être centrées et orientées par rapport aux solides à la convenance de l'utilisateur. Le repère intermédiaire de liaison permet la modélisation cinématique du contact surfacique. Le traitement spécifique des contacts extérieurs, relativement compliqué et imprécis, n'est pas nécessaire en coordonnées absolues. Le repère intermédiaire de liaison reste toutefois utile pour les liaisons qui ne sont pas centrées à l'origine des repères.

Ces liaisons génériques permettent en outre une mise en équation de toute contrainte de vitesse ou d'accélération au moyen des algorithmes simples présentés dans ce document. Elles ont été utilisées dans tous les algorithmes d'animation décrits dans ce document.

Chapitre 4

Approche cinématique de l'animation

Nous nous intéresserons dans ce chapitre à la résolution de contraintes cinématiques sur une structure articulée, par exemple un personnage. En effet, il est plus simple pour un utilisateur de spécifier le mouvement désiré sous forme d'un ensemble de contraintes (par exemple, amener la main en certain point) plutôt que sous forme d'un ensemble de lois de variations des articulations au cours du temps. Les contraintes peuvent s'écrire sous forme d'équations en utilisant les algorithmes vus au chapitre précédent. Il n'est toutefois pas toujours aisé de résoudre ces équations, et nous passerons en revue les principales approches avant de présenter la nôtre. Elle consiste à caractériser l'ensemble des mouvements autorisés par les contraintes. Nous présentons à la suite une formulation dynamique originale qui consiste à exprimer les équations de la dynamique sur l'espace des mouvements autorisés. Cette approche permet d'utiliser l'animation par modèle physique tout en respectant les contraintes cinématiques du modèle et du scénario.

4.1 Contrôle du mouvement

La résolution des équations linéaires cinématiques permet de calculer des vitesses ou accélérations compatibles avec les contraintes. Les équations cinématiques se présentent sous la forme:

$$\boldsymbol{J} \cdot \boldsymbol{x} = \boldsymbol{c} \tag{4.1}$$

où $\boldsymbol{J}$ est la matrice jacobienne des contraintes, $\boldsymbol{x}$ un vecteur qui représente l'ensemble des inconnues cinématiques, et $\boldsymbol{c}$ les valeurs des contraintes à satisfaire. Ce système d'équation peut être obtenu en utilisant les structures et algorithmes présentés au chapitre 3. La matrice jacobienne est généralement non carrée, ce qui implique que sa matrice inverse n'est pas définie. Beaucoup de méthodes classiques de résolution de systèmes (LU, Cholesky) fonctionnent avec des matrices carrées. Les sytèmes d'équations cinématiques peuvent selon les cas avoir aucune, une, ou une infinité de solutions comme sur la figure (4.1).

Les systèmes comportant un nombre infini de solutions (qualifiés de *redondants* en robotique) peuvent être traités par différentes méthodes, comme par exemple le classique

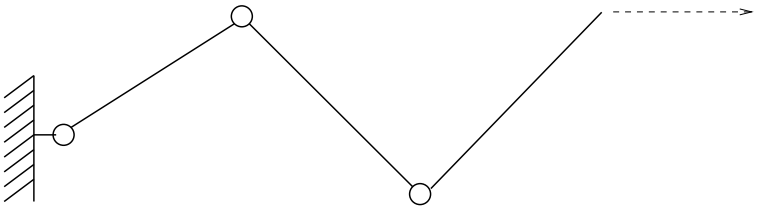


FIG. 4.1 - *Une infinité de solutions sont admissibles pour satisfaire la contrainte symbolisée par la flèche pointillée.*

pivot de Gauss. Mais parmi ces solutions, on peut vouloir trouver la meilleure suivant un certain critère. Nous verrons que cette question est également traitable par les méthodes adaptées aux systèmes à une ou aucune solution, que nous allons considèrer à présent.

4.1.1 Approches par optimisation

La figure (4.2) montre un exemple de contraintes à résoudre pour respecter un scénario: le personnage doit toucher l'objet suspendu à droite de l'image sans bouger les pieds du sol. Malheureusement, il n'a pas le bras assez long pour cela. Le système n'a pas de solution.

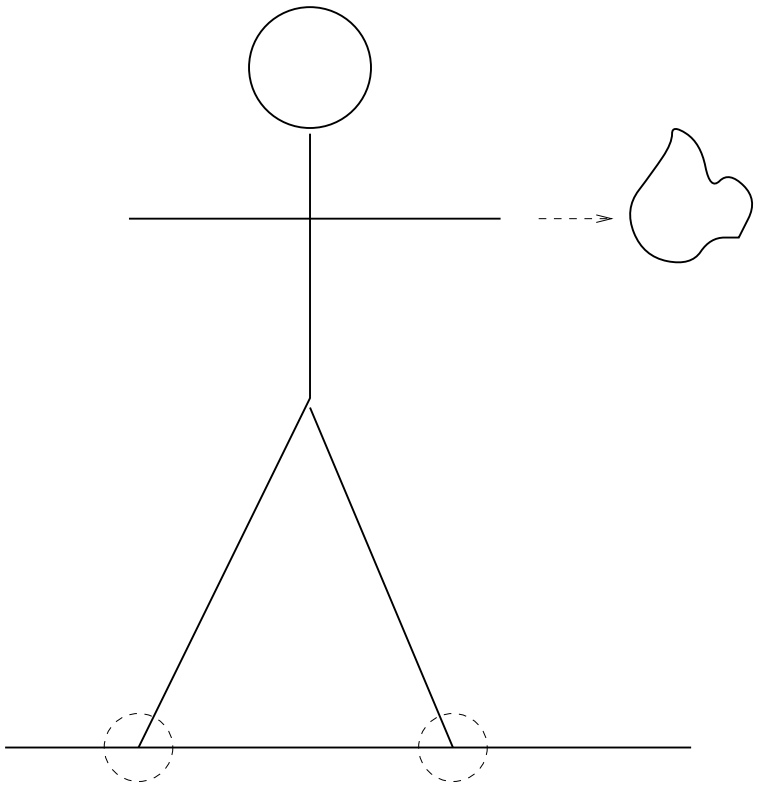


FIG. 4.2 - *Un personnage les pieds fixés au sol a pour contrainte de toucher l'objet à droite de l'image. Les pointillés représentent les contraintes cinématiques.*

Dans ce cas, on attend d'une méthode de résolution qu'elle fournisse une solution satisfaisant au mieux les contraintes.

C'est pourquoi des méthodes d'optimisation ont été proposées[WFB87, ZB94a]. Effectuer une optimisation consiste à chercher itérativement une solution minimisant un certain critère d'erreur, autrement dit trouver $\boldsymbol{x}$ tel que $E(\boldsymbol{x})$ soit minimal.

Dans le cas des contraintes cinématiques, ce critère peut être une somme pondérée des erreurs sur les différentes contraintes. Si $\boldsymbol{x}$ est une approximation de la solution et $\boldsymbol{b} = \boldsymbol{J} \cdot \boldsymbol{x}$, on peut utiliser $E(\boldsymbol{x}) = \sum_i \alpha_i.(b_i - c_i)^2$.

Si l'on résoud ce problème par optimisation, la solution sera un compromis entre les différentes contraintes et le résultat va tendre vers une configuration similaire à celle représentée sur la figure (4.3). Les pieds décollent ou pénètrent dans le sol, ce qui n'est pas satisfaisant du point de vue du réalisme.

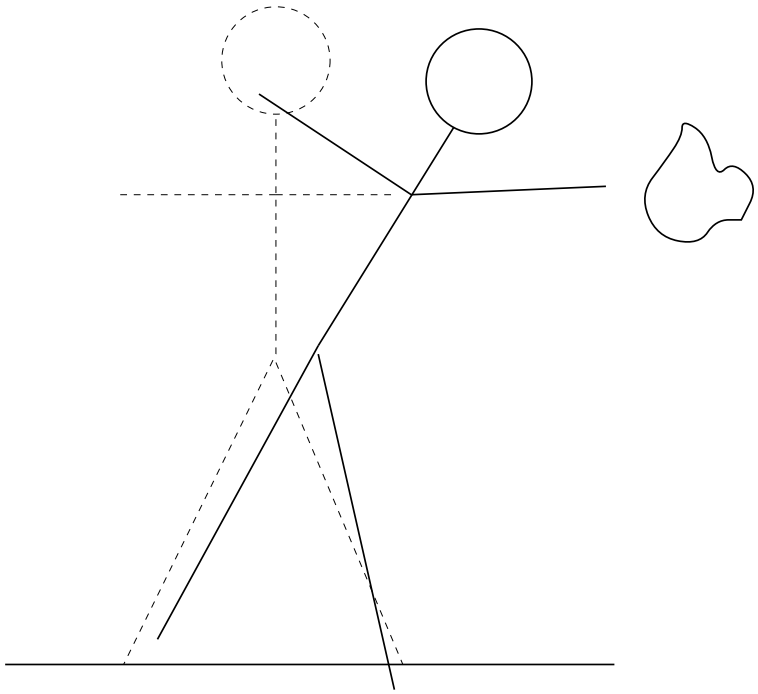


FIG. 4.3 - *Les méthodes d'optimisation cherchent le meilleur compromis entre les contraintes.*

On préfèrerait probablement que les pieds restent au sol, quitte à ce que la main manque plus largement sa cible. En théorie, on peut tendre vers ce résultat en affectant dans la fonction d'erreur à minimiser un poids considérablement plus important aux erreurs pied-sol qu'à l'erreur main-cible. En pratique, la mise au point de ces poids est très délicate. Les algorithmes d'optimisation fonctionnent d'autant mieux que la fonction d'erreur a une courbure régulière. Il est plus rapide en pratique de trouver le fond d'un bol que le fond d'une coque de kayak. Des poids trops forts aux erreurs pied-sol pénalisent le taux de convergence de la méthode, rendant celle-ci lente et difficile à contrôler.

A l'actif des approches par optimisation, on peut noter qu'elles sont intrinsèquement bien adaptées à la recherche de la meilleure parmi un ensemble de solution satisfaisant les contraintes. Dans ce cas, la meilleure solution peut être définie, par exemple, comme celle qui minimise les mouvements, ou tout autre critère jugé pertinent par l'utilisateur. Leur inconvénient est, comme on l'a vu, d'être mal adaptées au traitement de différents niveaux de priorité des contraintes.

4.1.2 Pseudo-inverses droite et gauche

Une autre technique de résolution d'équations cinématiques couramment rencontrée dans la littérature est le calcul de la *pseudo-inverse*, à droite ou à gauche, de la matrice jacobienne[Pau81, GM85, BTT90, BMT95]. La *pseudo-inverse à droite* de la matrice jacobienne $\boldsymbol{J}$, couramment utilisée en robotique, est utile pour les systèmes redondants, comportant une infinité de solutions. Elle est notée $\boldsymbol{J}^+$ et vaut:

$$\boldsymbol{J}^+ = \boldsymbol{J}^t \cdot \left(\boldsymbol{J} \cdot \boldsymbol{J}^t\right)^{-1}$$

Elle vérifie la propriété $\boldsymbol{J} \cdot \boldsymbol{J}^+ = \boldsymbol{I}$ où $\boldsymbol{I}$ est la matrice identité. On effectue la pseudo-inversion de l'équation cinématique

$$\boldsymbol{J} \cdot \boldsymbol{x} = \boldsymbol{c} \tag{4.2}$$

au moyen de l'expression:

$$\begin{array}{rcl} \boldsymbol{x} & = & \boldsymbol{J}^+ \cdot \boldsymbol{c} + (\boldsymbol{I} - \boldsymbol{J}^+ \cdot \boldsymbol{J}) \cdot \boldsymbol{q} \\ & = & \boldsymbol{q_1} + \boldsymbol{q_2} \end{array} \tag{4.3}$$

où $\boldsymbol{q}$ est un vecteur arbitraire. Le premier terme $\boldsymbol{q_1}$ est la solution de plus petite norme qui satisfait l'équation cinématique. Le deuxième terme décrit le noyau de la matrice jacobienne (i.e. $\boldsymbol{J} \cdot \boldsymbol{q_2} = \boldsymbol{0}$), et ainsi ne perturbe pas la réalisation de la consigne. C'est la projection de $\boldsymbol{q}$ sur le noyau de la matrice. Cet opérateur de projection est très utile car en prenant $\boldsymbol{q}$ dirigé selon le gradient d'une fonction potentielle à optimiser, ce terme permet de mettre à profit la redondance de la chaîne cinématique pour prendre en compte autant que faire se peut une contrainte secondaire, tout en respectant exactement la contrainte cinématique (4.2). Dans l'exemple de la figure (4.2), on spécifierait comme contraintes primaires les liaisons pied-sol, et comme fonction à optimiser le déplacement de la main vers la cible. On obtiendrait alors un résultat similaire à celui présenté sur la figure (4.4).

Dans les cas où aucune solution n'existe et que l'on veut trouver la meilleure solution à des contraintes antagonistes, on peut utiliser la *pseudo-inverse à gauche* définie par:

$$\boldsymbol{J}^- = \left(\boldsymbol{J}^t \cdot \boldsymbol{J}\right)^{-1} \cdot \boldsymbol{J}^t$$

et qui vérifie $\boldsymbol{J}^- \cdot \boldsymbol{J} = \boldsymbol{I}$. Le vecteur

$$\boldsymbol{q} = \boldsymbol{J}^- \cdot \boldsymbol{c}$$

minimise alors l'erreur sur la consigne.

L'approche par pseudo-inverse est séduisante mais souffre tout de même de certains inconvénients. Pour choisir d'utiliser la pseudo-inverse à droite ou à gauche, il est nécessaire de savoir si l'on se trouve dans un cas redondant ou non. La réponse peut être simple si l'on traite d'un mécanisme bien connu comme par exemple un bras de robot pour lequel des test *ad hoc* peuvent être utilisés. Pour la cinématique d'une structure quelconque, il n'y a pas de réponse facile. La stratégie consistant à comparer le nombre d'équations et d'inconnues est trop simpliste et échouera dans bien des cas. Dans l'exemple de la figure (4.1), si les liaisons sont des pivots, une contrainte normale au plan de la figure génèrera un système avec moins d'équations que d'inconnues et pourtant insoluble.

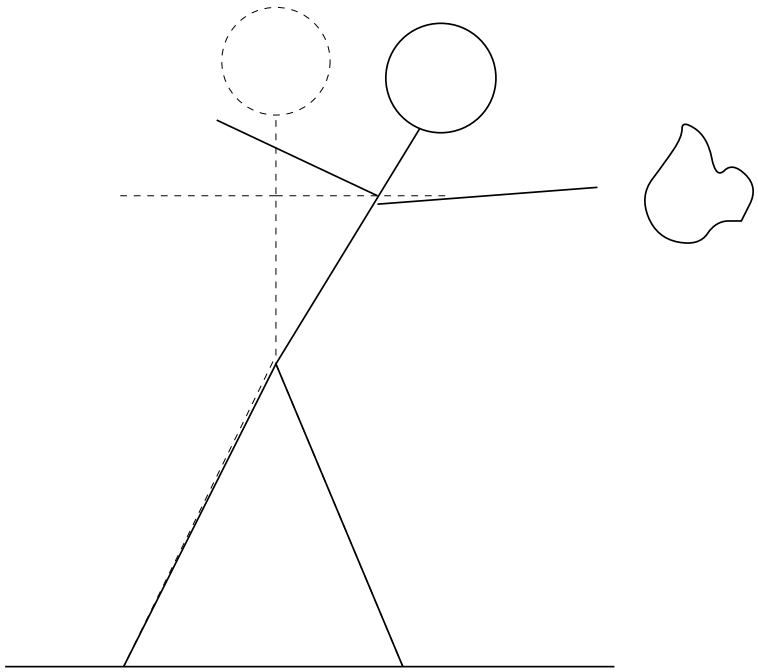


FIG. 4.4 - *La pseudo-inverse à droite permet de prendre en compte des priorités entre les contraintes.*

Une autre limitation est le nombre de niveaux de priorité. Une fois le premier niveau pris en compte (pied-sol dans l'exemple de la figure 4.4), toutes les autres contraintes sont traitées par optimisation, et l'on retrouve les problèmes évoqués au paragraphe 4.1.1. La figure (4.5) un cas où le personnage est capable d'atteindre une cible mais pas les deux.

Le créateur du scénario, s'il a envisagé le cas où le personnage ne peut atteindre aucune cible, aura assigné aux contraintes correspondantes une priorité inférieure à celles des contraintes pied-sol, pour éviter de se retrouver dans la situation de la figure (4.3). Mais s'il veut qu'au moins une cible puisse être atteinte, autrement dit ne pas se retrouver comme le bonhomme pointillé de la figure (4.5), il doit à nouveau définir une priorité entre les cibles. Or cette échelle de priorités, à trois niveaux, n'est pas traitable par l'approche pseudo-inverse.

4.2 Résolution de niveaux successifs de contraintes

Le travail présenté ici est une nouvelle approche des contraintes cinématiques permettant de traiter un nombre arbitraire de niveaux de priorité pour les contraintes cinématiques. L'idée de base est que chaque niveau de contraintes pris en compte entraîne une réduction du nombre de variables indépendantes. Ces variables indépendantes sont des combinaisons linéaires des variables indépendantes initialement disponibles, et qui respectent les nouvelles contraintes. Pour prendre en compte de nouvelles contraintes, on cherchera la ou les solutions parmi les variables disponibles.

La clé de voute de la méthode est la capacité de dégager des variables indépendantes satisfaisant un ensemble de contraintes. Nous allons voir que la décomposition par valeurs singulières de la matrice jacobienne permet d'isoler les variables indépendantes. En

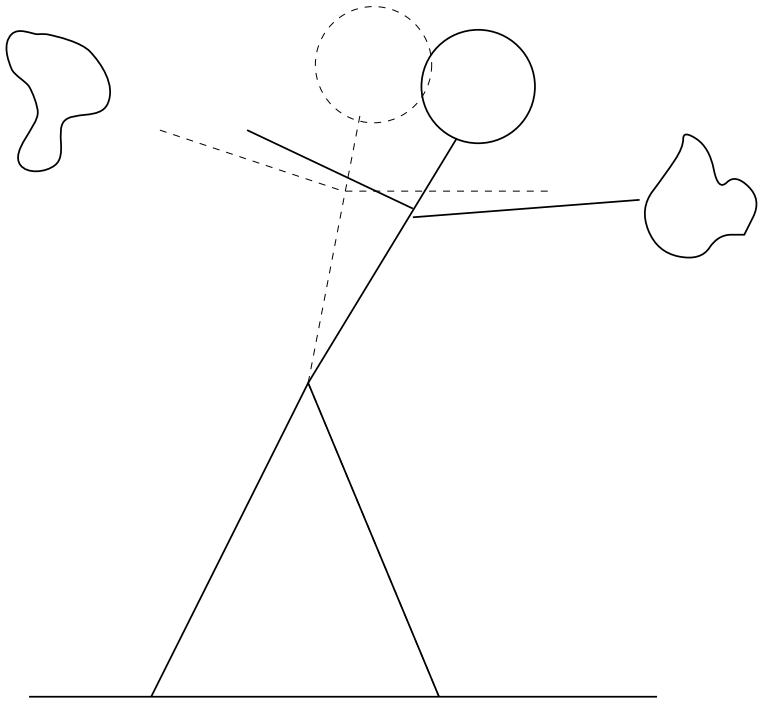


FIG. 4.5 - *En pointillé: résultat avec deux niveaux de priorité. En continu: résultat avec trois niveaux.*

l'absence de toute contrainte cinématique, les variables cinématiques indépendantes correspondent au paramétrage utilisé: vitesse(accélération) absolue des solides si l'on travaille en coordonnées absolues, vitesses(accélérations) articulaires si l'on travaille en coordonnées relatives. Nous verrons comment résoudre un niveau de contrainte et isoler les variables indépendantes restantes. Nous verrons ensuite comment enchaîner sur le niveau de contraintes suivant. Un algorithme permettant de traiter un nombre quelconque de niveaux de contraintes sera finalement proposé.

4.2.1 Décomposition par valeurs singulières de la matrice jacobienne

Toute matrice $\boldsymbol{J}$ possédant un nombre de lignes m supérieur ou égal à son nombre de colonnes n peut se décomposer en produit de trois matrices en utilisant l'algorithme de décomposition par valeur singulière[PTVF92].

$$\boldsymbol{J}_{m,n} = \boldsymbol{U}_{m,n} \cdot \boldsymbol{W}_{n,n} \cdot V^t_{n,n} \tag{4.4}$$

Les indices indiquent ici les tailles des matrices. Dans le cas où m est inférieur à n, il suffit de rajouter à la matrice le nombre approprié de lignes nulles. Les matrices $\boldsymbol{U}$ et $\boldsymbol{V}$ vérifient les relations:

$$\boldsymbol{U}^t \cdot \boldsymbol{U} = \boldsymbol{V}^t \cdot \boldsymbol{V} = \boldsymbol{I}$$

où $\boldsymbol{I}$ est la matrice identité de taille $n \times n$. Ceci signifie qu'elles sont toutes deux composés de vecteurs-colonnes de norme 1 et orthogonaux deux à deux. La matrice V

étant carrée, ses vecteurs-lignes vérifient aussi ces propriétés. La matrice $\boldsymbol{W}$, quant à elle, est diagonale et les éléments de sa diagonale (appelés valeurs singulières) sont positifs ou nuls.

Ces trois matrices contiennent des informations précieuses. Les colonnes de $\boldsymbol{U}$ correspondant aux éléments non nuls de $\boldsymbol{W}$ constituent une base orthonormale de l'image de $\boldsymbol{J}$, c'est-à-dire l'ensemble des vecteurs $\boldsymbol{y} = \boldsymbol{J} \cdot \boldsymbol{x}$ quand $\boldsymbol{x}$ décrit $\mathbb{R}^m$. Les colonnes de $\boldsymbol{V}$ correspondant aux éléments nuls de $\boldsymbol{W}$ forment une base orthonormale du noyau de $\boldsymbol{J}$, c'est-à-dire de l'ensemble des vecteurs $\boldsymbol{x}$ tels que $\boldsymbol{J} \cdot \boldsymbol{x} = \boldsymbol{0}$. Notons aussi que le rapport entre la plus grande et la plus petite valeur singulière est égal à la valeur de conditionnement de la matrice. Annuler les plus petites valeurs $\boldsymbol{W}$ permet de réduire les erreurs numériques lors de la résolution.

La résolution de l'équation cinématique (4.2) s'effectue au moyen du produit:

$$\boldsymbol{x} = \boldsymbol{V} \cdot \boldsymbol{W}^* \cdot \boldsymbol{U}^t \cdot \boldsymbol{c}$$

où la matrice $\boldsymbol{W}^*$ est une matrice diagonale dont chaque élément est soit l'inverse de l'élement correspondant de $\boldsymbol{W}$ si celui-ci est non nul, soit zéro s'il est nul. Le vecteur $\boldsymbol{x}$ est la plus petite solution à l'équation cinématique. Si aucune solution n'existe, il est la meilleure solution au sens des moindres carrés. Barzel et Barr[BB88] utilisent cette approche pour trouver une solution à leur système d'équations. La matrice $\boldsymbol{V} \cdot \boldsymbol{W}^* \cdot \boldsymbol{U}^t \cdot \boldsymbol{c}$ est parfois appelée la *pseudo-inverse généralisée* de $\boldsymbol{J}$. Nous appellerons par ailleurs *matrice noyau* de $\boldsymbol{J}$ la matrice composée des vecteurs-colonnes de $\boldsymbol{V}$ qui forment la base orthonormale du noyau de $\boldsymbol{J}$.

4.2.2 Résoudre un niveau de contrainte

Soit n le nombre de ces variables. Soit $\boldsymbol{J}_1$ la matrice jacobienne correspondant aux contraintes du premier niveau (les plus fortes), et $\boldsymbol{c}_1$ la contrainte. La décomposition par valeur singulière de $\boldsymbol{J}_1$ permet d'extraire sa pseudo-inverse généralisée $\boldsymbol{S}_1$ et sa matrice noyau $\boldsymbol{L}_1$. Nous pouvons maintenant exprimer la solution sous la forme:

$$\boldsymbol{x} = \boldsymbol{S}_1 \cdot \boldsymbol{c}_1 + \boldsymbol{L}_1 \cdot \boldsymbol{l}_1 \tag{4.5}$$

pour tout $\boldsymbol{l}_1 \in \mathbb{R}^k$. Tout vecteur $\boldsymbol{l}_1$ satisfait l'équation cinématique puisque par définition, $\boldsymbol{J}_1 \cdot \boldsymbol{L}_1 = \boldsymbol{0}$. Il reste maintenant k variables indépendantes ($k < n$) pour satisfaire le deuxième niveau de contraintes. S'il n'y a pas de contraintes supplémentaires, nous pouvons utiliser ces variables pour optimiser un éventuel critère de qualité.

Par rapport à l'utilisation de la pseudo-inverse à droite (eq. 4.3), nous avons gagné sur deux tableaux. Premièrement, nous pouvons encore traiter plusieurs niveaux de contraintes. Deuxièmement, si nous voulons plutôt optimiser un critère de qualité, nous devons ajuster k variables indépendantes au lieu de n variables dépendantes.

4.2.3 Résoudre le niveau suivant

Nous avons maintenant à résoudre l'équation $\boldsymbol{J}_2 \cdot \boldsymbol{x} = \boldsymbol{c}_2$ tout en respectant la première équation cinématique (eq. 4.5). Cela donne:

$$\boldsymbol{J}_2 \cdot (\boldsymbol{S}_1 \cdot \boldsymbol{c}_1 + \boldsymbol{L}_1 \cdot \boldsymbol{l}_1) = \boldsymbol{c}_2$$

$$\boldsymbol{J}_2 \cdot \boldsymbol{L}_1 \cdot \boldsymbol{l}_1 = \boldsymbol{c}_2 - \boldsymbol{J}_2 \cdot \boldsymbol{S}_1 \cdot \boldsymbol{c}_1$$

$$\boldsymbol{l}_1 = \boldsymbol{S}_2 \cdot (\boldsymbol{c}_2 - \boldsymbol{J}_2 \cdot \boldsymbol{S}_1 \cdot \boldsymbol{c}_1) + \boldsymbol{L}_2 \cdot \boldsymbol{l}_2$$

$$\boldsymbol{x} = \boldsymbol{S}_1 \cdot \boldsymbol{c}_1 + \boldsymbol{L}_1 \cdot \boldsymbol{S}_2 \cdot (\boldsymbol{c}_2 - \boldsymbol{J}_2 \cdot \boldsymbol{S}_1 \cdot \boldsymbol{c}_1) + \boldsymbol{L}_1 \cdot \boldsymbol{L}_2 \cdot \boldsymbol{l}_2$$

où $\boldsymbol{S}_2$ et $\boldsymbol{L}_2$ représentent respectivement la pseudo-inverse généralisée et la matrice noyau de la matrice $\boldsymbol{J}_2 \cdot \boldsymbol{L}_1$. Notons que cette matrice comporte k colonnes, puisque les inconnues sont les variables cinématiques indépendantes de l'équation (4.5). Les nouvelles inconnues cinématiques sont les composantes du vecteur $\boldsymbol{l}_2$ de taille k_2, avec $k_2 < k$. Le nombre d'inconnues diminue donc à chaque nouveau niveau de contraintes traité.

4.2.4 Algorithme complet

On présente ici un algorithme permettant de traiter un nombre N de niveaux de contraintes. Cet algorithme, présenté sur la figure (4.6), a été implémenté avec succès.

On suppose l'existence de matrices et vecteurs $(\boldsymbol{J}_i, \boldsymbol{c}_i)$, avec $1 \leq i \leq N$, traduisant les contraintes aux différents niveaux. La fonction $dcmp(\boldsymbol{J}, \boldsymbol{S}, \boldsymbol{L})$ range dans $\boldsymbol{S}$ et $\boldsymbol{L}$ la pseudo-inverse généralisée et la matrice noyau de $\boldsymbol{J}$. La fonction $nbCol(\boldsymbol{L})$ renvoie le nombre de colonnes de la matrice $\boldsymbol{L}$. La constante $\boldsymbol{I}$ représente la matrice identité, de taille du nombre total d'inconnues. Le vecteur $\boldsymbol{x}$ représente la solution particulière à l'ensemble des contraintes, et la matrice $\boldsymbol{L}_r$ représente les variables indépendantes restantes.

$\boldsymbol{x}_r = \boldsymbol{0}$
$\boldsymbol{L}_r = \boldsymbol{I}$
$i = 1$
tant que ($i < n$ et $nbCol(\boldsymbol{L}_r) > 0$):
  $\boldsymbol{J}_i \mathrel{*}= \boldsymbol{L}$
  $\boldsymbol{c}_i \mathrel{-}= \boldsymbol{J}_i \cdot \boldsymbol{x}_r$
  $dcmp(\boldsymbol{J}_i, \boldsymbol{S}, \boldsymbol{L})$
  $\boldsymbol{x}_r = \boldsymbol{L}_r \cdot \boldsymbol{S} \cdot \boldsymbol{c}_i$
  $\boldsymbol{L}_r \mathrel{*}= \boldsymbol{L}$

FIG. 4.6 - *Résoudre plusieurs niveaux de contrainte*

Après application de l'algorithme , la solution de l'ensemble des contraintes s'exprime sous la forme $\boldsymbol{x} = \boldsymbol{x}_r + \boldsymbol{L}_r \boldsymbol{l}$ où $\boldsymbol{l}$ est un vecteur représentant les variables indépendantes restantes. L'algorithme peut terminer avant d'avoir traité tous les niveaux s'il ne reste

plus de variables cinématiques disponibles. Le système est alors surcontraint et seules les contraintes de plus haut niveau sont satisfaites.

4.3 De la Cinématique à la Dynamique

Nous disposons de variables cinématiques satisfaisant les contraintes cinématiques de notre structure solide. Nous les appelons *variables libres* et les notons l_i. Ces variables sont des combinaisons linéaires des variables d'origine (coordonnées articulaires ou absolues) obtenues par la méthode décrite au chapitre 4.2. Nous les rassemblons dans un vecteur noté $\boldsymbol{l}$, et on peut en déduire les valeurs des variables d'origine, rassemblées dans un vecteur $\boldsymbol{q}$, par le produit matriciel $\boldsymbol{q} = \boldsymbol{L} \cdot \boldsymbol{l}$. La matrice $\boldsymbol{L}$ regroupe des vecteurs-colonnes qui codent les combinaisons linéaires de variables d'origine correspondant à chaque variable libre l_i. Ces variables sont toutes des vitesses, ou toutes des accélérations. Dans ce chapitre consacré à la dynamique, nous considérons qu'elles représentent des accélérations.

4.3.1 Principe de d' Alembert

Le principe de d'Alembert, aussi appelé principe des puissances virtuelles ou principe des travaux virtuels, est une application du théorème de l'énergie (eq. 2.31). C'est un outil bien connu et utilisé en synthèse d'animation. Les formulations dynamiques CRBM[WB85, IC87] reposent dessus; Isaacs et Cohen l'utilisent pour résoudre des contraintes cinématiques complexes[IC88]. L'originalité ici est de l'appliquer aux variables cinématiques indépendantes satisfaisant l'ensemble des contraintes cinématiques calculées préalablement. Pour un solide S, nous pouvons écrire:

$$\boldsymbol{v}_{S0}^{p\ t} \cdot \boldsymbol{f}_{\bar{S}S}^{p} = \boldsymbol{v}_{S0^p}^{t} \cdot \ddot{\boldsymbol{d}}_{S0}^{p}$$

Dans une structure solide, les actions dynamiques exercées sur un solide i peuvent se décomposer en celles exercées par les autres solides j de la structure et celles exercées par l'extérieur de la structure. On trouve dans la première catégorie les efforts aux liaisons, et dans la deuxième le poids. Nous noterons $\boldsymbol{f}_{ji}$ l'action dynamique de j sur i et $\boldsymbol{f}_i$ l' action dynamique sur i en provenance de l'extérieur de la structure. Pour appliquer le théorème de l'énergie à une structure de solides articulés, nous devons donc écrire:

$$\sum_i \sum_{j \neq i} \boldsymbol{v}_{i0}^{p_i t} \cdot \boldsymbol{f}_{ji}^{p_i} + \sum_i \boldsymbol{v}_{i0}^{p_i t} \cdot \boldsymbol{f}_i^{p_i} = \sum_i \boldsymbol{v}_{i0}^{p_i t} \cdot \ddot{\boldsymbol{d}}_{i0}^{p_i t} \tag{4.6}$$

Cette relation est valable quel que soient les points d'expression $\boldsymbol{P}_i$ choisis, et donc en particulier pour les centres de masses $\boldsymbol{g}_i$.

Conformément aux notations du chapitre 3, nous avons:

$$\begin{aligned} \boldsymbol{v}_{i0}^{g_i} &= \boldsymbol{J}_{i0}^{g_i} \cdot \ddot{\boldsymbol{q}} \\ \sum_{j \neq i} \boldsymbol{f}_{ji}^{g_i} &= \boldsymbol{F}_i^{g_i} \cdot \boldsymbol{f} \\ \ddot{\boldsymbol{d}}_{i0}^{g_i} &= \boldsymbol{M}_i^{g_i} \cdot \boldsymbol{J}_{i0}^{g_i} \cdot \ddot{\boldsymbol{q}} + \boldsymbol{M}_i^{g_i} \cdot \hat{\boldsymbol{\gamma}}_{i0}^{g_I} \end{aligned}$$

En utilisant la relation $\ddot{\boldsymbol{q}} = \boldsymbol{L} \cdot \ddot{\boldsymbol{l}}$, et en associant aux variables libres des actions dynamiques h telles que $\boldsymbol{f} = \boldsymbol{L} \cdot \boldsymbol{h}$, l'équation (4.6) s'exprime par:

$$\dot{\boldsymbol{q}}^t \cdot \boldsymbol{F} \cdot \boldsymbol{h} = \dot{\boldsymbol{q}}^t \cdot \boldsymbol{E} \cdot \ddot{\boldsymbol{l}} + \dot{\boldsymbol{q}}^t \cdot \boldsymbol{d} \tag{4.7}$$

avec:

$$\begin{aligned}
\boldsymbol{F} &= \boldsymbol{L}^t \cdot \left(\sum_i \boldsymbol{J}_{i0}^{g_i t} \cdot \boldsymbol{F}_i^{g_i} \right) \cdot \boldsymbol{L} \\
\boldsymbol{E} &= \boldsymbol{L}^t \cdot \left(\sum_i \boldsymbol{J}_{i0}^{g_i t} \cdot \boldsymbol{M}_i^{g_i} \cdot \boldsymbol{J}_{i0}^{g_i} \right) \cdot \boldsymbol{L} \\
\boldsymbol{d} &= \sum_i \boldsymbol{v}_{i0}^{g_i} \cdot \boldsymbol{M}_i^{g_i} \cdot \hat{\boldsymbol{\gamma}}_{i0}^{g_i} - \sum_i \boldsymbol{v}_{i0}^{g_i} \cdot \boldsymbol{f}_i
\end{aligned}$$

Le vecteur $\boldsymbol{h}$ contient les forces associées aux degrés de liberté, $\boldsymbol{d}$ les forces extérieures comme et la partie connue des torseurs dynamiques (celle qui ne dépend que des vitesses). Cette dernière équation est une équation scalaire, qui traduit le théorème de l'énergie. Elle ne permet donc pas, à elle seule, de calculer toutes les inconnues.

Le principe de d'Alembert stipule que le théorème de l'énergie est vrai pour *toutes* les vitesses possibles compatibles avec les liaisons. L'équation devient donc:

$$\boldsymbol{F} \cdot \boldsymbol{h} = \boldsymbol{E} \cdot \ddot{\boldsymbol{l}} + \boldsymbol{d} \tag{4.8}$$

Nous disposons maintenant d'une équation dynamique portant sur les variables libres de notre structure cinématique. Cette équation prend implicitement en compte les contraintes cinématiques du mécanisme, et porte sur un nombre de variables plus réduit que le nombre de variables articulaires, puisque le nombre de variables libres est toujours inférieur au nombre de variables articulaires.

Par exemple, le mécanisme plan représenté sur la figure (4.7) comporte quatre variables articulaires mais seulement une variable libre.

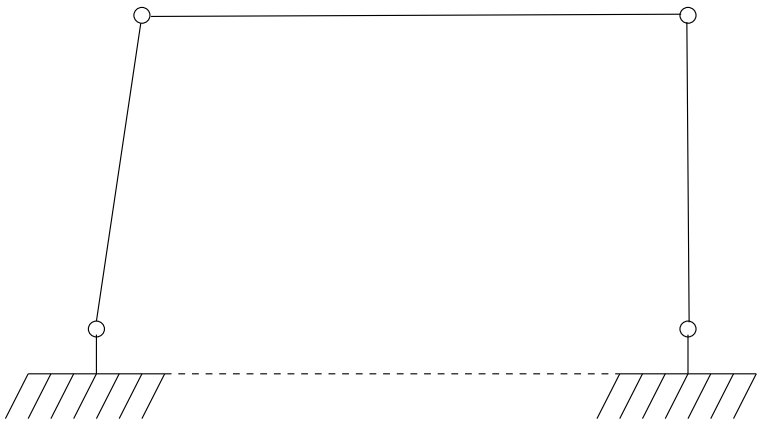


FIG. 4.7 - *Un mécanisme à un degré de liberté.*

L'équation dynamique (4.8) appliquée à ce mécanisme ne comporte donc qu'une seule inconnue d'accélération, et une inconnue de force associée. Les matrices $\boldsymbol{P_f}$ et $\boldsymbol{P_a}$ se

réduisent alors à des scalaires et l'équation dynamique est de la forme:

$$P_f.h + P_a.\ddot{l} = d$$

4.3.2 Applications

Nous disposons maintenant d'une équation (4.8) traduisant le comportement dynamique de notre système articulé. Cette équation matricielle se décompose en autant d'équations scalaires qu'il y a de variables cinématiques libres. Le nombre d'inconnues est quand à lui deux fois plus grand, puisqu'à chaque variable libre correspond une accélération et une force. Nous pouvons donc, pour chaque variable libre, imposer soit la force soit l'accélération. Ce résultat est bien connu depuis les travaux de Cohen[IC87] qui utilise, pour sa part, une autre formulation dynamique. Nous choisirons donc, selon les cas, d'imposer soit la force soit l'accélération à chaque variable libre.

Liaisons parfaites

On appelle liaison parfaite une articulation dont les efforts développent une puissance nulle. Ceci implique que les actions dynamiques sont orthogonales aux mouvements admissibles. La puissance est nulle, soit parce que le mouvement relatif est nul (mouvements interdits par la liaison), soit parce que les efforts sont nuls (ni frottements, ni actions motrices). Pour simuler le comportement d'un mécanisme composé de liaisons parfaites, on pose donc $\boldsymbol{h} = 0$, et l'équation (4.8) devient:

$$\boldsymbol{E} \cdot \ddot{\boldsymbol{l}} = -\boldsymbol{d}$$

Moteurs, ressorts, amortisseurs

Le terme "moteur" désigne ici un organe capable de générer une action dynamique donnée dans une direction de mouvement autorisé. Un ressort développe une action dynamique fonction des positions relatives, et un amortisseur une action dynamique fonction des vitesses relatives. Toutes ces actions étant connues lors du calcul des accélérations, on peut les considérer comme des actions extérieures. On se ramène ainsi au cas des liaisons parfaites.

Dynamique inverse

La dynamique inverse permet de garder un certain contrôle du mouvement des systèmes animés par modèles physiques. Etant donné un mouvement désiré, on souhaite connaître les forces nécessaires pour le produire. Nous sommes alors dans le cas où les mouvements sont connus, et l'équation dynamique devient:

$$\boldsymbol{F} \cdot \boldsymbol{h} = \boldsymbol{d} + \boldsymbol{E} \cdot \ddot{\boldsymbol{l}}$$

où $\ddot{\boldsymbol{l}}$ est connu. Notons que la valeur $\ddot{\boldsymbol{l}} = \boldsymbol{0}$ est valable si les mouvements désirés ont été pris en compte dans le processus de détermination des variables libres. Dans ce cas, les valeurs de $\ddot{\boldsymbol{l}}$ peuvent être utilisées pour optimiser certains critères comme par exemple essayer de minimiser les forces mises en jeu dans le mouvement.

4.4 Bilan

L'approche de cinématique des structures présentée dans ce chapitre permet un contrôle très large de l'animation des structures articulées. La décomposition par valeurs singulières de la matrice jacobienne fournit l'ensemble de toutes les combinaisons linéaires des variables articulaires satisfaisant les contraintes cinématiques. Contrairement aux pseudo-inverses à droite et à gauche, cette approche n'échoue jamais.

Dans le cas où les contraintes sont impossibles à satisfaire, on obtient la solution la moins mauvaise au sens des moindres carrés.

Dans le cas où il est possible de satisfaire les contraintes, on obtient une base orthonormée des mouvements autorisés qui définit les degrés de liberté de la structure globale. Cette base peut être utilisée pour tenter de satisfaire des contraintes moins prioritaires, et ce processus peut être répété jusqu'à épuisement des degrés de liberté.

Les degrés de liberté de la structure cinématique peuvent servir de variables dynamiques en utilisant le principe de d'Alembert. On peut ainsi calculer l'évolution de la structure respectant les lois de la physique et les contraintes cinématiques. Ceci permet la prise en compte de forces appliquées au système. Cette formulation dynamique basée sur la cinématique permet en outre d'effectuer des calculs de dynamique inverse.

L'algorithme de décomposition par valeurs singulières présente toutefois le grave défaut de reposer sur le traitement de matrices denses. Ceci implique donc une complexité quadratique en espace et cubique en temps en fonction du nombre de variables articulaires. Une telle complexité interdit en pratique l'application en temps "interactif" de l'approche présentée ici pour un nombre de variables articulaires supérieur à quelques dizaines.

Chapitre 5

Approche dynamique de l'animation

Ce chapitre présente l'approche dynamique de l'animation de solides par modèles physiques. Nous avons vu au chapitre 4 une façon de poser et résoudre les équations instantanées de la dynamique. Les inconnues étaient les forces et mouvements associés aux directions de mouvement autorisés. La méthode consistait à caractériser l'ensemble des mouvements possibles, puis à trouver dans cet ensemble une solution satisfaisant les principes de la dynamique.

L'approche présentée ici est tout autre. Elle consiste à ignorer dans un premier temps les contraintes cinématiques pour calculer comment évolueraient les solides soumis seulement à leur inertie et aux influences extérieures connues. On constate alors une violation des contraintes cinématiques, et on calcule les forces de blocage nécessaires au rétablissement de ces contraintes. Ces inconnues sont parfois appelées les multiplicateurs de Lagrange des contraintes cinématiques[Gle94, Bar96]. On peut entrevoir le bénéfice à retirer de cette approche par rapport à celle présentée au chapitre précédent: l'étape consistant à caractériser l'ensemble des mouvements autorisés n'est plus nécessaire. On se contente de trouver une seule solution cinématiquement admissible, en utilisant une méthode de recherche qui garantit le respect des lois de la dynamique.

Nous commencerons par établir l'équation de la dynamique qui relie les forces aux erreurs cinématiques à annuler. Nous présenterons ensuite différentes méthodes de résolution que nous avons implémentées et testées. Parmi celles-ci, une méthode que nous avons créée s'avère particulièrement efficace. Toutes ces méthodes seront évaluées selon les temps de calculs nécessaires, présentés à la fin du chapitre, mais aussi en fonction de leur capacité à résoudre des problèmes sous-contraints et sur-contraints. Nous terminerons par un bilan et des perspectives.

5.1 Écriture des équations

Nous nous placerons dans le cas d'un paramétrage en coordonnées absolues. Toutes les liaisons sont donc considérées comme des fermetures de chaînes (fig. 2.9). Les vecteurs $\dot{\boldsymbol{q}}_i$ et $\ddot{\boldsymbol{q}}_i$ représentent les vitesses et accélérations absolues des solides à leurs points d'origine. Nous considèrerons tout d'abord la prise en compte des influences extérieures comme le poids sur les solides. Nous poursuivrons en caractérisant l'erreur cinématique puis l'influence sur les solides des forces développées par les contraintes de mouvement relatif, pour déboucher sur la formulation générale de la dynamique des systèmes de solides par

l'approche des multiplicateurs de Lagrange.

5.1.1 Influences extérieures

Les influences extérieures appliquées aux solides incluent souvent le poids. L'influence du poids sur un solide i vaut:

$$\boldsymbol{f}_i^g = \begin{pmatrix} m.\boldsymbol{\mathcal{G}} \\ \mathbf{0} \end{pmatrix}$$

où g est le centre de masse du solide, m sa masse et $\boldsymbol{\mathcal{G}}$ la pesanteur.

On a souvent recours au frottement visqueux pour stabiliser les simulations, qui ont parfois tendance à diverger à cause des erreurs d'intégration. On peut lui donner une valeur arbitraire du type

$$\boldsymbol{f}_i^g = -\nu.\boldsymbol{v}_{i0}^g$$

où ν est un réel positif qu'on peut appeler facteur d'amortissement, $\boldsymbol{v}_{i0}^g$ la vitesse généralisée, à six dimensions, comme présentée au chapitre 2.

Dans la suite de ce chapitre, nous noterons $\boldsymbol{f}_i$ la somme des forces extérieures appliquées sur le solide i, et g_i le centre de masse de ce solide. Conformément au principe fondamental de la dynamique (eq. 2.27 et 2.25), nous pouvons écrire:

$$\boldsymbol{\gamma}_{i0}^{g_i} = \boldsymbol{M}_i^{g_i\,-1} \cdot \boldsymbol{f}_i^g$$

où

$$\ddot{\boldsymbol{x}}_i = \boldsymbol{\gamma}_{i0}^{O_i} = \boldsymbol{M}_i^{O_i\,-1} \cdot \boldsymbol{f}_i^{O_i} + \boldsymbol{\delta}_{i0}^{O_i g_i}$$

où O_i est l'origine du solide i.

Connaissant la vitesse et l'accélération de l'origine du repère soumis aux influences extérieures, nous pouvons maintenant calculer les erreurs cinématiques. Pour chaque articulation A_{ij} de centre L_{ij}, on calcule l'accélération absolue de i et j au point de liaison:

$$\boldsymbol{\gamma}_{i0}^{L_{ij}} = \boldsymbol{\nabla}_{L_{ij}O_i} \cdot \boldsymbol{\gamma}_{i0}^{O_i} \quad et \quad \boldsymbol{\gamma}_{j0}^{L_{ij}} = \boldsymbol{\nabla}_{L_{ij}O_j} \cdot \boldsymbol{\gamma}_{j0}^{O_j} \tag{5.1}$$

puis, en utilisant la relation (3.11), on exprime la dérivée de la vitesse relative au point de liaison. Nous noterons $\boldsymbol{e}_{ij}$ son opposé:

$$\boldsymbol{e}_{ij} \;=\; -\left(\dot{\boldsymbol{v}}_{ij}^{L_{ij}}\right)_{(ij)} \;=\; \gamma_{j0}^{L_{ij}} - \boldsymbol{\gamma}_{i0}^{L_{ij}} + \hat{\boldsymbol{\omega}}_{ij0}$$

Exprimer l'opposé de l'erreur cinématique permettra par la suite d'exprimer le système d'équations dynamiques sous la forme $\boldsymbol{D}\cdot\boldsymbol{f} = \boldsymbol{e}$ qui est la façon la plus courante d'exprimer un système d'équations linéaires, plutôt que $\boldsymbol{D}\cdot\boldsymbol{f} + \boldsymbol{e} = \mathbf{0}$ dans le cas contraire.

Le repère (ij) de dérivation est le repère intermédaire de laision. Selon le type d'articulation et la convention adoptée, le repère intermédiaire de liaison est lié au père ou au fils de l'articulation.

Nous pouvons maintenant construire un vecteur $\boldsymbol{e}$ comportant autant de composantes que le nombre total de directions indépendantes contraintes (blocages ou mobilités contraintes). La k-ième direction contrainte de l'articulation A_{ij} sera notée l_{ijk} et son axe, en dimension six, sera noté $\boldsymbol{a}_{ijk}$. Soit c_{ijk} la valeur de la contrainte cinématique suivant cette direction. Les composantes du vecteur $\boldsymbol{e}$ valent

$$\boldsymbol{e}[ijk] = c_{ijk} - \boldsymbol{a}_{ijk}^t \cdot \boldsymbol{e}_{ij}^{L_{ij}} \tag{5.2}$$

Le vecteur $\boldsymbol{e}$ rassemble donc toutes les erreurs cinématiques qui se produiraient en l'absence de forces de maintien des articulations.

5.1.2 Influence des articulations

Soit $\boldsymbol{f}_i^{O_i}$ le total des forces appliquées au solide i. D'après l'équation (2.27), une variation $\boldsymbol{\delta f}_i^{O_i}$ de cette force va engendrer une variation $\boldsymbol{\delta\gamma}_{i0}^{O_i}$ telle que:

$$\boldsymbol{\delta\ddot{x}}_i = \boldsymbol{\delta\gamma}_{i0}^{O_i} = \boldsymbol{M}_i^{O_i\,-1} \cdot \boldsymbol{\delta f}_i^{O_i}$$

Plus généralement, nous pouvons exprimer la variation $\boldsymbol{\delta\ddot{x}}$ des accélérations des solides en fonction de la variation $\boldsymbol{\delta f}$ de l'ensemble des forces articulaires par:

$$\boldsymbol{\delta\ddot{x}} = \boldsymbol{M}^{-1} \cdot \boldsymbol{F} \cdot \boldsymbol{\delta f}$$

où $\boldsymbol{M}^{-1}$ est une matrice creuse composée de blocs sur sa diagonale, chaque bloc étant la matrice de masse inverse à l'origine d'un solide, comme illustré en (5.3):

$$\boldsymbol{M}^{-1} = \begin{pmatrix} \boldsymbol{M}_1^{O_1\,-1} & & \\ & \ddots & \\ & & \boldsymbol{M}_n^{O_1\,-1} \end{pmatrix} \tag{5.3}$$

La matrice $\boldsymbol{F}$ relie les efforts de contraintes aux efforts sur les solides exprimés à leurs origines. L'exemple de la figure (5.1) représente un assemblage de trois solides.

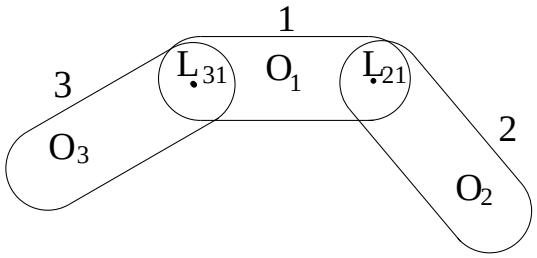


FIG. 5.1 - *Trois solides et deux liaisons*

La matrice $\boldsymbol{F}$ correspondant à une telle structure est:

$$\boldsymbol{F} = \begin{pmatrix} (\boldsymbol{F}_{12})_1 & (\boldsymbol{F}_{13})_1 \\ (\boldsymbol{F}_{12})_2 & \boldsymbol{0} \\ \boldsymbol{0} & (\boldsymbol{F}_{13})_3 \end{pmatrix}$$

où les $(\boldsymbol{F}_{ij})$ sont des matrices composées de vecteurs-colonnes tels que:

$$(\boldsymbol{F}_{ij})_j[][k] = \boldsymbol{\Delta}_{L_{ij}O_j} \cdot a_{ijk} \quad et \quad (\boldsymbol{F}_{ij})_i[][k] = -\boldsymbol{\Delta}_{L_{ij}O_i} \cdot a_{ijk}$$

5.1.3 Influence des solides

Une variation $\boldsymbol{\delta\ddot{q}}$ engendre une variation $\boldsymbol{\delta e}_{ij}$ telle que:

$$\boldsymbol{\delta e}_{ij} = (\boldsymbol{J}_{i0}^{L_{ij}} - \boldsymbol{J}_{j0}^{L_{ij}}) \cdot \boldsymbol{\delta\ddot{x}}$$

Plus généralement, une variation $\boldsymbol{\delta\ddot{x}}$ des accélérations des solides entraîne une variation $\boldsymbol{\delta e}$ des mouvements articulaires telle que:

$$\boldsymbol{\delta e} = \boldsymbol{J} \cdot \boldsymbol{\delta\ddot{x}} \tag{5.4}$$

La matrice $\boldsymbol{J}$ relie les accélérations des solides aux accélérations articulaires. Concernant l'exemple de la figure (5.1), cette matrice aurait la forme:

$$\boldsymbol{J} = \begin{pmatrix} \boldsymbol{J}_{211} & \boldsymbol{J}_{212} & \mathbf{0} \\ \boldsymbol{J}_{311} & \mathbf{0} & \boldsymbol{J}_{313} \end{pmatrix}$$

Vue comme une matrice de blocs, la matrice $\boldsymbol{J}$ comporte une ligne par articulation et une colonne par solide. Les seuls blocs non nuls sur la ligne de A_{ij} concerne les solides i et j. La matrice $\boldsymbol{J}$ est donc une matrice creuse. Cette propriété aura des conséquences importantes sur l'efficacité relative des différents algorithmes de résolution. Une autre conséquence est que cette matrice n'a pas forcément à être stockée explicitement. Les blocs qui la composent peuvent être des champs des articulations.

Les sous-matrices $\boldsymbol{J}_{iji}$ et $\boldsymbol{J}_{ijj}$ expriment les mouvements articulaires en fonction des mouvements des solides. La sous-matrice $\boldsymbol{J}_{iji}$ exprime l'influence du solide i sur le mouvement relatif au centre de liaison, tout comme $\boldsymbol{J}_{ijj}$ exprime l'influence du solide j sur cette même articulation. Elles sont composées de vecteurs-lignes dont les expressions valent:

$$\boldsymbol{J}_{iji}[k][*] = \boldsymbol{a}_{ijk}{}^t \cdot \boldsymbol{\nabla}_{O_i L_{ij}}{}^t \quad et \quad \boldsymbol{J}_{ijj}[k][*] = -\boldsymbol{a}_{ijk}{}^t \cdot \boldsymbol{\nabla}_{O_j L_{ij}}{}^t \tag{5.5}$$

Or, la relation $\boldsymbol{\Delta}_u{}^t = \boldsymbol{\nabla}_{-u}$ entraîne la propriété sympathique:

$$\boldsymbol{F} = \boldsymbol{J}^t \tag{5.6}$$

Nous noterons

$$\begin{aligned} \boldsymbol{\jmath}_{ijkj} &= \left(\boldsymbol{\nabla}_{O_j L_{ij}} \cdot \boldsymbol{a}_{ijk}\right)^t \\ \boldsymbol{\jmath}_{ijki} &= -\left(\boldsymbol{\nabla}_{O_i L_{ij}} \cdot \boldsymbol{a}_{ijk}\right)^t \end{aligned} \tag{5.7}$$

Ces vecteurs-lignes relient directement les contraintes aux solides, et seront beaucoup utilisés dans la suite de ce document. Les matrices $\boldsymbol{J}_{iji}$ et $\boldsymbol{J}_{ijj}$ sont respectivement l'union des $\boldsymbol{\jmath}_{ijki}$ et des $\boldsymbol{\jmath}_{ijkj}$

5.1.4 Équation dynamique

Les contraintes cinématiques devant être respectées, la question est de connaître les valeurs des forces dans les directions contraintes nécessaires pour rétablir les contraintes. Autrement dit, les forces de maintien doivent générer le mouvement relatif $-\boldsymbol{e}$.

On doit donc résoudre le système $\boldsymbol{J} \cdot \boldsymbol{M}^{-1} \cdot \boldsymbol{F} \cdot \boldsymbol{f} = \boldsymbol{e}$. Compte tenu de la propriété (5.6), le système à résoudre s'écrit généralement:

$$\boldsymbol{J} \cdot \boldsymbol{M}^{-1} \cdot \boldsymbol{J}^t \cdot \boldsymbol{f} = \boldsymbol{e} \tag{5.8}$$

On appellera *matrice dynamique* du système, notée $\boldsymbol{D}$, le produit

$$\boldsymbol{D} = \boldsymbol{J} \cdot \boldsymbol{M}^{-1} \cdot \boldsymbol{J}^t \tag{5.9}$$

Le système d'équation s'écrit alors

$$\boldsymbol{D} \cdot \boldsymbol{f} = \boldsymbol{e} \tag{5.10}$$

5.2 Méthodes cubiques

5.2.1 Résolution directe de la matrice dynamique

La forme particulière de la matrice dynamique (eq. 5.9) suggère l'algorithme d'écriture présenté sur la figure (5.2). On commence par numéroter toutes les contraintes cinématiques (blocages ou mobilités contraintes) du système. L'élément $\boldsymbol{D}[m][n]$ exprime l'influence d'une force dans l'axe de la contrainte n sur le mouvement relatif dans l'axe de la contrainte m. Le voisinage du solide i, noté $\mathcal{V}(i)$, rassemble toutes les articulations reliant i à un autre solide. Ces articulations contiennent des contraintes désignées par un entier qui représente leur rang dans la numérotation globale des contraintes. Le vecteur $\boldsymbol{\jmath}_{mi}$ représente l'influence de la contrainte m sur le solide i conformément à la convention décrite par la relation (5.7). L'indice m rassemble ici les trois premiers indices de la notation précédente.

$\boldsymbol{D} = \boldsymbol{0}$
pour chaque solide i:
  pour chaque contrainte n de $\mathcal{V}(i)$:
    pour chaque contrainte m de $\mathcal{V}(i)$:
      $\boldsymbol{D}[m][n] += \boldsymbol{\jmath}_{mi} \cdot {M_i^{O_i}}^{-1} \cdot {\boldsymbol{\jmath}_{ni}}^t$

FIG. 5.2 - *Écriture de la matrice dynamique*

La structure de la matrice dynamique appelle quelques commentaires. Nous savons que les matrices $\boldsymbol{J}$ et $\boldsymbol{M}^{-1}$ de l'équation (5.8) sont creuses et peuvent être stockées par blocs dans les structures de données des solides et articulations. Qu'en est-il de $\boldsymbol{D}$?

La matrice dynamique est de taille (N, N) où N est le nombre total de contraintes cinématiques. Elle décrit les interactions entre les différentes contraintes cinématiques. Au vu du pseudocode (5.2), il apparaît que les $\boldsymbol{D}[m][n]$ sont non nuls si les contraintes m et n portent sur le même solide.

Considérons deux exemples. La figure (5.3) montre un mécanisme composé de quatre solides et trois liaisons.

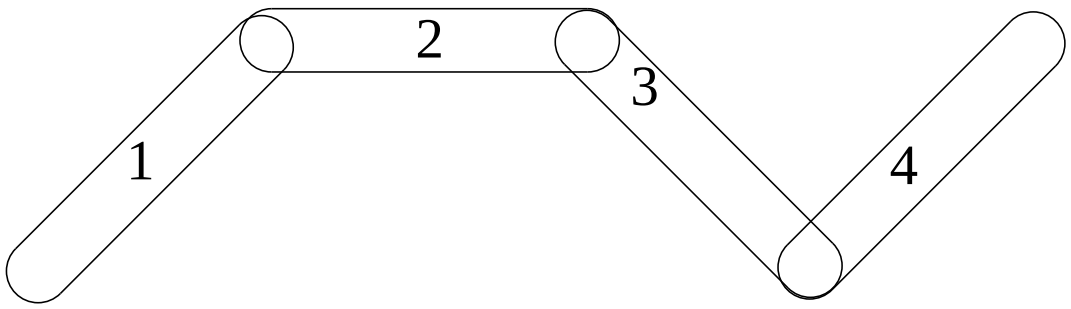


FIG. 5.3 - *Un mécanisme composé de quatre solides*

Les matrices $\boldsymbol{J}$ et $\boldsymbol{D}$ correspondantes ont pour structures:

$$\boldsymbol{J} = \begin{pmatrix} \boldsymbol{J}_{211} & \boldsymbol{J}_{212} & & \\ & \boldsymbol{J}_{322} & \boldsymbol{J}_{323} & \\ & & \boldsymbol{J}_{433} & \boldsymbol{J}_{434} \end{pmatrix} \quad et \quad \boldsymbol{D} = \begin{pmatrix} \boldsymbol{D}_{2121} & \boldsymbol{D}_{2132} & \\ \boldsymbol{D}_{2132}{}^{t} & \boldsymbol{D}_{3232} & \boldsymbol{D}_{4332} \\ & \boldsymbol{D}_{4332}{}^{t} & \boldsymbol{D}_{4343} \end{pmatrix}$$

La matrice dynamique du mécanisme possède des blocs nuls car les articulations 21 et 43 ne concernent aucun solide commun. Le mécanisme de la figure (5.4) ne possède pas cette caractéristique.

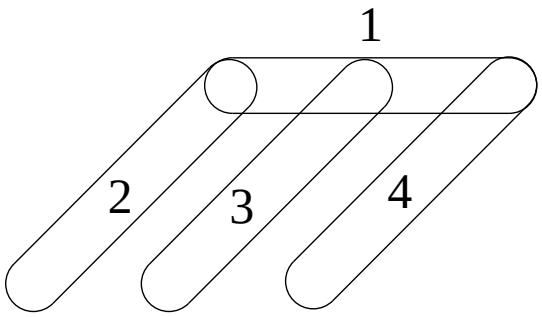


FIG. 5.4 - *Un autre mécanisme composé de quatre solides*

Les matrices $\boldsymbol{J}$ et $\boldsymbol{D}$ de ce dernier mécanisme ont pour structures:

$$\boldsymbol{J} = \begin{pmatrix} \boldsymbol{J}_{211} & \boldsymbol{J}_{212} & & \\ \boldsymbol{J}_{311} & & \boldsymbol{J}_{313} & \\ \boldsymbol{J}_{411} & & & \boldsymbol{J}_{414} \end{pmatrix} \quad et \quad \boldsymbol{D} = \begin{pmatrix} \boldsymbol{D}_{2121} & \boldsymbol{D}_{2131} & \boldsymbol{D}_{2141} \\ \boldsymbol{D}_{2131}{}^{t} & \boldsymbol{D}_{3131} & \boldsymbol{D}_{4131} \\ \boldsymbol{D}_{2141}{}^{t} & \boldsymbol{D}_{4131}{}^{t} & \boldsymbol{D}_{4141} \end{pmatrix}$$

La matrice $\boldsymbol{D}$ est pleine car toutes les articulations ont un solide en commun. Cet exemple prouve qu'on ne peut stocker $\boldsymbol{D}$ sous forme de matrice creuse dans le cas général. La stratégie consistant à calculer le produit des trois matrices nécessitera donc ensuite une résolution de matrice dense, dont la complexité en temps est cubique par rapport au nombre de contraintes cinématiques. La plus rapide est la décomposition de Cholesky[PTVF92] car la matrice est symétrique. Les limites de cette stratégie sont atteintes quand le temps de calcul, cubique, ou l'espace mémoire nécessaire, quadratique, deviennent prohibitifs. Une autre limite est que la matrice peut être singulière. Dans ce cas, la décomposition de Cholesky comme la décomposition LU échouent.

5.2.2 Matrices singulières

Si la matrice dynamique est forcément symétrique par construction, elle n'est hélas pas forcément définie positive. Cela signifie que selon les cas, les systèmes pourront avoir une infinité de solutions, ou pire, aucune solution. Tous ces cas peuvent survenir en présence de chaînes fermées. La figure (5.5) montre un exemple de système avec une infinité de solutions. Les erreurs cinématiques, symbolisées par des flèches, peuvent être annulées par une infinité de combinaisons de forces. Il suffit que la somme des forces horizontales soit égale à la valeur des contraintes divisée par la masse de la barre.

FIG. 5.5 - *Système possédant une infinité de solutions.*

De telles configurations sont un problème épineux pour les résolutions directes, car les méthodes les plus efficaces, Cholesky et LU, échouent sur les matrices singulières. On peut tout de même utiliser une méthode de pivot de Gauss, mais celles-ci présentent l'inconvénient d'être relativement lentes.

Un autre type de configuration singulière est celui où aucune solution n'existe. De tels systèmes mal formés peuvent se rencontrer pendant la résolution de contraintes de déplacements, pour le traitement des collisions notamment. Ils peuvent aussi provenir d'erreurs diverses dans la modélisation des contraintes et leur représentation numérique. La figure (5.6) présente un exemple de configuration sans solution.

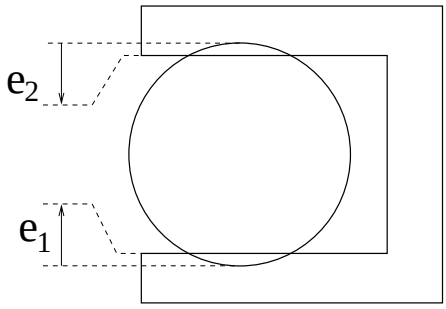


FIG. 5.6 - *Système ne possédant aucune solution.*

Dans ce cas, on peut encore envisager d'utiliser une méthode de pivot, qui s'arrête dès qu'une ligne nulle est rencontrée. L'effet obtenu est que certaines contraintes sont alors ignorées.

5.2.3 Approche locale

Les approches locales constituent une famille de techniques itératives de résolution. L'idée principale est de résoudre chaque contrainte indépendamment des autres, et de recommencer jusqu'à ce que l'erreur globale soit suffisamment petite. Ce genre d'approche est appliqué par Gascuel et Gascuel[GG94] pour résoudre des contraintes de petits déplacements. Nous les avons placées d'emblée parmi les méthodes cubiques bien que cette

caractéristique ne se dégage pas de façon évidente des algorithmes. Ce sont les tests qui la font apparaître clairement.

La valeur de la k-ième contrainte cinématique de l'articulation A_{ij} sera notée e_{ijk}, la force développée f_{ijk}, et le mouvement dû aux forces d'interaction x_{ijk}. Similairement, les contraintes, forces et déplacements aux articulations seront notés $\boldsymbol{e}_{ij}$, $\boldsymbol{f}_{ij}$ et $\boldsymbol{x}_{ij}$. Les déplacements $\boldsymbol{e}_{ij}$ et $\boldsymbol{x}_{ij}$ correspondent au mouvement de i par rapport à j, tandis que les forces $\boldsymbol{f}_{ij}$ correspondent aux forces de j sur i.

Les vecteurs $\boldsymbol{f}_{ij}$, $\boldsymbol{e}_{ij}$ et $\boldsymbol{x}_{ij}$ rassemblent, au niveau de l'articulation ij, les valeurs correspondant aux contraintes ijk de l'articulation. Les vecteurs $\boldsymbol{e}$, $\boldsymbol{f}$ et $\boldsymbol{x}$ représentent l'ensemble de toutes les contraintes, forces et déplacements aux articulations.

Le mouvement du solide i dû aux forces d'interaction sera noté $\boldsymbol{x}_i$. La somme des forces articulaires sur ce solide exprimée au point O_i sera notée $\boldsymbol{f}_i$ et sa matrice de masse inverse à l'origine sera notée $\boldsymbol{M}_i^{-1}$.

Résoudre une contrainte indépendamment des autres consiste à trouver la force associée qui satisfasse la contrainte. Nous appellerons *autoréponse* d'une contrainte le mouvement relatif généré par un effort unitaire dans la direction de cette contrainte. Si on note m_{ijk} l'autoréponse d'une direction contrainte f d'axe $\boldsymbol{a}_{ijk}$ entre les solides i et j, l'autoréponse vaut:

$$m_{ijk} = \boldsymbol{\jmath}_{ijki} \cdot \boldsymbol{M}_i^{O_i t} \cdot \boldsymbol{\jmath}_{ijki}{}^t$$

qui n'est autre que le terme diagonal de la matrice dynamique. La force nécessaire vaut alors $f_{ijk} = e_{ijk}/r_{ijk}$ où e_{ijk} est la contrainte cinématique, conformément aux notations de l'équation (5.10).

L'algorithme de résolution correspondant est présenté dans le pseudocode de la figure (5.7).

$\boldsymbol{f} = \boldsymbol{0}$
pour chaque solide i:
  $\boldsymbol{f}_i = \boldsymbol{0}$
tant que $\|\boldsymbol{e}\| > \epsilon$:
  pour chaque contrainte ijk:
    $f^+_{ijk} = e_{ijk}/m_{ijk}$
    $\boldsymbol{f}_i \mathrel{+}= f^+_{ijk}.\boldsymbol{\jmath}^t_{ijki}$
    $\boldsymbol{f}_j \mathrel{+}= f^+_{ijk}.\boldsymbol{\jmath}^t_{ijkj}$
  pour chaque solide i:
    $\boldsymbol{x}_i = \boldsymbol{M}_i^{-1} \cdot \boldsymbol{f}_i$
    $\boldsymbol{f}_i = \boldsymbol{0}$
  pour chaque articulation ij:
    $\boldsymbol{f}_{ij} \mathrel{+}= f^+_{ij}$
    $\boldsymbol{e}_{ij} \mathrel{-}= \boldsymbol{J}^t_{iji} \cdot \boldsymbol{x}_i + \boldsymbol{J}^t_{ijj} \cdot \boldsymbol{x}_j$

FIG. 5.7 - *Résolution locale itérative*

La question soulevée par cette approche, proposée par certains auteurs[GG94], est de savoir si une telle résolution itérative converge vers la solution de l'équation dynamique. En effet, à chaque itération, les influences entre contraintes ne sont pas prises en compte

et une nouvelle erreur cinématique existera à l'itération suivante. Nous nous proposons de montrer ici que cet algorithme équivaut à effectuer le produit de la contrainte cinématique par le développement infini de l'inverse de la matrice dynamique. On en déduira des idées sur la convergence de l'algorithme.

L'algorithme génère une suite de vecteurs-forces que nous noterons $\boldsymbol{f}_{(i)}$, où i représente le nombre de passages (diminué de un) dans la boucle principale de l'algorithme. Le vecteur $\boldsymbol{f}^{(i)}$ représente la force cumulée à l'issue de l'itération i, et $\boldsymbol{f}_{(i)}$ l'incrément de force à l'itération i. On a donc la relation: $\boldsymbol{f}^{(i+1)} = \boldsymbol{f}^{(i)} + \boldsymbol{f}_{(i+1)}$

Nous pouvons représenter la matrice dynamique sous la forme:

$$\boldsymbol{D} = \boldsymbol{K} \cdot (\mathbf{1} - \boldsymbol{B})$$

où la matrice $\boldsymbol{K}$ est la diagonale de $\boldsymbol{D}$, la matrice $\mathbf{1}$ représente l'identité, et $\boldsymbol{B}$ les termes non diagonaux de $\boldsymbol{D}$, divisés par les valeurs de la diagonale. Effectuer une résolution locale des contraintes revient à effectuer le produit $\boldsymbol{f}_{(i)} = \boldsymbol{K}^{-1} \cdot \boldsymbol{e}_{(i)}$ où $\boldsymbol{e}_{(i)}$ représente la contrainte cinématique à l'itération i.

On peut montrer la relation de récurrence suivante:

$$si \quad \boldsymbol{f}^{(i)} = \sum_{j=0}^{i} \boldsymbol{B}^j \boldsymbol{K}^{-1} \boldsymbol{e} \quad alors \quad \boldsymbol{f}^{(i+1)} = \sum_{j=0}^{i+1} \boldsymbol{B}^j \cdot \boldsymbol{K}^{-1} \boldsymbol{e}$$

où $\boldsymbol{B}^j$ représente la matrice $\boldsymbol{B}$ à la puissance j, et $\boldsymbol{B}^0 = \mathbf{1}$.

preuve:

$$\begin{aligned} \boldsymbol{f}_{(i+1)} &= \boldsymbol{K}^{-1}(\boldsymbol{e} - \boldsymbol{D}\boldsymbol{f}^{(i)}) \\ &= \boldsymbol{K}^{-1}\boldsymbol{e} - \textstyle\sum_{j=0}^{i}(\mathbf{1} - \boldsymbol{B})\boldsymbol{B}^j\boldsymbol{K}^{-1}\boldsymbol{e} \\ &= \boldsymbol{K}^{-1}\boldsymbol{e} - \textstyle\sum_{j=0}^{i}(\boldsymbol{B}^j\boldsymbol{K}^{-1}\boldsymbol{e} - \boldsymbol{B}^{j+1}\boldsymbol{K}^{-1}\boldsymbol{e}) \\ &= \boldsymbol{B}^{i+1}\boldsymbol{K}^{-1}\boldsymbol{e} \end{aligned}$$

On a donc bien: $\boldsymbol{f}^{(i+1)} = \boldsymbol{f}^{(i)} + \boldsymbol{f}_{(i+1)} = \sum_{j=0}^{i+1} \boldsymbol{B}^i \cdot \boldsymbol{K}^{-1}\boldsymbol{e}$ □

D'autre part, le terme initial de la récurrence est vérifié. En effet, nous avons bien, après la première itération:

$$\boldsymbol{f}_{(0)} = \boldsymbol{K}^{-1}e = \boldsymbol{B}^0\boldsymbol{K}^{-1}e$$

La force calculée par l'algorithme est donc:

$$\boldsymbol{f} = (\mathbf{1} + \boldsymbol{B} + \boldsymbol{B}^2 + \ldots)\boldsymbol{K}^{-1}\boldsymbol{e}$$

On reconnaît dans cette expression le développement de l'inverse de $(\mathbf{1} - \boldsymbol{B})$ sous la forme:

$$(\mathbf{1} - \boldsymbol{B})^{-1} = (\mathbf{1} + \boldsymbol{B} + \boldsymbol{B}^2 + \ldots)$$

Ceci amène quelques remarques:

- cet algorithme ne fournit pas de solution exacte;
- le taux de convergence dépend des valeurs propres de $\boldsymbol{B}$;
- des valeurs propres supérieures à 1 peuvent faire échouer l'algorithme.

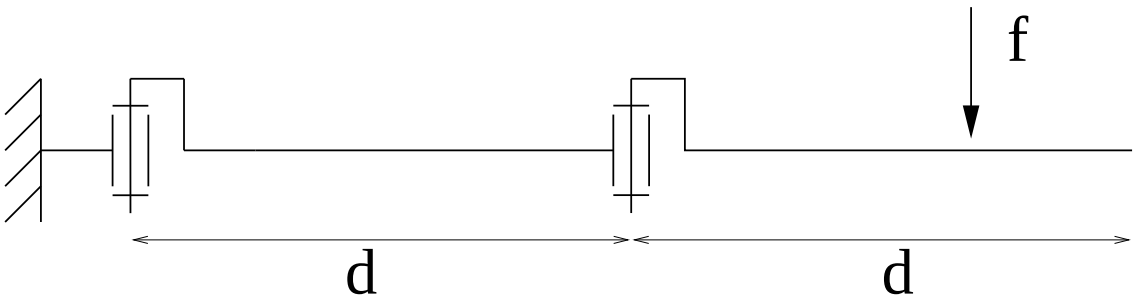


FIG. 5.8 - *Une configuration problematique pour les méthodes locales*

La figure (5.8) montre l'exemple de deux barres articulées. La première est accrochée au repère absolu, la deuxième est accrochée à la première. Les barres sont de masse 1 et d'inertie **1**, centrées à mi-longueur. Si nous appliquons sur ce mécanisme un chargement $\boldsymbol{f}$ parallèlement aux axes des pivots, l'ensemble doit rester immobile.

La convergence ne se produit que si la longueur d des barres est inférieure à 1 mètre. Si $d = 1$, l'algorithme tombe dans un cycle sans fin. Pour une longueur supérieure à un, il diverge. Une masse trop concentrée autour du barycentre provoque un mauvais conditionnement du système (fig. 5.9). Une barre homogène fournit un conditionnement correct.

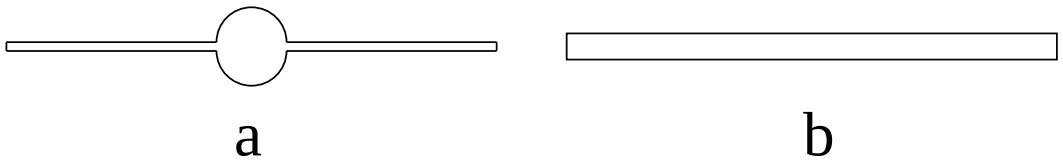


FIG. 5.9 - *(a): barre mal conditionnée. (b): barre bien conditionnée*

En pratique, cet algorithme nécessite souvent de moduler l'incrément de force à chaque itération par un facteur inférieur à un, déterminé par l'utilisateur. Le taux de convergence dépend fortement du mécanisme.

Doubler la taille du système multiplie par quatre le nombre d'itérations requises pour diviser l'erreur par un facteur donné. La durée de chaque itération étant proportionnelle à la taille du système, le temps de calcul total se voit donc multiplié par huit. Nous pouvons en déduire que la complexité de cet algorithme est cubique en temps. Son avantage sur la décomposition de Cholesky est qu'il ne nécessite pas de stocker une matrice dynamique, la complexité en espace reste donc linéaire.

5.2.4 Approche locale à propagation immédiate

L'approche locale à propagation immédiate se base sur la même idée que la précédente mais applique les corrections de forces aussitôt après les avoir calculées. Les contraintes cinématiques de toutes les articulations des solides concernés sont immédiatement mises à jour. Cette approche s'apparente à la méthode de Jacobi pour les équations de diffusion traitées par relaxation de contraintes[PTVF92]. L'algorithme correspondant est donné dans le pseudocode de la figure (5.10).

$\boldsymbol{f} = \mathbf{0}$
tant que $\|\boldsymbol{e}\| > \epsilon$:
  pour chaque contrainte ijk:
    $f^+_{ijk} = e_{ijk}/m_{ijk}$
    $f_{ijk} \mathrel{+}= f^+_{ijk}$
    $\boldsymbol{x}_i \mathrel{+}= f^+_{ijk}.\boldsymbol{M}_i^{-1} \cdot \boldsymbol{\jmath}_{ijki}{}^t$
    $\boldsymbol{x}_j \mathrel{+}= f^+_{ijk}.\boldsymbol{M}_j^{-1} \cdot \boldsymbol{\jmath}_{ijkj}{}^t$
    pour chaque A_{il} de $\mathcal{V}(i)$:
      $\boldsymbol{e}_{il} \mathrel{-}= \boldsymbol{J}_{il} \cdot \boldsymbol{x}_i$
    pour chaque A_{jl} de $\mathcal{V}(j)$:
      $\boldsymbol{e}_{jl} \mathrel{-}= \boldsymbol{J}_{jl} \cdot \boldsymbol{x}_j$

FIG. 5.10 - *Résolution locale à propagation immédiate*

Cet algorithme nécessite en général deux fois moins d'itérations que le précédent. Toutefois, les itérations sont plus coûteuses, ce qui compense à peu près le temps gagné. Le véritable avantage de cette approche sur la précédente est d'être moins sensible aux problèmes de conditionnement. Sur l'exemple de la figure (5.8), il converge bien pour des longueurs dépassant dix mètres. Aucune divergence n'est observée pour des longueurs supérieures, mais la convergence est extêmement lente. L'approche utilisant la propagation immédiate semble donc nettement préférable à l'approche locale simple.

Le comportement en présence de configurations singulières est également différent. Sur l'exemple de la figure (5.11), chaque résolution locale satisfait simultanément les deux contraintes. L'approche locale simple va donc appliquer une correction deux fois trop importante, et la barre va osciller sans fin entre deux positions fausses de part et d'autre des points d'attache. La résolution avec propagation immédiate, au contraire, s'arrête après la première résolution locale puisque toutes les contraintes seront alors satisfaites. La force portera sur une seule contrainte, ce qui n'est pas très satisfaisant du point de vue "esthétique", mais équivalent à une force répartie entre les deux contraintes du point de vue dynamique.

FIG. 5.11 - *Oscillation infinie de la résolution locale itérative.*

Il existe toutefois des cas où l'approche simple a un comportement plus satisfaisant que l'approche à propagation immédiate. La figure (5.12) montre l'exemple de contraintes incompatibles portant sur une même barre. L'une tend à déplacer la barre tandis que l'autre tend à la garder en place. Dans ce cas, l'approche simple tend vers un compromis alors que l'approche à propagation immédiate oscille d'une contrainte satisfaite à l'autre, en accroissant les forces cumulées à chaque itération.

De telles contraintes incompatibles peuvent exister quand on traite les collisions de façon simultanée.

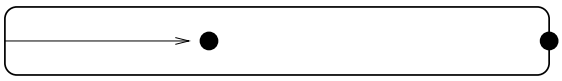


FIG. 5.12 - *Une barre soumise à deux contraintes incompatibles*

5.2.5 Approche semi-locale

On peut accélérer la résolution dynamique par approche locale en résolvant articulation par articulation au lieu de résoudre contrainte par contrainte. Cela a pour effet d'allonger la durée des itérations mais d'en réduire grandement le nombre. En pratique, cette approche est plus rapide. Le comportement demeure le même en présence de configurations singulières.

La résolution commence par le calcul d'une matrice de masse associée à chaque articulation, qui correspond à l'inverse de l'autoréponse dans l'approche locale:

$$\boldsymbol{M}_{ij} = (\boldsymbol{J}_{ijj} \cdot \boldsymbol{M}_j^{-1} \cdot \boldsymbol{J}_{ijj}^t + \boldsymbol{J}_{iji} \cdot \boldsymbol{M}_i^{-1} \cdot \boldsymbol{J}_{iji}^t)^{-1}$$

Puis on inverse ces matrices, et les corrections semi-locales de contraintes peuvent être effectuées par: $\boldsymbol{f}_{ij} = \boldsymbol{M}_{ij}^{-1} \cdot \boldsymbol{x}_{ij}$. Les figures (5.13) et (5.14) donnent les pseudocodes pour l'approche simple et l'approche par propagation immédiate de la résolution semi-locale.

```
f = 0
pour chaque solide i:
    f_i = 0
tant que ||e|| > ε:
    pour chaque articulation A_ij:
        f+_ij = M_ij · e_ij
        f_i  += J^t_iji · f+_ij
        f_j  += J^t_ijj · f+_ij
    pour chaque solide i:
        x_i = M_i^-1 · f_i
        f_i = 0
    pour chaque articulation A_ij:
        e_ij  -= J_iji · x_i + J_ijj · x_j
        f_ij  += f+_ij
```

FIG. 5.13 - *Approche semi-locale*

5.3 Méthode quadratique: le gradient conjugué

La méthode du gradient conjugué[PTVF92, Gle94] consiste à trouver une solution au système d'équations $\boldsymbol{A} \cdot \boldsymbol{x} = \boldsymbol{b}$ de N équations à N inconnues en minimisant itérativement la fonction scalaire $E(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x} \cdot \boldsymbol{A} \cdot \boldsymbol{x} - \boldsymbol{b} \cdot \boldsymbol{x}$. En effet, cette fonction est minimale quand son gradient est nul, c'est à dire $\boldsymbol{A} \cdot \boldsymbol{x} - \boldsymbol{b} = \boldsymbol{0}$.

$\boldsymbol{f} = \mathbf{0}$
tant que $\|\boldsymbol{e}\| > \epsilon$:
  pour chaque articulation A_{ij}:
    $\boldsymbol{f}_{ij}^{+} = M_{ij} \cdot \boldsymbol{e}_{ij}$
    $\boldsymbol{x}_i = \boldsymbol{M}_i^{-1} \cdot \boldsymbol{f}_{ij}^{+}$
    $\boldsymbol{x}_j = \boldsymbol{M}_j{-}1 \cdot \boldsymbol{f}_{ij}^{+}$
    $\boldsymbol{f}_{ij} \mathrel{+}= \boldsymbol{f}_{ij}^{+}$
    pour chaque articulation A_{ik} de $\mathcal{V}$(i):
      $\boldsymbol{e}_{ik} \mathrel{-}= \boldsymbol{J}_{iki} \cdot \boldsymbol{x}_i$
    pour chaque articulation A_{jk} de $\mathcal{V}$(j):
      $\boldsymbol{e}_{jk} \mathrel{-}= \boldsymbol{J}_{jkj} \cdot \boldsymbol{x}_j$

FIG. 5.14 - *Approche semi-locale à propagation immédiate*

L'algorithme génère une série de tentatives $\boldsymbol{x}^{(i)}$ telles que $\boldsymbol{x}^{(i+1)}$ minimise E sur l'espace affine

$$\boldsymbol{x}^{(0)} + \left(\boldsymbol{p}^{(0)}, \boldsymbol{p}^{(1)}, \ldots, \boldsymbol{p}^{(i)}\right)$$

où les $\boldsymbol{p}^{(k)}$ représentent les précédentes directions de recherche. L'intérêt est que si les $\boldsymbol{p}^{(k)}$ sont linéairement indépendants, la dimension de l'hyperplan $\boldsymbol{x}^{(0)} + \left(\boldsymbol{p}^{(0)}, \boldsymbol{p}^{(1)}, \ldots, \boldsymbol{p}^{(i)}\right)$ augmente de un à chaque itération. Ceci implique une convergence théorique en N itérations.

Une particularité remarquable de cet algorithme est qu'il n'utilise pas directement la matrice $\boldsymbol{A}$ mais uniquement son produit par un vecteur. Suivant les variantes et le type de matrice, un ou deux produits sont nécessaires à chaque itération. Ceci implique que si nous sommes capables d'effectuer le produit matriciel en temps linéaire, l'algorithme converge vers la solution en temps quadratique. Si la matrice est dense, le produit s'effectue en temps quadratique et la méthode n'a pas d'intérêt par rapport aux méthodes classiques de résolution de matrices denses[PTVF92].

La bonne nouvelle est que nous sommes capables, dans le cas des équations dynamiques qui nous intéressent, d'effectuer le produit en temps linéaire. Effectuer le produit consiste à calculer les déplacements résultant d'un ensemble de forces articulaires. Nous avons vu que la matrice dynamique $\boldsymbol{D}$ est dense, mais qu'elle est le produit de trois matrices creuses. Son produit par un vecteur peut donc s'effectuer par trois produit successifs, chacun en temps linéaire, comme détaillé dans le pseudocode (5.15). Le premier produit matriciel consiste pour chaque solide i à accumuler dans $\boldsymbol{f}_i$ les forces des différentes articulations le concernant. Pour cela, $\boldsymbol{f}_i$ doit être initialisé à zéro au préalable. La prise en compte des forces peut être faite en parcourant successivement les solides ou les articulations, c'est cette dernière option qui est retenue dans le pseudocode. Une fois les forces connues, le mouvement de chaque solide peut être calculé. On peut en déduire par un nouveau parcours des articulations les déplacements articulaires.

Une fois la procédure de produit matriciel implémentée, on peut utiliser un algorithme standard de gradient conjugué[PTVF92].

Il existe plusieurs variantes de l'algorithme du gradient conjugué. La plus simple traite

pour chaque solide i:
$\boldsymbol{f}_i \;=\; \mathbf{0}$
pour chaque articulation A_{ij}:
$\boldsymbol{f}_i \;+\!=\; \boldsymbol{J}^t_{iji} \cdot \boldsymbol{f}_{ij}$
$\boldsymbol{f}_j \;+\!=\; \boldsymbol{J}^t_{ijj} \cdot \boldsymbol{f}_{ij}$
pour chaque solide i:
$\boldsymbol{x}_i \;=\; \boldsymbol{M}_i^{-1} \cdot \boldsymbol{f}_i$
pour chaque articulation A_{ij}:
$\boldsymbol{x}_{ij} \;=\; \boldsymbol{J}_{iji} \cdot \boldsymbol{x}_i + \boldsymbol{J}_{ijj} \cdot \boldsymbol{x}_j$

FIG. 5.15 - *Produit matriciel décomposé réalisant* $\boldsymbol{x} = \boldsymbol{D} \cdot \boldsymbol{f}$

les matrices symétriques définies positives. Or si $\boldsymbol{D}$ est toujours symétrique, elle n'est pas définie positive dans certains cas avec chaînes fermées. On a alors recours au gradient *biconjugué*, qui effectue à chaque itération un produit matriciel supplémentaire, avec la transposée de la matrice. Notre matrice dynamique étant toujours symétrique, on utilise la même procédure de produit. L'algorithme effectue donc deux produits matriciels à chaque itération. Il est alors capable de résoudre des configurations singulières possédant une infinité de solutions.

Une dernière variante est l'option "résidu minimum". Au prix d'un produit matriciel supplémentaire préalable aux itérations, l'algorithme minimise alors la fonction $E'(\boldsymbol{x}) = (\boldsymbol{A} \cdot \boldsymbol{x} - \boldsymbol{b})^2$. Cette dernière version est la plus robuste de toutes car elle traite toutes les configurations singulières. Dans les cas ne comportant aucune solution, la minimisation de E' fournit le meilleur compromis au sens des erreurs cinématiques.

5.4 Une méthode linéaire

De nombreuses méthodes de complexité linéaire ont été proposées. Elles sont toutes limitées aux graphes acycliques, c'est à dire aux mécanismes ne comportant pas de chaînes fermées. Elles sont d'une grande utilité en robotique où ce type de configuration (bras de robot, etc.) est fréquent. Elles sont généralement appliquées à un paramétrage en coordonnées articulaires.

Récemment, Baraff en a proposé une exprimée en coordonnées absolues, ainsi qu'une extension permettant de traiter les chaînes fermées. L'extension présentée est de complexité cubique, mais la "cubicité" reste limitée au traitement des chaînes fermées. Cette méthode mixte est donc très efficace pour les systèmes possédant un nombre faible de fermetures de chaînes par rapport au nombre total de contraintes.

Nous présentons ici les grandes lignes de la méthode linéaire en coordonnées absolues. Les principaux pseudocodes utiles sont fournis en annexe B. La figure (5.16) représente un mécanisme acylique et le graphe cinématique associé. La racine du graphe est à droite de la figure. Les mouvements libres ne comportent aucune contrainte de mouvement et ne font donc pas partie du graphe utilisé par l'algorithme.

Les données d'entrée de la résolution sont les contraintes de déplacements à satis-

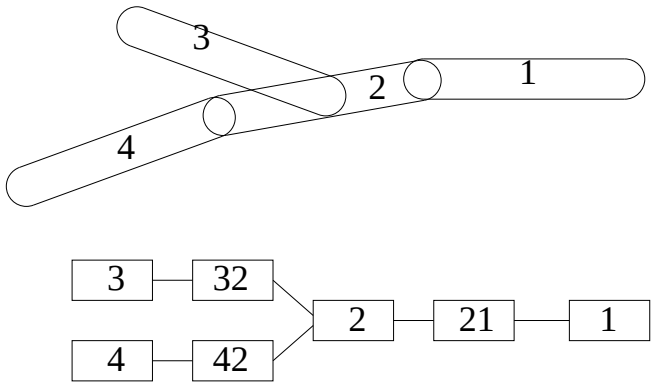


FIG. 5.16 - *Un mécanisme acyclique*

faire, c'est à dire le vecteur $\boldsymbol{e}$ de l'équation (5.10). La résolution repose sur une écriture matricielle différente du problème. Au lieu d'exprimer $\boldsymbol{J} \cdot \boldsymbol{M}^{-1} \cdot \boldsymbol{J}^t \cdot \boldsymbol{f} = \boldsymbol{e}$, on utilise une formulation équivalente qui a l'avantage d'être toujours creuse contrairement à la précédente:

$$\begin{pmatrix} \boldsymbol{M} & -\boldsymbol{J}^t \\ -\boldsymbol{J} & \boldsymbol{0} \end{pmatrix} \cdot \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{f} \end{pmatrix} = \begin{pmatrix} \boldsymbol{0} \\ -\boldsymbol{e} \end{pmatrix} \tag{5.11}$$

Cette formulation est celle qu'emploient Isaacs et Cohen[IC88] pour résoudre des contraintes complexes, mais dans le cadre d'un paramétrage en coordonnées relatives. Leurs sous-matrices ne sont donc pas creuses. Le vecteur $\boldsymbol{x}$ rassemble les mouvements des solides dûs aux forces de rétablissement des contraintes, et la première ligne donne: $\boldsymbol{x} = \boldsymbol{M}^{-1} \cdot \boldsymbol{J}^t \cdot \boldsymbol{f}$. Intuitivement, elle signifie que les mouvements de correction des solides sont dus aux forces associées aux contraintes. La deuxième ligne $\boldsymbol{J} \cdot \boldsymbol{x} = \boldsymbol{e}$ signifie que les mouvements des solides produisent les corrections voulues sur les contraintes.

On peut se faire une autre idée de cette résolution en suivant son algorithme, qui procède en deux parcours du graphe. Chaque articulations est munie des deux habituelles matrices jacobiennes ($\boldsymbol{J}_{iji}$ et $\boldsymbol{J}_{ijj}$) mais aussi d'une matrice de masse de la dimension du nombre de contraintes de l'articulation. Le premier parcours de graphe s'effectue de gauche à droite (sur la figure 5.16). Lors de ce parcours, chaque élément (solide ou articulation) répond à deux questions:

- quel serait mon mouvement s'il n'y avait rien à ma droite (compte tenu de ce qu'il y a à ma gauche);

- comment réagirais-je à un mouvement à ma droite (compte tenu de ce qu'il y a à ma gauche).

L'effet de ce premier parcours est de propager des informations jusqu'à la racine du graphe (à droite). A la fin du parcours, le mouvement de la racine est connu, puisqu'il n'y a effectivement rien à droite, et qu'on a pris en compte tout le reste du graphe. On effectue alors un parcours vers la gauche, en propageant les mouvements réels.

5.5 Méthodes mixtes

Les méthodes mixtes consistent à découper le problème dynamique en deux. Les contraintes de fermeture de chaînes sont traitées à part des contraintes acycliques. Nous en présentons ici deux approches différentes.

Considérons le problème des fermetures. La figure (5.17) montre l'exemple de deux chaînes fermées non résolues, dont les erreurs sont représentées par $\boldsymbol{e}_1 = (e_{1x}, e_{1y})^t$ et $\boldsymbol{e}_2 = (e_{2x}, e_{2y})^t$. La méthode de Baraff consiste à exprimer une matrice $\boldsymbol{D}_c$ telle que:

$$\begin{pmatrix} e_{1x} \\ e_{1y} \\ e_{2x} \\ e_{2y} \end{pmatrix} = \boldsymbol{D}_c \cdot \begin{pmatrix} f_{1x} \\ f_{1y} \\ f_{2x} \\ f_{2y} \end{pmatrix}$$

tout en maintenant respectées les contraintes acycliques.

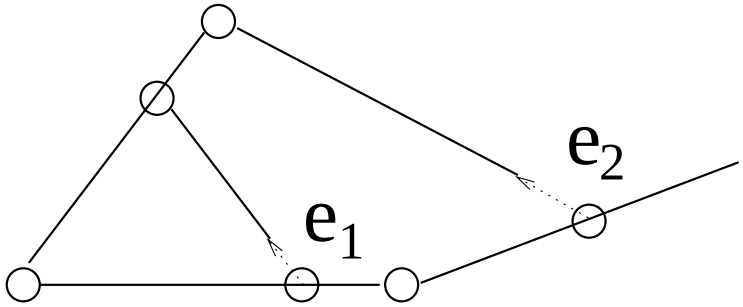


FIG. 5.17 - *Deux chaînes fermées à résoudre.*

Pour mieux comprendre, nous devons réécrire le système d'équations dynamiques du système sous la forme:

$$\begin{pmatrix} \boldsymbol{D}_{aa} & \boldsymbol{D}_{ac} \\ \boldsymbol{D}_{ca} & \boldsymbol{D}_{cc} \end{pmatrix} \cdot \begin{pmatrix} \boldsymbol{f}_a \\ \boldsymbol{f}_c \end{pmatrix} = \begin{pmatrix} \boldsymbol{e}_a \\ \boldsymbol{e}_c \end{pmatrix}$$

L'indice a concerne les contraintes acycliques et les forces développées par ces contraintes. La méthode linéaire présentée au chapitre précédent résoud $\boldsymbol{D}_{aa} \cdot \boldsymbol{f}_a = \boldsymbol{e}_a$ en temps linéaire. L'indice c se rapports aux contraintes de fermetures de chaînes. Les forces satisfont les équations si et seulement si:

$$\boldsymbol{D}_{aa} \cdot \boldsymbol{f}_a + \boldsymbol{D}_{ac} \cdot \boldsymbol{f}_c = \boldsymbol{e}_a \tag{5.12}$$

$$\boldsymbol{D}_{ca} \cdot \boldsymbol{f}_a + \boldsymbol{D}_{cc} \cdot \boldsymbol{f}_c = \boldsymbol{e}_c \tag{5.13}$$

En substituant dans l'équation (5.13) l'expression de $\boldsymbol{f}_a$ extraite de l'équation (5.12), nous obtenons:

$$(\boldsymbol{D}_{cc} - \boldsymbol{D}_{ca} \cdot \boldsymbol{D}_{aa}^{-1} \cdot \boldsymbol{D}_{ac}) \cdot \boldsymbol{f}_c = \boldsymbol{e}_c - \boldsymbol{D}_{ca} \cdot \boldsymbol{D}_{aa}^{-1} \cdot \boldsymbol{e}_a \tag{5.14}$$

La matrice $\boldsymbol{D}_{aa}$ est forcément inversible car c'est la matrice dynamique d'un mécanisme acyclique. Cela revient à dire que n'importe quelle erreur initiale sur les contraintes peut

être résolue si le graphe est acyclique. Nous avons trouvé notre matrice dynamique $\boldsymbol{D}_c$ qui vaut:

$$\boldsymbol{D}_c = \boldsymbol{D}_{cc} - \boldsymbol{D}_{ca} \cdot \boldsymbol{D}_{aa}^{-1} \cdot \boldsymbol{D}_{ac}$$

Si nous résolvons l'équation (5.14), nous connaissons alors la valeur des efforts aux fermetures de boucle. Les efforts aux contraintes acycliques se déduisent alors par substitution dans l'équation (5.12) puis par appel à l'algorithme de résolution acyclique.

5.5.1 Résolution mixte directe

Nous ne possédons pas de représentation explicite des matrices entrant dans la composition de $\boldsymbol{D}_c$. L'approche de Baraff consiste à écrire colonne par colonne la matrice $\boldsymbol{D}_c$. En effet, la k-ième colonne de $\boldsymbol{D}_c$ est la réponse du mécanisme, compte tenu des contraintes acycliques, à un effort unitaire selon la k-ième composante de $\boldsymbol{f}_c$. Nous obtenons une matrice dense carrée de dimension égale au nombre de contraintes de fermeture de chaînes.

Supposons que nous disposons d'une procédure $resolutionAcyclique(\boldsymbol{x}, \boldsymbol{f})$ qui calcule et écrit dans $\boldsymbol{f}$ les forces nécessaires pour effectuer les mouvements spécifiés dans $\boldsymbol{x}$. Cette méthode concerne uniquement les contraintes acycliques. Seules les contraintes acycliques sont satisfaites, les forces acycliques sont écrites dans $\boldsymbol{f}$ et les forces non acycliques de $\boldsymbol{f}$ sont renvoyées inchangées. Nous supposons également que nous avons n_a contraintes acycliques numérotées de 0 à $n_a - 1$, et n_c contraintes de fermetures numérotées de n_a à $n_a + n_c - 1$.

L'écriture de la matrice $\boldsymbol{D}_c$ s'effectue en appelant la procédure $ecrireMatriceFermetures(\boldsymbol{D}_c)$ dont pseudocode est fourni sur la figure (5.18).

```
pour i de 0 à n_c − 1:
    f = 0;     f[n_a + i] = 1
    x = D · f
    resolutionAcyclique(x, f)
    x  −=  D · f
    pour j de 0 à n_c − 1:
        D_c[i][j]  =  x[n_a + j]
```

FIG. 5.18 - $ecrireMatriceFermetures(\boldsymbol{D}_c)$

Notons que le premier produit $\boldsymbol{D} \cdot \boldsymbol{f}$ de la boucle concerne un vecteur force dont une seule composante est non nulle. On peut donc limiter le calcul de $\boldsymbol{x}$ aux contraintes concernant les deux solides qui subissent la force. D'autre part, après l'exécution de $\boldsymbol{x}\ -= \boldsymbol{D} \cdot \boldsymbol{f}$, les n_a premières composantes de $\boldsymbol{x}$ doivent être nulles puisque la résolution acyclique a compensé les erreurs acycliques.

L'algorithme utilisant la matrice $\boldsymbol{D}_c$ consiste en plusieurs étapes:

- résoudre les seules contraintes acycliques
- mettre à jour l'erreur aux chaines fermées et calculer $\boldsymbol{D}_c$;

- calculer les forces de fermetures
- appliquer les forces de fermeture
- réappliquer les contraintes acycliques

Un pseudocode plus détaillé est fourni en figure (5.19). Nous appellerons cette méthode la *résolution mixte directe*, par opposition à la résolution mixte itérative que nous présenterons au chapitre 5.5.2.

```
pour i de 0 à n_c − 1:
    e_c[i] = e[n_a + i]
    f[n_a + i] = 0
resolutionAcyclique(e, f)
e = D · f
pour i de 0 à n_c − 1:
    e_c[i] −= e[n_a + i]
ecrireMatriceFermetures(D_c)
resoudre D_c · f_c = e_c
pour i de 0 à n_c − 1
    f[n_a + i] = f_c[i]
e = D · f
resolutionAcyclique(e, f_1)
f += f_1
```

FIG. 5.19 - *Résolution mixte directe de* $\boldsymbol{D} \cdot \boldsymbol{f} = \boldsymbol{e}$

La complexité algorithmique de cette approche s'exprime en fonction du nombre de contraintes acycliques n_a et du nombre de contraintes de fermetures de chaînes n_c. Le nombre total de contraintes est $n = n_a + n_c$. Nous nous plaçons dans le cas où le graphe cinématique de la scène est un arbre. Les forêts peuvent se traiter arbre par arbre.

Remarquons tout d'abord que le nombre de solides est égal au nombre d'articulations acycliques si la racine du graphe est attachée au repère absolu, et à ce même nombre diminué de 1 si la racine de l'arbre est un objet en mouvement libre. Chaque articulation prise en compte comportant au minimum une contrainte, le nombre de solides est inférieur ou égal au nombre de contraintes acycliques. Le produit $\boldsymbol{D}.\boldsymbol{f}$ décrit dans le pseudocode de la figure s'effectue donc en temps $O(n)$. La résolution acyclique fonctionne en temps $O(n_a)$. La construction de la matrice $\boldsymbol{D}_c$ nécessite n_c résolutions acycliques, et se fait donc en temps $O(n_c n_a)$. La matrice $\boldsymbol{D}_c$ n'étant pas creuse dans le cas général, et elle est stockée dans un espace $O(n_c^2)$ et sa résolution s'effectue en temps $O(n_c^3)$. La matrice $\boldsymbol{D}_c$ pouvant être singulière, il faut utiliser une résolution robuste comme un pivot de Gauss perfectionné ou une décomposition SVD. La complexité totale de cette résolution est donc $O(n) + O(n_a n_c) + O(n_c^3)$, tandis que la complexité en espace est $O(n) + O(n_c^2)$.

On voit qu'il y a grand avantage à utiliser une telle résolution mixte dans le cas où le nombre de fermetures de chaînes est faible devant le nombre total de contraintes.

5.5.2 Résolution mixte itérative

L'idée d'une résolution mixte itérative est née du rêve de créer une méthode de résolution efficace pour toute structure cinématique. Avec peu de chaînes fermées, la résolution mixte directe est la plus efficace. Si l'on augmente le nombre de chaînes fermées, la composante cubique de sa complexité la rend de moins en moins intéressante par rapport à un gradient conjugué. Avec un grand nombre de chaînes fermées, le gradient conjugué est la seule méthode efficace. On pourrait être tenté d'utiliser l'une ou l'autre de ces méthodes suivant le type de structure à simuler. Malheureusement, des contacts peuvent apparaître ou disparaître au cours de l'animation, ce qui peut rendre totalement inefficace une méthode de résolution initialement bien choisie. La figure (5.20) montre un exemple évoquant un collier de perles posé sur une table. Le nombre de chaînes fermées, initialement grand, diminue jusqu'à zéro si l'on continue à tirer dans le sens de la flèche. Inversement, il peut réaugmenter si l'on repose le collier.

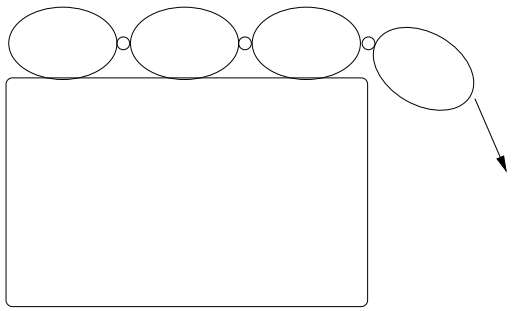


FIG. 5.20 - *Une structure cinématique variable.*

La méthode mixte itérative, en combinant résolution linéaire acyclique et gradient conjugué, s'adapte automatiquement à de tels changements de configurations. Nous avons vu au chapitre 5.3 que le gradient conjugué n'utilise la matrice dynamique qu'à travers son produit par un vecteur, et résoud un système de taille (n, n) en n itérations. L'idée fondamentale de la méthode mixte itérative est de résoudre le système $\boldsymbol{D}_c \cdot \boldsymbol{f}_c = \boldsymbol{e}_c - \boldsymbol{D}_{ca} \cdot \boldsymbol{D}_{aa}^{-1} \cdot \boldsymbol{e}_a$ par gradient conjugué. Pour ce faire, nous n'avons donc pas besoin d'écrire explicitement la matrice $\boldsymbol{D}_c$ mais uniquement de calculer des produits $\boldsymbol{x}_c = \boldsymbol{D}_c \cdot \boldsymbol{f}_c$. Ce produit s'effectue par la procédure $produitRestreint(\boldsymbol{f}_c, \boldsymbol{x}_c)$ selon le pseudocode fourni dans l' annexe B.

Nous disposons d'une procédure $gradientConjugue(\boldsymbol{b}, produit, \boldsymbol{f})$ où $\boldsymbol{b}$ est la contrainte et *produit* une procédure qui à un vecteur $\boldsymbol{f}$ associe un vecteur $\boldsymbol{x}$. Le vecteur $\boldsymbol{f}$ est le résultat de la résolution. Naturellement, nous utiliserons *produitRestreint* qui met en relation les forces et les déplacements aux fermetures de chaînes. Nous sommes maintenant prêts à effectuer la résolution. Nous commençons par résoudre les contraintes acycliques pour nous placer dans une configuration où seules les contraintes de fermetures restent à résoudre. Nous résolvons ces contraintes grace au gradient conjugué, puis appliquons les forces trouvées. Une dernière résolution acyclique mène à une configuration où toutes les contraintes sont satisfaites. Cette méthode est représentée sur le pseudocode de la figure (5.21).

L'analyse de complexité utilise les mêmes variables que pour la méthode mixte directe. Les résolutions acycliques s'effectuent en temps $O(n_a)$ et les produits $\boldsymbol{D} \cdot \boldsymbol{f}$ en temps $O(n)$.

$\boldsymbol{f} = \mathbf{0}$
$resolutionAcyclique(\boldsymbol{e}, \boldsymbol{f})$
$\boldsymbol{x} = \boldsymbol{D} \cdot \boldsymbol{f}$
pour i de 0 à n_c:
    $\boldsymbol{e}_c[i] = \boldsymbol{e}[n_a + i] - \boldsymbol{x}[n_a + i]$
$gradientConjugue(\boldsymbol{e}_c, produitRestreint, \boldsymbol{f}_c)$
$\boldsymbol{f}_1 = \mathbf{0}$
pour i de 0 à n_c:
    $\boldsymbol{f}_1[n_a + i] = \boldsymbol{f}_c[i]$
$\boldsymbol{e} = \boldsymbol{D} \cdot \boldsymbol{f}_1$
$resolutionAcyclique(\boldsymbol{e}, \boldsymbol{f}_1)$
$\boldsymbol{f} \mathrel{+}= \boldsymbol{f}_1$

FIG. 5.21 - *Résolution mixte itérative de* $\boldsymbol{D} \cdot \boldsymbol{f} = \boldsymbol{e}$

En conséquence, le produit $\boldsymbol{D}_c \cdot \boldsymbol{f}_c$ effectué par la procédure *produitResreint* est fait en temps $O(n)$. Le gradient conjugué convergeant en effectuant $O(n_c)$ produits, la complexité totale en temps de l'algorithme est $O(n) + O(n_c n)$, et la complexité en espace est $O(n)$. Par rapport à la résolution mixte directe, cette méthode possède donc plusieurs avantages:

- une meilleure complexité théorique en temps, puisque celle-ci tend vers $O(n^2)$ à mesure que la proportion de fermetures augmente;
- une meilleure complexité en espace car on n'utilise plus aucune matrice dense;
- la possibilité donnée par le gradient conjugué d'arrêter la résolution avant la fin; cela peut permettre de limiter le temps de calcul dans un contexte d'application en temps réel;
- la capacité donnée par le gradient conjugué avec résidu minimum de trouver le meilleur compromis entre des contraintes incompatibles.

5.6 Résultats

Les différents algorithmes présentés dans ce chapitre ont été implémentés et testés sur des scènes-jouets. Il est toujours délicat de présenter des temps de calcul, tant ceux-ci varient en fonction de la qualité de l'implémentation, de la compilation et de la machine utilisée. Les chiffres présentés ont été obtenus sur une SGI Indigo2 équipée d'un processeur R5000 cadencé à 150 Mhz. Aucun effort pour optimiser l'implémentation des algorithmes n'a été fait. Le code utilisé a été développé à des fins de recherche, pour évaluer l'efficacité relative des différents algorithmes et non pas pour obtenir des records en termes de performances pures. En conséquence, les chiffres absolus sont supérieurs à certains qu'on peut trouver dans la littérature. Leur principal intérêt est de donner des éléments de comparaisons entre algorithmes qui indiquent les pistes à suivre. Une implémentation optimisée des meilleurs est envisagée. Les temps de calcul ont été mesurés entre l'instant

où le système $\boldsymbol{j} \cdot \boldsymbol{M}^{-1} \cdot \boldsymbol{J}^t \boldsymbol{f} = \boldsymbol{e}$ est écrit et l'instant où la solution est écrite dans le vecteur $\boldsymbol{f}$.

5.6.1 Chaîne simple

Un premier test permet de se faire une idée des vitesses relatives des différentes résolutions. Une simple série de barres articulées par des liaisons rotules à leurs extrémités est accrochée à l'univers. L'ensemble subissant la pesanteur, colinéaire aux barres. La configuration est représentée sur la figure (5.22).

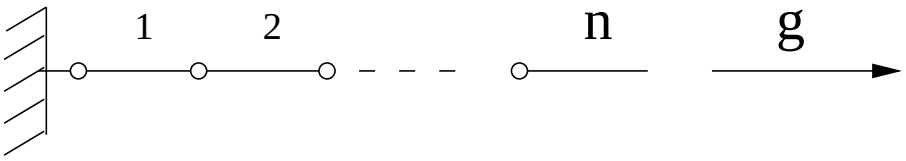


FIG. 5.22 - *Première configuration test.*

On observe la variation du temps de calcul en fonction du nombre n de barres. Les temps de calcul est exprimé en centièmes de secondes. Les algorithmes testés sont l'approche itérative locale, l'approche itérative locale à propagation immédiate, la résolution par pivot de Gauss et par décomposition de Cholesky de la matrice dense. Les résultats, présentés en figure (5.23), montrent un très net avantage de la décomposition de Cholesky. Les autres méthodes, beaucoup plus lentes, sont par contre susceptibles de traiter des matrices singulières. On voit que la différence est faible entre la méthode du pivot et la méthode locale à propagation immédiate. Le taux de convergence de cette dernière dépend par contre fortement du coefficient que l'on applique à la correction. Il est intéressant de constater que des coefficients supérieurs à un accélèrent grandement la convergence dans certains cas. Le coefficient utilisé ici est 1.6, déterminé expérimentalement. La résolution locale itérative simple ne peut pas être "accélérée"par un coefficient supérieur à un sous peine de diverger immédiatement. Notons qu'il est intéressant d'étudier ces méthodes itératives car elles sont de complexité spatiale linéaire, contrairement aux décompositions de matrices denses qui sont pourtant encore largement utilisées.

Les courbes de la figure (5.24) montrent une comparaison entre la plus rapide méthode cubique, Cholesky, et des méthodes de complexité moindre, le gradient conjugué et la résolution en temps linéaire présentée au chapitre 5.4. Le gradient conjugué prend l'avantage sur Cholesky à partir de 40 barres. Notons que le mécanisme utilisé étant essentiellement unidimensionnel, et la gravité dans le sens des barres, le gradient conjugué a effectivement traité 40 contraintes. Les liaisons entre les barres sont des rotules, avec trois contraintes. La décomposition de Cholesky a donc, pour le même nombre de barres, traité 120 contraintes, sans prendre en compte le fait que les deux tiers d'entre elles pouvaient sans inconvénient être ignorées.

5.6.2 Comparaison des méthodes locales itératives

Le cas étudié dans ce paragraphe donne des indications sur l'efficacité comparée des approches locales ou semi-locales. Dans les approches locales, les erreurs sont prises en

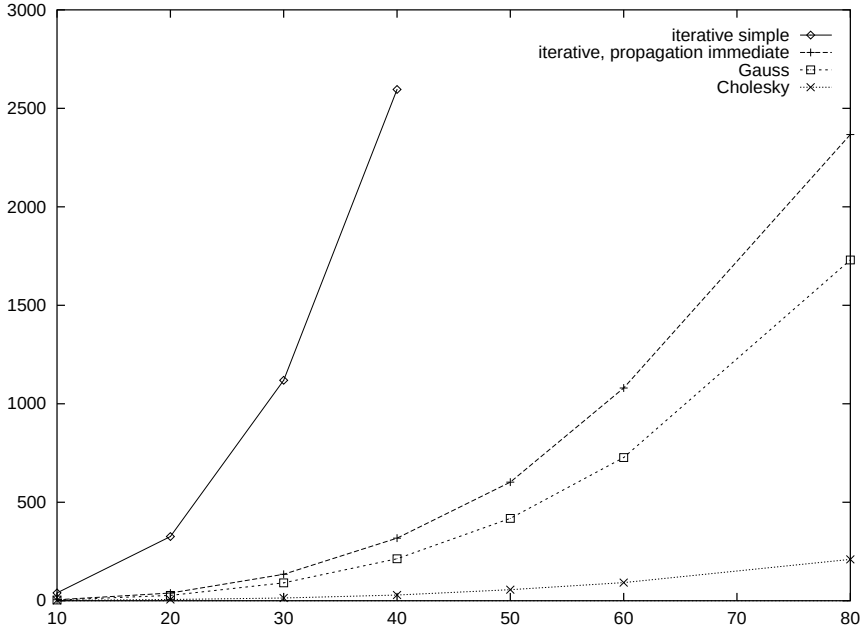


FIG. 5.23 - *Résultats de quatre approches de complexité cubique.*

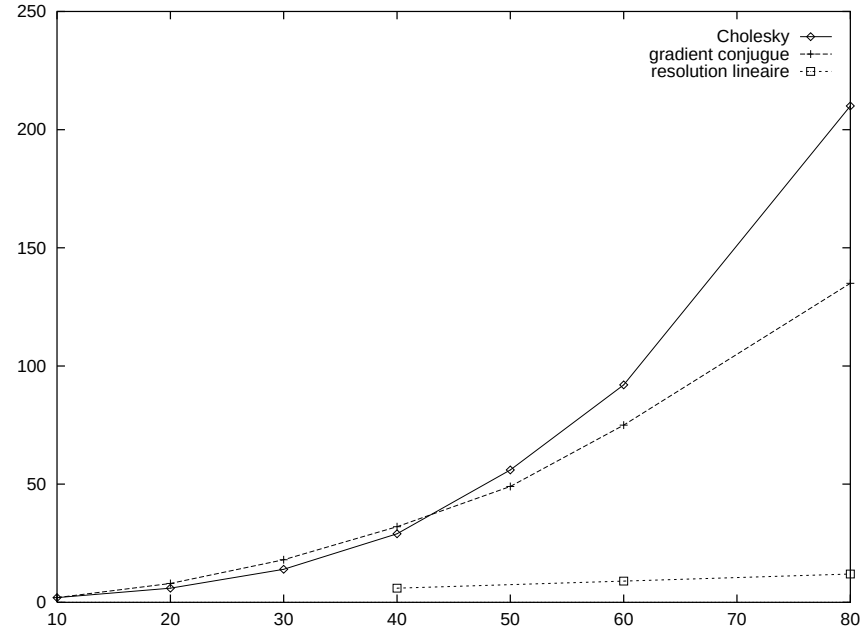


FIG. 5.24 - *Comparaison entre méthodes de différentes complexités théoriques*

compte contrainte par contrainte, alors qu'elles sont prises en compte articulation par articulation dans les méthodes semi-locales. La figure (5.25) montre l'exemple d'un mécanisme composé de barres articulées par des liaisons pivots. La gravité est imposée perpendiculairement aux barres, de sorte que l'ensemble reste immobile. L'erreur initiale d'accélération vaut $-g\boldsymbol{y}$ entre la barre 1 et son support fixe. Une force verticale de rétablissement tend à faire pivoter les barres autour de l'axe perpendiculaire à la feuille. Ce mouvement est interdit par les liaisons pivots. Inversement, le rétablissement d'une contrainte d'alignement des axes des pivots tend par rotation à faire monter ou descendre les points de fixation. Cette configuration génère de fortes interactions entre contraintes de positions et d'orientation au sein d'une même liaison.

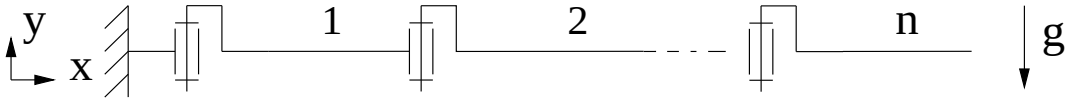


FIG. 5.25 - *Un mécanisme dont les contraintes interagissent fortement*

Les résultats présentés sur la figure (5.26) montrent encore une fois un net avantage des méthodes à propagation immédiate. Par contre, la supériorité des approches semi-locales reste assez faible, malgré la forte interaction entre les contraintes au sein de chaque liaison. Globalement, la progression des temps de calcul semble plus que cubique. L'amplification optimale de correction est limitée à 0.5 pour les méthodes sans propagation immédiate, et croît de 1.4 à 1.9 en fonction du nombre de barres pour les méthodes à propagation.

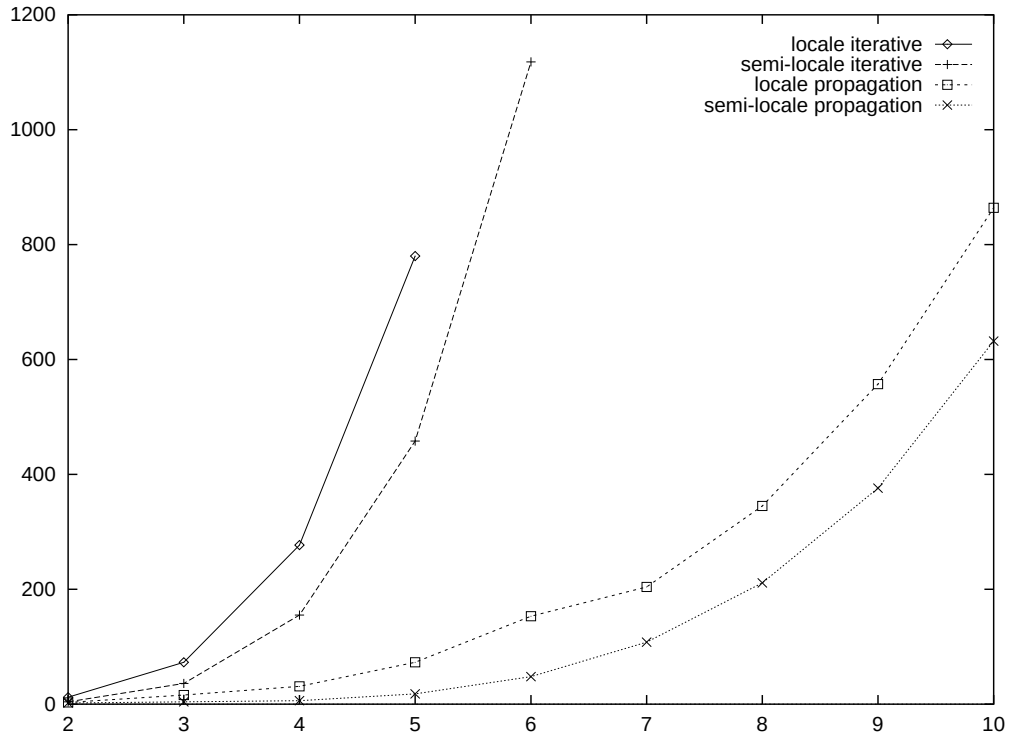


FIG. 5.26 - *Supériorité des méthodes à propagation immédiate parmi les méthodes locales*

5.6.3 Méthodes rapides

Les essais suivants ont pour but de comparer les méthodes rapides. Nous appelons ici méthodes rapides le gradient conjugué et les méthodes mixtes, car ce sont les seules meilleures en pratique à partir de quelques dizaines de contraintes. La complexité théorique en temps du gradient conjugué est quadratique, alors que la complexité des méthodes mixtes a une composante linéaire qui concerne la partie acyclique du graphe cinématique. La composante dûe aux fermetures de chaînes est cubique pour la méthode mixte directe et quadratique pour la méthode mixte itérative. La question que nous nous posons est de savoir dans quel cas il faudra préférer telle ou telle méthode. L'intuition nous dit que le gradient conjugué devrait être préférable pour des structures fortement cycliques, c'est à dire possédant un grand nombre de chaînes fermées. Inversement, on s'attend à ce que la méthode mixte directe prenne l'avantage pour les structures faiblement cycliques. On peut supposer que la méthode mixte itérative, qui n'a pas de composante cubique mais nécessite deux applications de forces à chaque itération, est la mieux adaptée dans les cas intermédiaires.

Le premier exemple porte sur une structure composée de deux séries de tiges articulées par liaisons rotules, et reliées entre elles par des barres transversales un peu à la manière d'une échelle, comme le montre la figure (5.27). Les barres sont de longueur, masse, et inertie unitaire. Le système est initialement au repos, et on applique à chaque barre du système une force de direction aléatoire et d'intensité $10000N$. À l'instant suivant, $0.04s$ après, le système se retrouve dans l'état représenté sur la figure (5.28). C'est en partant de cet état que les temps de calcul sont chronométrés, en réappliquant les mêmes forces. Ceci a son importance pour les méthodes itératives dont les nombres d'itérations sont variables. On évite ainsi de se placer dans le cas trop particulier des barres parfaitement alignées ou perpendiculaires.

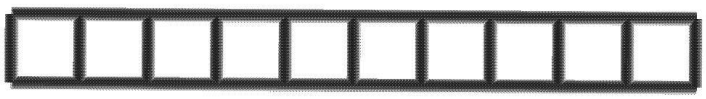


FIG. 5.27 - *Le réseau dans son état initial.*

FIG. 5.28 - *Le réseau après application des forces.*

Le problème est paramétré en fonction du nombre de mailles élémentaires du réseau. L'exemple de la figure (5.27) correspond à $n = 10$. Le réseau compte $3n + 1$ barres et $4n$ liaisons dont n de fermetures de chaînes. Chaque liaison comporte 3 contraintes. Le total

de contraintes est donc de $12n$ dont $3n$ contraintes de fermetures, soit une proportion de un quart. Les résultats pour différentes valeurs de n sont présentés en figure (5.29).

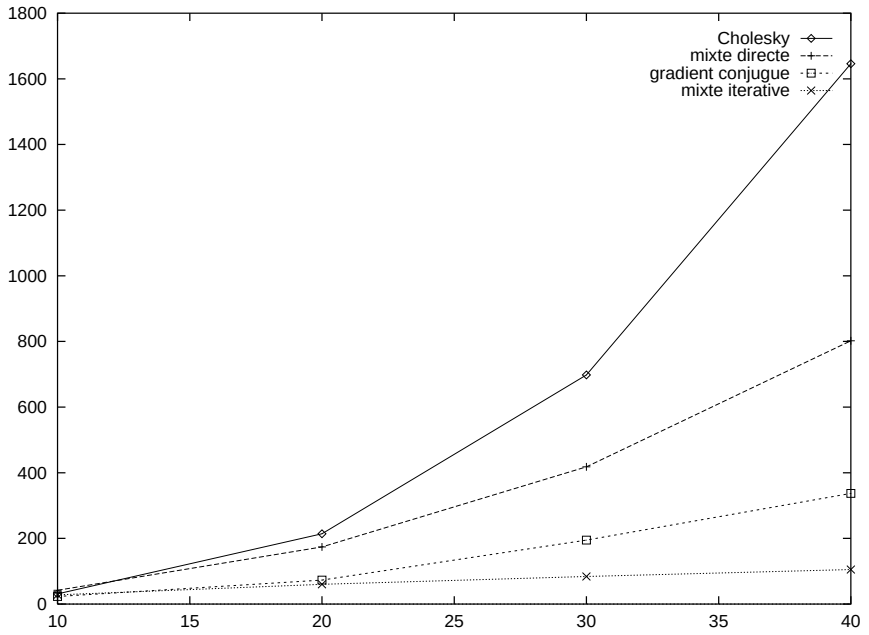


FIG. 5.29 - *Temps de calcul pour différentes méthodes en fonction de la taille du problème.*

Les résultats partagent les méthodes en deux catégories, cubiques ou non cubiques. La décomposition de Cholesky, de complexité entièrement cubique, devient réellement inefficace à partir de 20 mailles. La méthode itérative directe, pour laquelle seules les chaînes fermées induisent une composante cubique, est plus efficace. Cette composante cubique est pourtant un handicap certain puisque le gradient conjugué, quadratique, prend nettement l'avantage. La méthode mixte itérative, grâce à sa composante linéaire, reste la plus efficace sur cet exemple. Notons que les méthodes les plus rapides prennent l'avantage dès le début. Ceci indique que leur avantage en termes de complexité n'est pas uniquement théorique ou réservé aux très grands nombres de contraintes.

Les deux méthodes rapides du dernier exemple présentent en outre l'avantage d'êtres itératives, puisqu'elles reposent sur l'utilisation du gradient conjugué avec résidu minimum. La minimisation itérative de l'erreur peut s'arrêter dès qu'un seuil de précision voulue est atteint, contrairement aux méthodes basée sur la résolution de matrices denses. Nous avons choisi pour critère d'arrêt une valeur de diminution de l'erreur maximum. Nous appelerons *taux de réduction* ce critère, noté r et défini par:

$$r = \frac{\|\boldsymbol{b}\|_\infty}{\|\boldsymbol{D} \cdot \boldsymbol{f} - \boldsymbol{b}\|_\infty}$$

Les valeurs utilisées pour les résultats de la figure (5.29) correspondent à une réduction de $r = 10000$ de l'erreur maxi. Il est clair qu'un tel taux de réduction n'est pas toujours nécessaire, aussi avons-nous effectué des chronométrages pour différentes valeurs de réduction d'erreur. Les courbes de la figure (5.30) montrent les temps de calcul pour la méthodes du gradient conjugué avec différents taux de réduction. On observe qu'un taux de réduction de 100 s'obtient deux à trois fois plus vite qu'un taux de 10000.

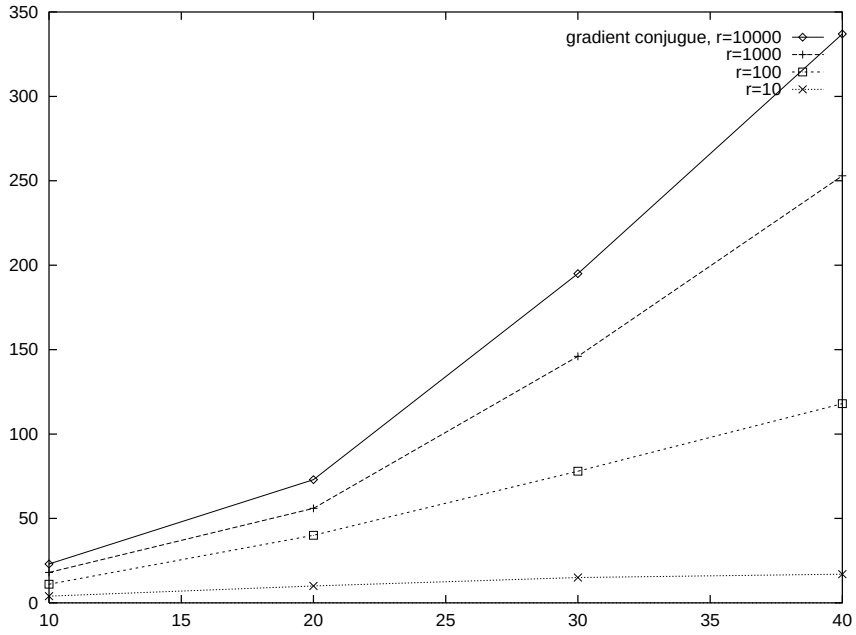


FIG. 5.30 - *Temps de calcul pour différents facteurs de réduction d'erreur, méthode du gradient conjugué.*

Les résultats de la figure (5.31) montrent les résulats obtenus par la méthode mixte itérative pour différents taux de réduction. La principale surprise est que, vus à l'échelle adéquate, les résultats montrent que le temps de calcul augmente moins vite que la taille du problème! Ce résultat est difficile à interpréter; il signifie que le nombre d'itérations décroît à mesure que le nombre de mailles (et donc de chaînes fermées) augmente. Nous avons probablement affaire ici à un cas particulier. L'irrégularité observable pour $n = 30$, $r = 100$ suggère que la méthode est peut-être instable.

Le test suivant a pour but d'évaluer l'influence que peut avoir la proportion de chaînes fermées par rapport au nombre total de contraintes. Nous reprenons donc la même expérience mais retirons quatre cinquièmes des barres transversales. Le paramètre n représente maintenant le nombre de barres longitudinales. Pour $n = 10$, nous avons donc une structure équivalente à celle de la figure (5.27), avec seulement une barre transversale au milieu et une à chaque extrémité. La proportion de chaînes fermées par rapport à l'ensemble des contraintes est maintenant voisine de 1/20. Les résultats correspondants sont présentés sur la figure (5.32). On voit que la méthode mixte directe a pris l'avantage sur le gradient conjugué, grace à la plus forte importance relative de sa composante linéaire. La méthode mixte itérative reste la plus rapide.

Les figures suivantes (5.33 et 5.34) montrent à nouveau l'évolution des temps de calculs des méthodes itératives pour différents taux de réduction. La méthode mixte itérative paraît là encore moins stable que le gradient conjugué.

Le dernier test effectué porte sur une proportion très importante de chaînes fermées. La structure présentée sur la figure 5.35) est un treillis carré à maille triangulaire. Les liaisons entre les barres sont des rotules, comportant chacune trois contraintes. Le treillis est paramétré en fonction du nombre n de barres le long d'un côté. L'ensemble compte $3n^2 + 2n$ barres et $5n^2 + 4n - 3$ liaisons, ce qui représente un total de $15n^2 + 12n - 9$ contraintes dont $6n^2 + 6n - 2$ contraintes de fermetures de chaînes. Le taux de fermetures

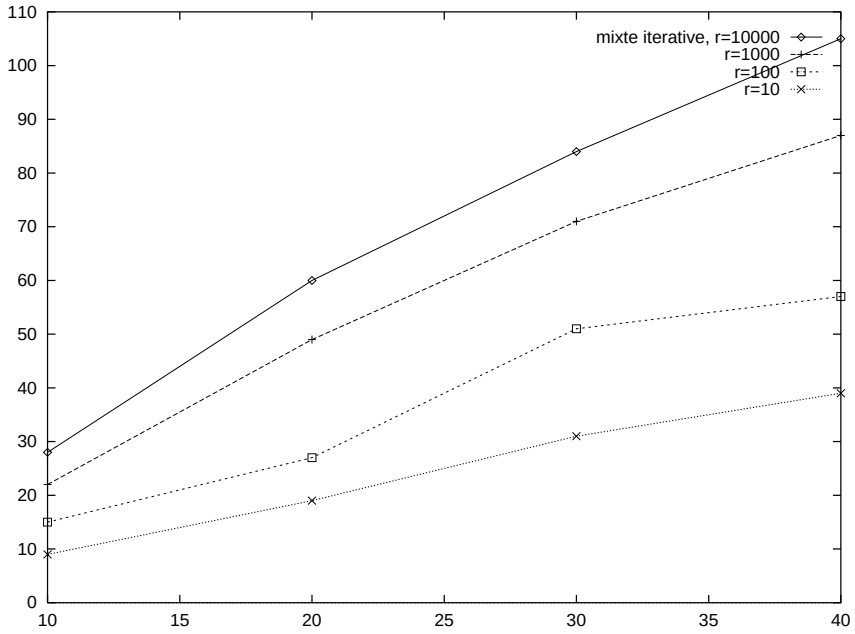


FIG. 5.31 - *Temps de calcul pour différents facteurs de réduction d'erreur, méthode mixte itérative.*

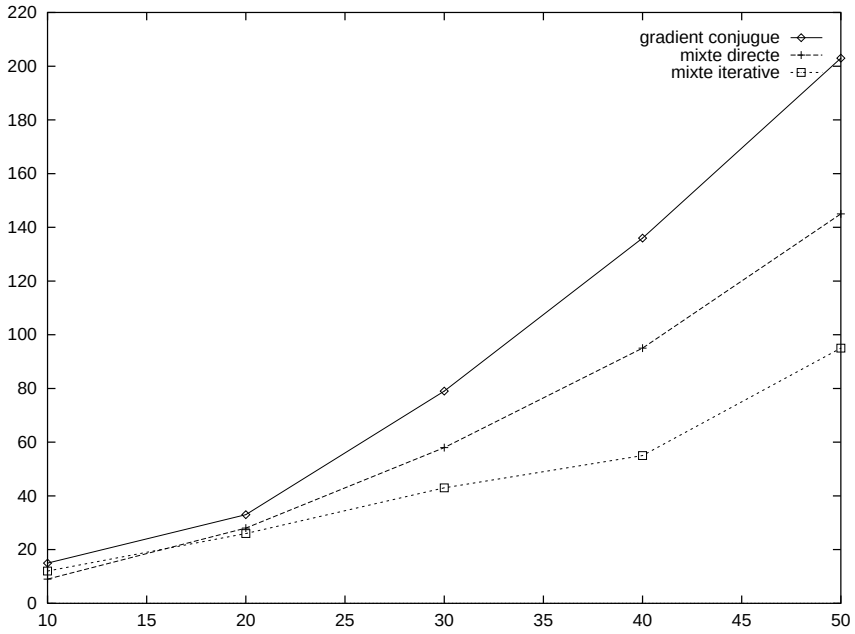


FIG. 5.32 - *Temps de calcul sur une structure comportant peu de chaînes fermées*

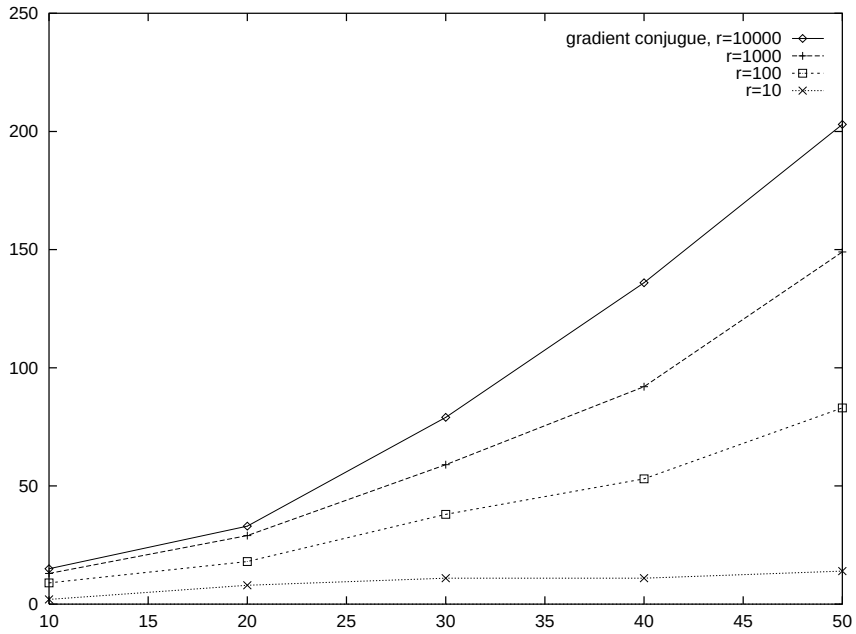


FIG. 5.33 - *Temps de calcul pour différents facteurs de réduction d'erreur, méthode du gradient conjugué.*

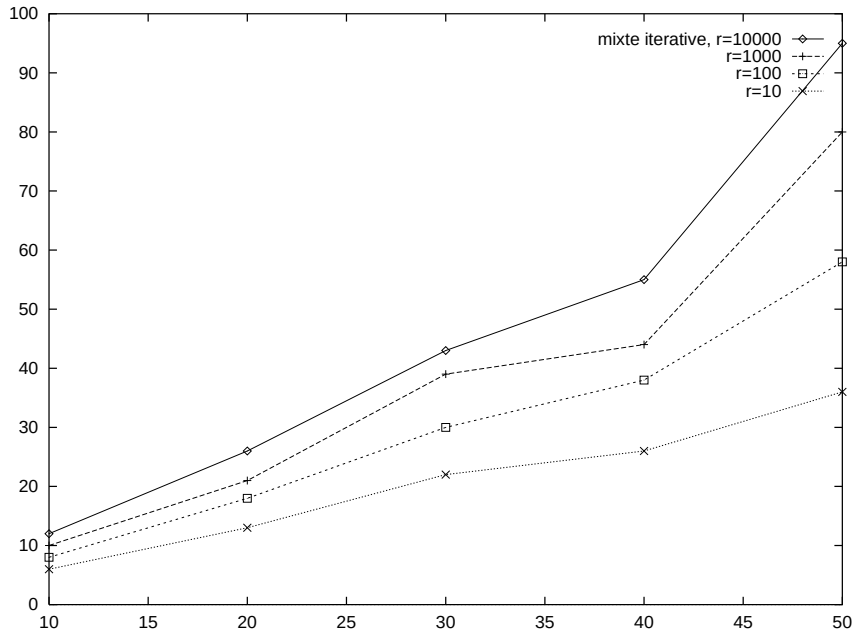


FIG. 5.34 - *Temps de calcul pour différents facteurs de réduction d'erreur, méthode mixte itérative.*

de chaînes est donc proche de 2/5.

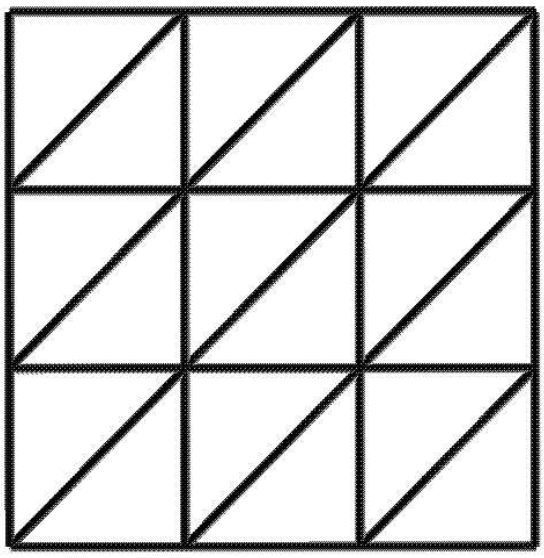


FIG. 5.35 - *Treillis à mailles triangulaires, $n = 3$.*

Les résultats, présentés sur la figure (5.36), montrent dans ce cas un avantage du gradient conjugué sur les deux méthodes mixtes. Ce n'est guère surprenant dans la mesure où le gradient conjugué est la méthode de référence pour les structures fortement bouclées. La méthode itérative directe subit l'influence de sa composante cubique. Le résultat le moins attendu est que la méthode mixte itérative ne produit pas de meilleurs résultats et semble suivre la même progression que la méthode mixte directe. Pour analyser ce résultat, il faut se souvenir que la méthode consiste en un gradient conjugué restreint aux fermetures de chaînes. Le conditionnement de ce système d'équations n'étant pas forcément bon, il semble que nous soyons là dans un cas où un mauvais conditionnement de la matrice dynamique dû à un nombre élevé de contraintes provoquent des erreurs numériques.

Les résultats de la figure (5.37) montrent en revanche qu'une réduction d'erreur d'un facteur 100 s'obtiennent avec une bien plus grande rapidité. Ceci semble confirmer l'idée que les erreurs numériques freinent la convergence pour les petites valeurs d'erreur.

5.7 Dynamique inverse

Nous nous intéresserons ici à imposer un mouvement dans une articulation. Les mouvements des solides étant déterminés par les forces qu'ils subissent, la question est de calculer les forces à mettre en oeuvre pour obtenir un mouvement désiré. Nous avons surtout parlé jusqu'à présent de liaisons "passives", dans le sens où elles ne font que réagir aux mouvements des solides pour interdire les mouvements relatifs dans les directions des blocages. Les liaisons "actives" imposent, en plus, un mouvement relatif dans les directions des mobilités. Si nous prenons l'exemple du pivot, une liaison passive laisse libre la rotation le long de l'axe alors que la liaison pilotée impose une certaine vitesse de rotation. Ceci se traduit par les différents ensembles de contraintes à résoudre comme illustré sur

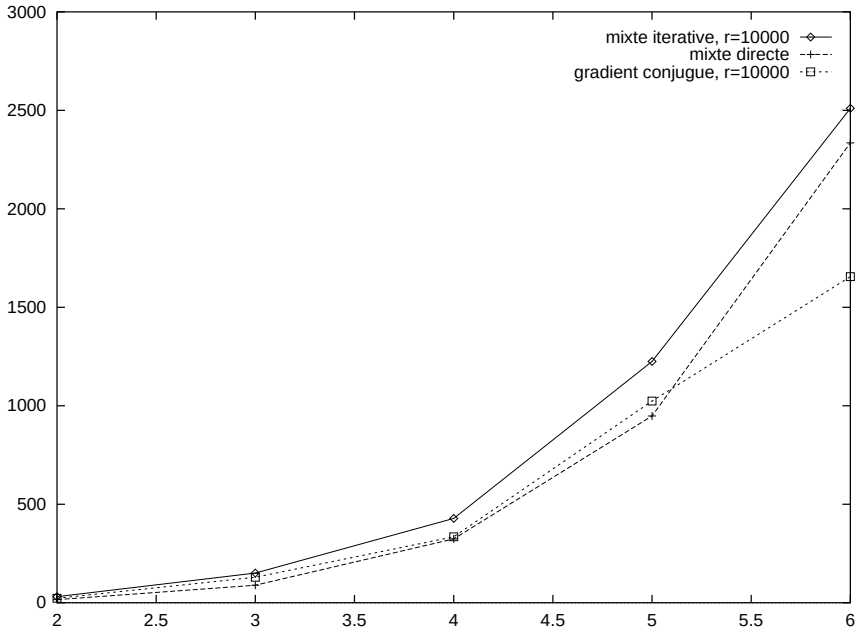


FIG. 5.36 - *Temps de calcul pour le treillis.*

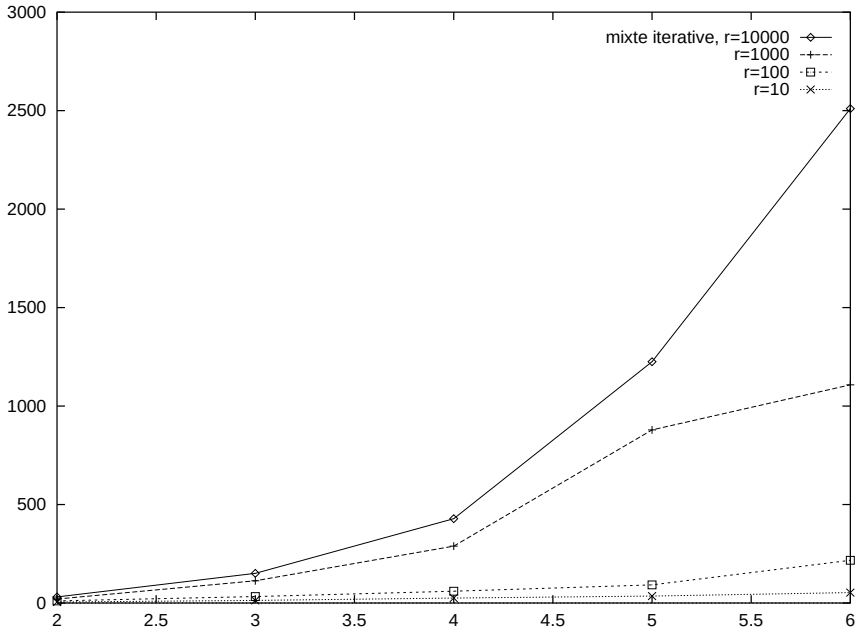


FIG. 5.37 - *Temps de calcul à différentes précisions pour le treillis par la méthode mixte itérative.*

la figure (5.38), qui représente une barre en pivot par rapport au repère absolu et soumise à la gravité $\boldsymbol{g}$.

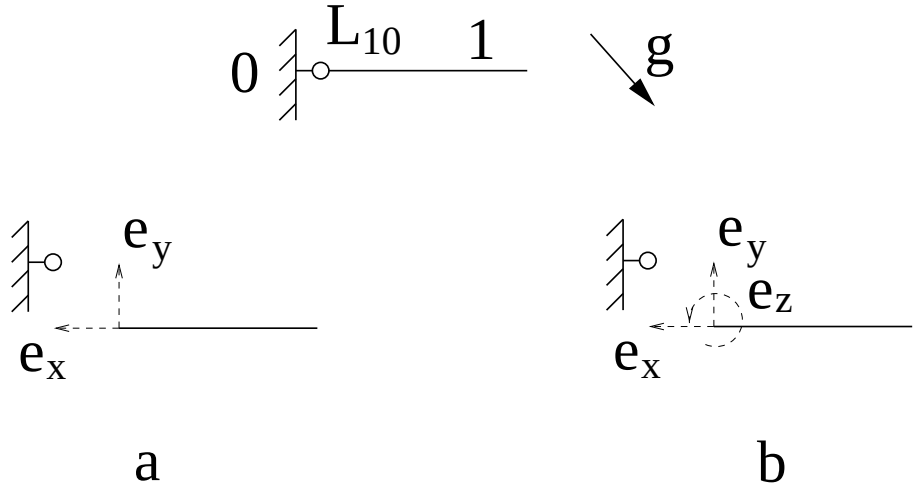


FIG. 5.38 - *Les contraintes d'une liaison passive (a) et d'une liaison active (b).*

Les valeurs des mouvements relatifs désirés s'ajoutent aux mouvements relatifs mesurés, conformément aux équations (5.1) et (5.2). L'erreur mesurée sur l'axe $\boldsymbol{z}$ est une erreur en rotation et la force de contrainte correspondante est donc un moment. Les formules (5.15) illustrent les conséquences de cette nouvelle contrainte en comparant les équations de la liaison passive et celles de la liaison active.

$$\begin{array}{c|c} \boldsymbol{D}_{(2,2)} \cdot \begin{pmatrix} f_x \\ f_y \end{pmatrix} = \begin{pmatrix} e_x \\ e_y \end{pmatrix} & \boldsymbol{D}_{(3,3)} \cdot \begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix} = \begin{pmatrix} e_x \\ e_y \\ e_z \end{pmatrix} \\ \boldsymbol{f}_1^{O_1} = \boldsymbol{\Delta}_{O_1 L_{10}} \cdot (f_x \boldsymbol{a}_x + f_y \boldsymbol{a}_y) & \boldsymbol{f}_1^{O_1} = \boldsymbol{\Delta}_{O_1 L_{10}} \cdot (f_x \boldsymbol{a}_x + f_y \boldsymbol{a}_y + f_z \boldsymbol{a}_z) \end{array} \quad (5.15)$$

Les indices de la matrice $\boldsymbol{D}$ indiquent sa taille. Dans le cas de la liaison active, nous avons imposé un mouvement relatif $-e_z$ dans la direction de rotation $\boldsymbol{z}$, et calculé une force selon cette direction. Dans le cas de la liaison passive, c'est comme si nous avions imposé une force nulle selon $\boldsymbol{z}$, et il en résulte un mouvement. La force imposée peut ne pas être nulle, si par exemple nous voulons modéliser la présence d'un ressort de torsion. On retrouve le résultat bien connu[IC87], qu'à chaque degré de liberté d'une structure on peut imposer soit le mouvement soit la force.

L'intérêt de la formulation dynamique par multiplicateurs de Lagrange (eq. 5.8) est qu'il n'y a absolument rien à modifier dans l'algorithme pour prendre en compte de telles contraintes de dynamique inverse. Seul le nombre de contraintes cinématiques change. Il suffit de créer de nouvelles classes d'articulations comportant des blocages dont la contrainte de mouvement est non nulle. Ce travail a été réalisé sur les liaisons pivot par Gilles Debunne pendant son stage de DEA et a permis l'analyse dynamique de mouvements de marche bipède précalculés cinématiquement. Ce travail a débouché sur une présentation de résultats au Workshop Eurographics'97 sur l'Animation et la Simulation[FDCGM97].

5.8 Conclusion

L'implémentation et les tests de différents algorithmes de résolution de systèmes dynamiques ont montré de considérables différences d'efficacité. Les méthodes cubiques deviennent très inefficaces à partir de quelques dizaines de contraintes. Parmi celles-ci, la décomposition de Cholesky de la matrice dynamique est la plus efficace mais ne fonctionne pas pour les matrices singulières, obligeant à utiliser une méthode de pivot, considérablement plus lente. Les méthodes de résolution locale à propagation immédiate des mouvements peuvent alors s'avérer compétitives, bien que délicates à régler.

Les formulations en coordonnées absolues permettant d'exprimer la matrice dynamique comme le produit de trois matrices creuses, l'algorithme du gradient conjugué est une alternative intéressante. Il permet de résoudre les systèmes en temps quadratique en fonction du nombre de contraintes. Il ne manipule que des vecteurs et nécessite donc un espace mémoire proportionnel au nombre de solides et de contraintes. Il améliore itérativement une solution, ce qui laisse l'utilisateur libre de définir un niveau de précision requis ou bien un temps de calcul à ne pas dépasser. Enfin, il traite les systèmes sous-contraints et trouve un compromis pour les systèmes surcontraints. Ses inconvénients sont une certaine sensibilité aux matrices mal conditionnées, et sa complexité systématiquement quadratique ce qui est très inefficace dans certains cas.

Les méthodes mixtes reposent sur une résolution en temps linéaire des systèmes ne comportant aucune chaîne fermée, et une résolution additionnelle des fermetures de chaînes. Elles surpassent donc largement le gradient conjugué pour les systèmes comportant peu de chaînes fermées. La méthode mixte directe consiste à écrire et résoudre un système d'équations concernant uniquement les chaînes fermées. Cette résolution, de complexité cubique, est un lourd handicap quand les chaînes fermées sont nombreuses. La méthode que nous proposons cumule les avantages de plusieurs approches. Elle utilise une résolution en temps linéaire ce qui la rend extrêmement efficace sur les structures faiblement bouclées. La résolution des chaînes fermées par une adaptation du gradient conjugué lui confère en plus les avantages de ce dernier. Il nécessite un espace mémoire strictement proportionnel au nombre de solides et de liaisons quelle que soit la structure cinématique du problème. Il traite les problèmes sous-contraints et sur-contraints. Sa capacité à prendre en compte des contraintes de dynamique inverse a permis de l'utiliser pour une application concrète d'analyse dynamique de la marche bipède. Sa complexité adaptative, variant de linéaire à quadratique selon la structure du problème, le rend efficace pour toutes structures. Il procède par réduction itérative de l'erreur de résolution ce qui permet de définir au choix le degré de précision ou le temps de calcul. Cette caractéristique pourrait en faire un outil très utile pour les environnements de réalité virtuelle où les contraintes de temps réel ont une importance

Chapitre 6

Vers une approche temps réel

Ce chapitre présente des approches du problème de la simulation permettant de prendre en compte des contraintes de temps réel. On désigne par cette expression la nécessité de pouvoir limiter le temps de calcul à une certaine valeur permettant d'animer une scène à une vitesse jugée réaliste par un spectateur. Cette question est particulièrement importante pour des simulateurs interactifs par lesquels on veut calculer la réponse d'un système à des commandes d'un utilisateur. On trouve parmi les applications des outils d'apprentissage en tout genre comme pour la télémanipulation ou la simulation de gestes techniques, l'interaction à distance...

Nous ne traiterons pas ici des architectures logicielles aptes à gérer le temps comme une ressource et allouer des durées maximales de calculs à différents sous-programmes. Nous nous intéresserons plutôt à mettre en oeuvre des algorithmes capables faire au mieux en un temps donné. Pour cela, on cherche à procéder par améliorations progressives d'une solution imparfaite. Il est ainsi possible d'arrêter le raffinement en cours et de fournir une réponse approximative.

La résolution mixte itérative des équations dynamiques présentée au chapitre précédent présente cette caractéristique. On pourrait donc l'utiliser dans une application à temps limité. Mais la résolution des équations dynamiques n'est pas le seul problème à résoudre pour effectuer une simulation. Les vitesses et accélérations doivent être intégrées pour faire évoluer les positions des objets au cours du temps. Les inévitables erreurs d'intégration numériques induisent des erreurs de position qu'on doit être capable de corriger. Les collisions entre objets créent de nouvelles contraintes cinématiques qu'il faut prendre en compte. La première partie de ce chapitre est consacrée à ces problèmes, et propose une approche qui évite au maximum de réduire les pas de temps, tout en effectuant une résolution globale et itérative des contraintes.

Un autre domaine de recherche est le calcul des forces de contacts entre solides, particulièrement quand ils suivent les lois du frottement de Coulomb. Nous proposons dans la deuxième partie de ce chapitre une approche originale qui permet là encore et contrairement aux approches précédentes, d'effectuer le raffinement progressif d'une solution globale.

6.1 Algorithme d'animation

6.1.1 Intégration numérique

Notre algorithme d'animation repose sur une intégration selon un schéma du type Euler modifié avec impulsions, et une prise en compte différée des collisions. L'ensemble est intégré dans une boucle globale dont chaque exécution fait avancer le temps d'un pas. Le principe de base du schéma d'Euler modifié est le suivant:

- calculer l'accélération $\boldsymbol{a}$ à l'instant t
- mettre à jour les vitesses $\boldsymbol{v}(t+dt) = \boldsymbol{v}(t) + \boldsymbol{a}.dt$
- mettre à jour les positions $\boldsymbol{x}(t+dt) = \boldsymbol{x}(t) + \boldsymbol{v}(t).dt + \frac{1}{2}\boldsymbol{a}.dt^2$

Nous animons un système articulé, aussi désirons-nous assurer la compatibilité des vitesses avant l'intégration. Cela correspond à résoudre le système $\boldsymbol{D}\cdot\boldsymbol{f} = \boldsymbol{e}_v$ où $\boldsymbol{e}_v$ est la contrainte nécessaire pour rendre les vitesses compatibles et $\boldsymbol{f}$ l'ensemble des impulsions nécessaires. Pour obtenir des accélérations compatibles avec les liaisons, nous devons d'autre part résoudre le système $\boldsymbol{D}\cdot\boldsymbol{f} = \boldsymbol{e}_a$ où $\boldsymbol{e}_a$ est la contrainte d'accélération à satisfaire et $\boldsymbol{f}$ l'ensemble des forces nécessaires. Nous fusionnons cette résolution en $\boldsymbol{D}\cdot\boldsymbol{f} = \boldsymbol{e}_v + \boldsymbol{e}_a.dt$ ce qui permet de résoudre les deux problèmes d'un coup en obtenant les vitesses $\boldsymbol{v}(t+dt)$ compatibles avec les liaisons et prenant en compte les influences extérieures. L'accélération n'étant pas explicitement calculée, on détermine les nouvelles positions en appliquant simplement $\boldsymbol{x}(t+dt) = x(t) + \frac{1}{2}(\boldsymbol{v}(t) + \boldsymbol{v}(t+dt))dt$.

6.1.2 Compatibilité des vitesses

Assurer la compatibilité des vitesses consiste à modifier les vitesses des solides pour que les contraintes de vitesses soient respectées. Les équations de vitesses sont communément utilisées pour traiter les collisions entre solides. À l'instant du choc, on pose comme contrainte que la vitesse relative au point de contact projetée sur la normale doit être nulle. Ceci a pour effet d'annuler la vitesse d'interpénétration des solides.

Comme dit dans le paragraphe précédent, nous utilisons aussi les équations de compatibilité des vitesses (eq. 3.10) pour forcer le système à retrouver un état cohérent. En effet, l'intégration numérique des accélérations génère forcément une erreur, d'autant plus grande que le pas de temps est important. Cette erreur se traduit par des vitesses relatives non conformes aux directions des mobilités. Cette erreur a pour effet d'amplifier les erreurs de position, dont nous parlerons dans le paragraphe suivant. Nous l'annulons par un traitement analogue au traitement des collisions.

Les efforts calculés sont d'un ordre supérieur aux forces et moments habituels puisqu'ils concernent des vitesses et non pas des accélérations. Ils s'expriment en $kg.m.s^{-1}$ ou en $kg.m^2.s^{-1}$ suivant qu'ils s'apparentent à des forces ou à des moments.

6.1.3 Compatibilité des positions

Les erreurs d'intégration numérique provoquent une dérive des positions qui peut devenir assez vite visible. Le contrôle des positions sert à annuler ce phénomène en garantissant

l'intégrité géométrique des liaisons au cours du temps. La figure (6.1) résume les variations de positions entre l'instant t et l'instant $t + dt$.

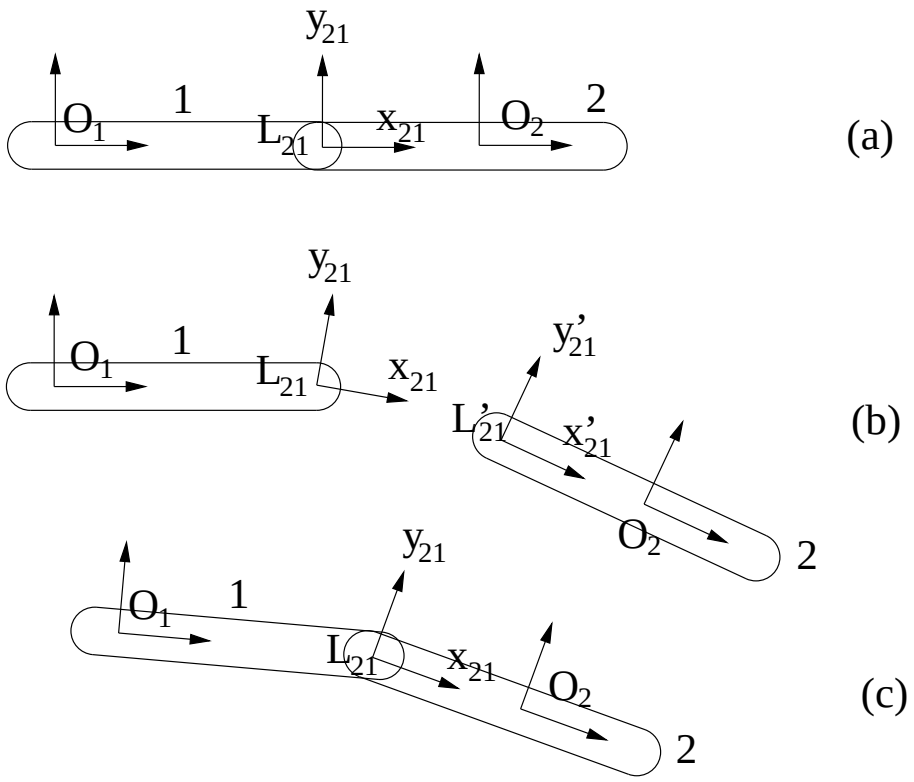


FIG. 6.1 - *(a): configuration à l'instant t. (b): configuration après intégration des mouvements (c): configuration à l'instant $t + dt$*

À l'instant t représenté par la configuration (a), deux solides reliés par une liaison pivot coïncident au point L_{21}. La variation de position des solides et des différentes matrices de passage est calculée et appliquée, ce qui mène à la configuration (b). Les erreurs d'intégration ont provoqué une rupture de la liaison, qui se traduit par un décalage des centres et des orientations du repère intermédiaire de liaison défini dans chaque solide. Mathématiquement, ce décalage se traduit par $\boldsymbol{P}_{L1} \cdot \boldsymbol{P}_{10} \neq \boldsymbol{P}_{L2} \cdot \boldsymbol{P}_{20}$. Les matrices $\boldsymbol{P}_{L1}$ et $\boldsymbol{P}_{L2}$ définissent le repère intermédiaire de liaison dans chaque repère, et les matrices $\boldsymbol{P}_{10}$ et $\boldsymbol{P}_{20}$ sont les matrices de positions des solides dans le repère absolu.

En nous inspirant de [GG94], nous pouvons rétablir la cohérence en quatre étapes:

1. calculer le décalage $\boldsymbol{P} = \boldsymbol{P}_{L2} \cdot \boldsymbol{P}_{20} \cdot (\boldsymbol{P}_{L1} \cdot \boldsymbol{P}_{10})^{-1}$

2. décomposer $\boldsymbol{P}$ en une translation et une rotation relative, les projeter dans le repère absolu et les rassembler dans un vecteur $\boldsymbol{p}$ à six dimensions

3. calculer et appliquer les déplacements nécessaires au rétablissement des contraintes en résolvant $\boldsymbol{D} \cdot \boldsymbol{f} = \boldsymbol{e}$ avec $e_k = -\boldsymbol{p}^t \cdot \boldsymbol{a}_k$ où les $\boldsymbol{a}_k$ sont les directions des contraintes de la liaison

4. remettre en correspondance les matrices du repère intermédaire de liaison en en calculant la nouvelle matrice $\boldsymbol{P}_{L1} = \boldsymbol{P}_{L2} \cdot \boldsymbol{P}_{20} \cdot \boldsymbol{P}_{10}^{-1}$

La première étape consiste à exprimer le décalage entre les deux versions du repère intermédiaire. L'étape 2 est une mise en forme de ce décalage afin de faciliter la mise en équation dans l'étape 3. L'étape 3 consiste à replacer les solides en leur appliquant des déplacements compatibles avec le principe de réciprocité des forces. On résoud des équations sur des déplacements linéarisés. Les efforts ainsi calculés sont d'un ordre supérieur aux impulsions, et s'expriment en $kg.m$ ou $kg.m^2$ selon qu'ils s'apparentent à des forces ou à des moments. L' approximation des déplacements linéarisés rend éventuellement nécessaire de recommencer à l'étape 1, jusqu'à obtention d'un vecteur d'erreur $\boldsymbol{e}$ quasi nul. L'étape 4 rétablit la correspondance totale des deux descriptions du repère intermédiaire de liaison. En effet, à l'issue de l'étape 3, il peut subsister un décalage selon les directions non contraintes de la liaison. À la différence de [GG94], nous nous gardons de réactualiser les vitesses en fonction du rétablissement géométrique afin de ne pas induire d'énergie dans le système.

Certains auteurs utilisent des méthodes apparemment moins "brutales" pour maintenir les contraintes géométriques. Isaacs et Cohen[IC88] préconisent l'usage d'un PD-contrôleur qui module la valeur des contraintes cinématiques, et stabilise la liaison en agissant à la manière d'un ressort et amortisseur. Le procédé de rétablissement des contraintes géométriques présenté ici offre certains avantages par rapport à cette approche classique. En premier lieu, les PD-contrôleurs sont paramétrés par des coefficients définis par l'utilisateur, et la détermination des bons coefficients peut s'avérer délicate. De trop faibles coefficients laissent trop de libertés aux contraintes, et de trop forts coefficients peuvent générer de fortes instabilités. Dans tous les cas, les forces de liaisons induisent ou retranchent de l'énergie au système car elles s'appliquent à des points en mouvement relatif. Notre approche ne nécessite pas de réglages de paramètres et ne modifie pas directement l'énergie du système car elle laisse les vitesses inchangées. Notons tout de même que le fait de "remonter" un solide dans un champ de pesanteur lui rajoute de l'énergie potentielle, notre méthode n'est donc pas non plus parfaite du point de vue énergétique. C'est d'ailleurs ce type d'erreur qui fixe la limite des pas de temps acceptables avec cette méthode. Elle permet par contre d'utiliser des pas de temps relativement grands tout en conservant l'intégrité des liaisons. Pour finir, elle permet l'auto-assemblage des solides, tout comme les approches à base de PD-contrôleurs.

6.1.4 Collisions

Les collisions doivent, en toute rigueur, être prises en compte à l'instant précis où elles se produisent. Ceci nécessite, quand une collision est détectée, de remonter dans le temps jusqu'à l'instant où elle s'est produite, traiter la collision en résolvant la contrainte de vitesses correspondante, et reprendre l'intégration du temps. La recherche de l'instant de collision $t + \alpha dt$ peut se faire par recherche dichotomique ou par une procédure *ad hoc*[Bar90]. Ce procédé est illustré sur la figure (6.2) représentant la collision de deux balles. La double flèche en (c) représente l'impulsion nécessaire au traitement de la collision.

Ceci est la manière courante de procéder[MW88, Hah88] mais pose des problèmes quand de multiples collisions se produisent au cours d'un même pas de temps. Il faut alors effectuer le processus précédemment décrit pour chacune des collisions, ce qui peut mener à une discrétisation du temps prohibitive en termes de temps de calcul. Récem-

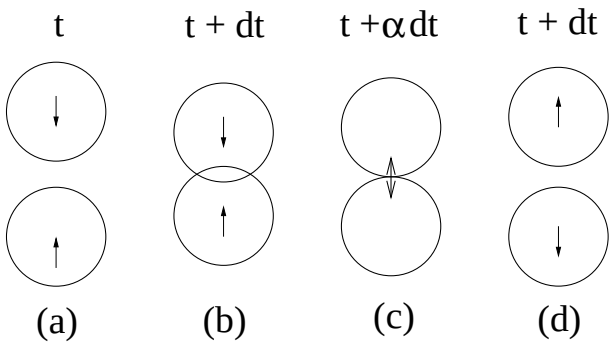


FIG. 6.2 - *Traitement classique des collisions. (a): instant t. (b): instant t+dt, détection de collision (c): retour à l'instant de collision et traitement des contraintes (d): nouvelle configuration à $t + dt$*

ment a été proposée une approche des collisions uniquement basée sur le traitement des positions[Mil96]. L'auteur propose une résolution purement géométrique consistant à trouver globalement à la fin de chaque pas de temps une configuration qui satisfait toutes les contraintes de non-pénétration. Une résolution globale suffit à traiter simultanément toutes les collisions intervenues au cours du pas de temps. L'approche géométrique utilisée, très originale, est extrêmement rapide mais limitée à des objets convexes en translation. Elle est même tellement géométrique qu'il en devient difficile d'y incorporer des notions physiques comme la vitesse ou l'inertie. Nous avons repris l'idée de la résolution globale différée des collisions, mais en les traitant de manière plus physique, contournant par là les limitations inhérentes de l'approche purement géométrique. Nous utilisons pour cela, tout simplement, la méthode de contrôle des positions présentée en 6.1.3. Elle nécessite pour cela d'être étendue au traitement d'inégalités, puisque les contraintes de non-pénétration ne s'activent qu'en cas de pénétration et non de décollement. Nous détaillerons ce point ultérieurement. La figure (6.3) illustre la méthode. Après avoir détecté la collision en (*b*), on ne remonte pas le temps mais on modélise le contact. En (c), le contrôle des positions est utilisé pour rétablir la non-pénétration, ce qui implique les efforts de déplacements symbolisées par la double flèche à pointe pleine. En (d), le contrôle des vitesses traite la contrainte de non-pénétration et applique le rebond. Le rebond se traite de manière classique[MW88, Hah88] en réappliquant au point de contact une partie des efforts qui avaient été nécessaires pour annuler la vitesse de pénétration.

Le rebond est donc traité au début du pas de temps suivant. L'inconvénient de cette méthode est d'induire un retard qui se traduit par le fait que toutes les collisions intervenues au cours du pas de temps apparaissent comme des contacts à la fin de ce pas de temps, même si les objets devraient s'être déjà éloignés. En pratique, pour des pas de temps de taille "normale" (0.04 s), cela ne génère pas d'artefacts visibles. L'avantage de cette approche est qu'elle n'oblige pas à de multiples retours en arrière qui ralentissent considérablement le processus de simulation.

6.1.5 Boucle d'animation

La boucle d'animation incorpore les éléments précedemment décrits. Elle consiste en une boucle infinie simulant le déroulement du temps. Le pseudocode de la figure (6.4) en

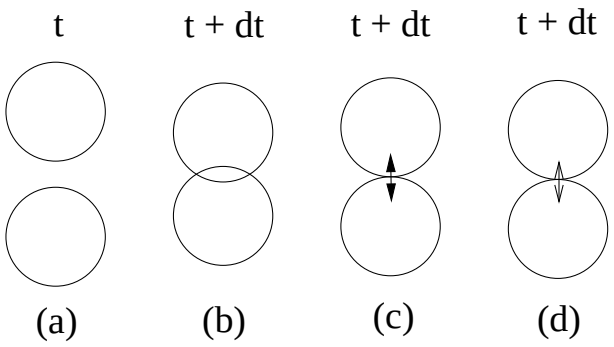


FIG. 6.3 - *Prise en compte différée des collisions. (a): instant t. (b): instant t+dt, détection de collision (c): rétablissement de la contrainte géométrique (d): traitement des vitesses.*

résume les principales étapes. Elle commence par une résolution en vitesse incluant les influences extérieures et se termine par une résolution en position permettant d'afficher les solides dans des positions respectant les contraintes géométriques. Les contraintes géométriques étant traitées par approximation en déplacements linéarisés, plusieurs passes peuvent être nécessaires. Cela est surtout vrai pour l'auto-assemblage de la scène au début de l'animation. En général par la suite, une passe suffit à chaque pas de temps dans les simulations classiques.

sommer les forces extérieures
incrémenter les vitesses
calculer la contrainte $\boldsymbol{e}_v + \boldsymbol{e}_a.dt$
résoudre la contrainte
réactualiser les vitesses
réactualiser les positions et les matrices $\boldsymbol{J}$ et $\boldsymbol{M}$
calculer la contrainte de position $\boldsymbol{e}_x$
tant que $\|\boldsymbol{e}_x\| > \epsilon$
    résoudre la contrainte
    réactualiser les positions et les matrices $\boldsymbol{J}$ et $\boldsymbol{M}$
    calculer la contrainte de position $\boldsymbol{e}_x$
$t \mathrel{+}= dt$
afficher

FIG. 6.4 - *Corps de la boucle d'animation*

6.2 Résolution itérative des forces de contact

Le calcul des forces de contact avec frottements de Coulomb est un des problèmes les plus difficiles de la simulation de solides par modèles physiques. Cette difficulté provient du fait que les contraintes aux contacts se traduisent par des inéquations, dont certaines sont non linéaires. Ce problème a été prouvé NP-dur par Baraff[Bar91].

6.2.1 Contraintes de contact

On modélise un contact ponctuel par une liaison entre deux solides 1 et 2 comme illustré sur la figure (6.5). Les contacts linéiques ou surfaciques peuvent être modélisés en utilisant plusieurs contacts ponctuels. La liaison est centrée en L_{21} et sa direction privilégiée est $\boldsymbol{x}_{21}$, orienté selon la normale commune aux surfaces. Le cône de friction est centré sur l'axe $\boldsymbol{x}$ et son demi-angle a pour tangente le coefficient de frottement, sur lequel nous aurons l'occasion de revenir.

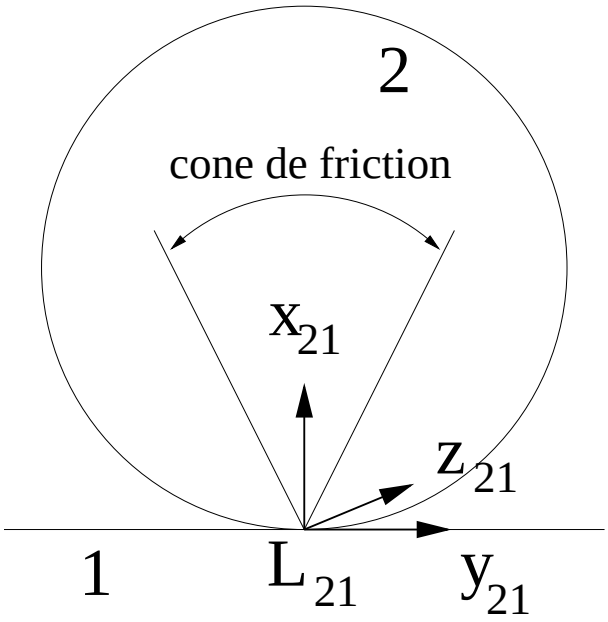


FIG. 6.5 - *Géométrie du contact entre les solides* 1 *et* 2.

Pour alléger les notations, nous représenterons l'accélération relative dans la direction de la normale au contact par a_x, et les accélérations relatives tangentielles par a_y et a_z. Similairement, nous noterons f_x la force normale, f_y et f_z les forces tangentielles. L'originalité des contacts surfaciques par rapport aux autres liaisons est qu'il possède trois types de comportement, que nous allons passer en revue. Dans tous les cas, l'accélération a_n ne peut être que répulsive (sans quoi les solides s'interpénètreraient) tout comme la force f_n car nous ne considérons pas les solides "collants". Nous prenons comme convention de signes d'écrire ces deux conditions:

$$a_x \geq 0$$
$$f_x \geq 0$$

Une accélération normale positive se traduit par un décollement des surfaces en contact. La force est alors nulle. Réciproquement, une force normale non nulle garantit $a_n = 0$. Nous devons donc dans tous les cas satisfaire l'équation:

$$a_n f_n = 0$$

Certains auteurs justifient cette équation par le fait que les forces normales sont conservatives[Bar94], c'est à dire qu'elles ne dissipent pas d'énergie. Cette interprétation est incorrecte, car les forces élastiques sont elles aussi conservatives et ne vérifient pas une telle équation.

Frottement statique

Le frottement statique, aussi appelé adhérence, se produit quand la vitesse relative est nulle. La force tangentielle est alors limitée en fonction de la force normale par une valeur appelée *coefficient de frottement*, noté μ. Le mouvement est alors régi par les contraintes dynamiques suivantes:

$$\begin{aligned} a_x &= 0 \\ a_y &= 0 \\ a_z &= 0 \\ |f_t| &\leq \mu f_n \end{aligned} \tag{6.1}$$

Cette dernière condition stipule que la force de contact doit se situer à l'intérieur du "cône de frottement"représenté sur la figure (6.5).

La force tangentielle ne peut en aucun cas sortir du cône de frottement. Si la force tangentielle nécessaire au maintien des conditions $a_y = 0$ et $a_z = 0$ est excessive, on la limite à μf_n et on renonce aux contraintes sur l'accélération tangentielle. L'accélération tangentielle devient alors non nulle, la vitesse tangentielle devient elle aussi non nulle et le contact passe dans l'état de frottement glissant.

Frottement glissant

Le frottement glissant, aussi appelé frottement dynamique, se caractérise par une vitesse relative non nulle. Les forces de frottement s'opposent alors à cette vitesse au maximum de leur intensité possible, et dans une direction strictement opposée au glissement. Si nous posons $v_t = \sqrt{v_y^2 + v_z^2}$, les équations sont alors:

$$\begin{aligned} a_n &= 0 \\ f_y &= -\mu f_n v_y / v_t \\ f_z &= -\mu f_n v_z / v_t \end{aligned}$$

En réalité, le coefficient de frottement glissant est toujours un peu plus faible que le coefficient de frottement statique. Cela n'a pas d'importance pour la suite et nous négligerons ce phénomène pour simplifier les notations.

Décollement

Le cas le plus simple est le décollement du contact. Les surfaces s'éloignent et aucune force n'est possible. Les contraintes dynamiques sont alors les suivantes:

$$\begin{aligned} a_n &\geq 0 \\ f_x &= 0 \\ f_y &= 0 \\ f_z &= 0 \end{aligned}$$

6.2.2 Approche itérative globale des forces de contact

Nous désirons procéder par raffinement progressif d'une solution globale. Soient deux vecteurs $\boldsymbol{f}$ et $\boldsymbol{a}$ qui satisfont l'équation dynamique $\boldsymbol{D} \cdot \boldsymbol{f} + \boldsymbol{b} = \boldsymbol{a}$. Nous dirons que $(\boldsymbol{f}, \boldsymbol{a})$ est une solution dynamique de notre problème. Cette solution ne satisfait pas forcément toutes les inéquations de contact. Nous pouvons chercher une correction $(\boldsymbol{f}_1, \boldsymbol{a}_1)$ telle que $\boldsymbol{D} \cdot \boldsymbol{f}_1 = \boldsymbol{a}_1$. En l'ajoutant à la solution dynamique, nous obtenons une nouvelle solution dynamique $(\boldsymbol{f} + \boldsymbol{f}_1, \boldsymbol{a} + \boldsymbol{a}_1)$ puisque $\boldsymbol{D} \cdot (\boldsymbol{f} + \boldsymbol{f}_1) + \boldsymbol{b} = \boldsymbol{a} + \boldsymbol{a}_1$. L'idée de base de notre méthode est de construire itérativement une série de corrections qui améliorent à chaque itération notre solution dynamique. A chaque itération, la correction est construite de façon à annuler la partie incompatible de la solution dynamique courante. Comme selon chaque direction nous pouvons imposer soit la force soit le mouvement, la correction se répartit en quatre sous-vecteurs $(\boldsymbol{f}^*, \boldsymbol{f}', \boldsymbol{a}^*, \boldsymbol{a}')$. Les caractères * en exposant indiquent les valeurs imposées, et l'apostrophe indique les valeurs déduites. La dimension de $\boldsymbol{f}^*$ est égale à la dimension de $\boldsymbol{a}'$, et réciproquement. L'ensemble $\boldsymbol{f}$ de toutes les forces est l'union de $\boldsymbol{f}^*$ et $\boldsymbol{f}'$, et l'ensemble $\boldsymbol{a}$ de toutes les accélérations est l'union de $\boldsymbol{a}^*$ et $\boldsymbol{a}'$.

Classes de contacts

Nous savons que selon chaque direction indépendante il est possible d'imposer soit la force soit le mouvement. Nous partageons les valeurs de force et d'accélération des directions indépendantes en deux ensembles, H^* et H'. L'ensemble H^* contient des valeurs imposées, et l'ensemble H' contient les valeurs que nous déduisons des valeurs imposées. Ces deux ensembles sont complémentaires. L'ensemble H^* comprend les valeurs de $\boldsymbol{a}^*$ et $\boldsymbol{f}^*$, et H' contient les valeurs de $\boldsymbol{a}'$ et $\boldsymbol{f}'$.

Tout contact i peut être en frottement statique, frottement dynamique ou décollement. Si la vitesse tangentielle au contact est nulle, le contact est en frottement statique ou en décollement. En frottement statique, $a_{xi}, a_{yi}, a_{zi} \in H^*$ et $f_{xi}, f_{yi}, f_{zi} \in H'$. Les valeurs d'accélération imposée sont nulles. En décollement, $a_{xi}, a_{yi}, a_{zi} \in H'$ et $f_{xi}, f_{yi}, f_{zi} \in H^*$. Les valeurs de force imposée sont nulles.

Si la vitesse au contact est non nulle, le contact est soit en décollement soit en frottement dynamique. Le frottement dynamique se caractérise par $a_{xi} \in H$ et $a_{yi}, a_{zi}, f_{xi}, f_{yi}, f_{zi} \in H'$. Dans ce cas, le système d'équations doit inclure les relations (6.2) et (6.2) entre force normale, vitesse tangentielle et forces tangentielle.

La question est de savoir comment répartir les variables dans ces deux ensembles pour que les conditions de contacts soient partout satisfaites. Nous créons une première solution dynamique en imposant des hypothèses arbitraires. En pratique, nous choisissons de commencer avec des forces nulles. Si ces hypothèses se traduisent par des valeurs incompatibles, nous mettons à jour les ensembles et spécifions une correction comme expliqué ci-dessous.

6.2.3 Résolution de contraintes mixtes

La résolution de contraintes composées de forces imposées et de déplacements imposés s'effectue en trois temps:

1. appliquer les forces imposées;

2. résoudre les contraintes de mouvement;

3. calculer les déplacements.

Toutes les forces sont connues à l'issue de la deuxième étape. Tous les déplacements sont connus à la fin de la troisième étape. Plus formellement, on peut faire un découpage théorique du système d'équation sous la forme:

$$\begin{pmatrix} \boldsymbol{D}^{**} & \boldsymbol{D}^{*'} \\ \boldsymbol{D}^{'*} & \boldsymbol{D}'' \end{pmatrix} \cdot \begin{pmatrix} \boldsymbol{f}^{*} \\ \boldsymbol{f}' \end{pmatrix} = \begin{pmatrix} \boldsymbol{a}' \\ \boldsymbol{a}^{*} \end{pmatrix}$$

et effectuer la résolution:

$$\boldsymbol{D}''\boldsymbol{f}' = \boldsymbol{a}^{*} - \boldsymbol{D}^{'*}\boldsymbol{f}^{*}$$

puis calculer les mouvements inconnus:

$$\boldsymbol{a}' = \boldsymbol{D}^{**}\boldsymbol{f}^{*} + \boldsymbol{D}^{*'}\boldsymbol{f}'$$

La résolution peut s'effectuer avec n'importe quel algorithme de résolution dynamique, en particulier la résolution mixte itérative présentée au chapitre précédent. Nous considérons maintenant disposer d'une procédure $rsolution(\boldsymbol{a}^{*}, \boldsymbol{f}^{*}, \boldsymbol{a}', \boldsymbol{f}')$ qui à partir des accélérations imposées $\boldsymbol{a}^{*}$ et des forces imposées $\boldsymbol{f}^{*}$ calcule les forces nécessaires $\boldsymbol{f}'$ et les accélérations non spécifiées $\boldsymbol{a}'$.

6.2.4 Amélioration itérative des solutions dynamiques

La spécification de correction consiste à examiner toutes les variables de la solution dynamique, évaluer leur compatibilté avec les contraintes de contact, et si nécessaire, spécifier une correction.

Cet algorithme a été initialement implanté et présenté[Fau96] en utilisant la résolution dynamique présentée au chapitre 4.3. Il a été testé et a donné satisfaction sur des petites scènes (quelques contacts). La norme des forces de correction se réduit à chaque itération, d'un facteur voisin de 0.5 en moyenne. On peut voir cet algorithme comme une redistribution des forces et mouvements incompatibles dans le système à chaque itération. On peut tendre vers une solution correcte sans jamais l'atteindre. On sait toutefois que si les forces redistribuées sont faibles, la solution correcte est très voisine de la solution courante. Il reste encore des améliorations à apporter pour traiter les frottements. En particulier, un contact glissant ne peut pas redevenir collant. Ceci peut entraîner un début de glissement alors que les forces de frottement ne sont pas au maximum. En pratique, sur des "petites" scènes dont on initialise tous les contacts dans l'état collé (sauf bien sûr ceux dont la vitesse de glissement est non nulle), il donne des résultats satisfaisants.

On peut simplifier l'algorithme quand les contacts sont sans frottements. Il donne alors entière satisfaction même pour des scènes comportant plusieurs centaines de contacts.

Collisions

Les forces de contact interviennent aussi lors des collisions. On peut alors effectuer le même traitement sur les vitesses que sur les accélérations, avec toutefois une différence:

recommencer
  pour chaque contact i
    si $a_{xi} \in H'$
      si $a_{xi} < 0$
        $a_{xi}* \ = \ -a_{xi}$
        si $v_{ti} \neq 0$
          mettre a_{xi} dans H^*
          mettre $a_{yi}, a_{zi}, f_{xi}, f_{yi}, f_{zi}$ dans H'
        sinon
          mettre a_{xi} dans H^*
          mettre f_{xi} dans H'
      sinon
        si $f_{xi} < 0$
          mettre a_{xi}, a_{yi}, a_{zi} dans H'
          mettre f_{xi}, f_{yi}, f_{zi} dans H^*
          $f^*_{xi} \ = \ -f_{xi}$
          $f^*_{yi} \ = \ -f_{yi}$
          $f^*_{zi} \ = \ -f_{zi}$
        sinon
          si $f^2_{yi} + f^2_{zi} > \mu^2 f^2_{xi}$
            mettre f_{yi}, f_{zi} dans H^*
            mettre a_{yi}, a_{zi} dans H'
            $f_t \ = \ \sqrt{f^2_{yi} + f^2_{zi}}$
            $F_{tM} \ = \ \mu f_{xi}$
            $f^*_{yi} \ = \ (f_{tM}/f_t - 1) f_{yi}$
            $f^*_{zi} \ = \ (f_{tM}/f_t - 1) f_{zi}$
  $resolution(\boldsymbol{a}^*, \boldsymbol{f}^*, \boldsymbol{a}', \boldsymbol{f}')$
  $\boldsymbol{a} \ += \ (\boldsymbol{a}', \boldsymbol{a}^*)$
  $\boldsymbol{f} \ += \ (\boldsymbol{f}^*, \boldsymbol{f}')$
  tant que $\|(\boldsymbol{a}', \boldsymbol{a}^*)\| > \epsilon$

FIG. 6.6 - *Calcul itératif des forces de contact*

la notion de frottement dynamique n'existe plus. Dans tous les cas les forces tangentielles peuvent varier entre 0 et μf_x si elles sont suffisantes pour annuler la vitesse tangentielle.

C'est sous forme de traitement de collisions que nous avons choisi de calculer les forces de contact dans notre boucle d'animation. L'avantage est que cela évite de se trouver dans des cas de configuration inconsistante qui existent avec le frottement dynamique[Bar91]. Il existe en effet des configurations de frottement dynamique où la seule force possible pour garder nulle l'accélération tangentielle est une force attractive.

6.3 Conclusion

La boucle d'animation présentée inclut la prise en compte de forces extérieures, d'actions de liaisons et de collisions. La prise en compte différée des collisions permet de traiter en une résolution globale l'ensemble des collisions intervenues au cours d'un pas de temps. On évite ainsi de multiples retours en arrière. Les erreurs de positions dûs aux erreurs d'intégration sont compensées en effectuant une résolution en position basée sur l'approximation des mouvements linéarisés. On évite ainsi d'avoir recours à des PD-contrôleurs dont la mise au point peut être délicate. Les pas de temps utilisables sont limités par des considérations énergétiques et non par des problèmes géométriques. Un algorithme de calcul des forces de contact par résolution globale itérative a été proposé. Encore perfectible pour les contraintes de Coulomb, il fonctionne de manière satisfaisante pour les contacts sans frottement et a permis le traitement de scènes comportant plusieurs centaines de contact. Il permet d'approcher une solution compatible avec les contraintes de contact en un temps qu'on peut limiter par un nombre donné d'itérations.

Conclusion

Au terme de ces trois années de travail sur l'animation de solides articulés, environ 25000 lignes de code C++ et 20000 lignes de langage d'interface tcl ont été écrites. Une bibliothèques d'articulations et les algorithmes d'animation correspondants ont été crées, et le tout est en cours d'intégration dans la plate-forme commune d'animation de l'équipe iMAGIS. Le cheminement n'a pas été rectiligne mais plutôt semblable à une réduction itérative d'incohérence intellectuelle. On aura peut-être trouvé des traces de ces errements à la lecture de ce document. Néanmoins, diverses approches pour la synthèse d'animation ont été testées et des méthodes nouvelles ont été imaginées et implantées. Certaines se sont révélées efficaces et semblent indiquer des voies à poursuivre.

L'approche cinématique consistant à caractériser l'ensemble des mouvements autorisés par les contraintes permet d'effectuer des choix parmi un ensemble de mouvements autorisés. Les contraintes cinématiques peuvent être organisées suivant une hiérarchie comportant un nombre arbitraire de niveaux. Le principe des travaux virtuels permet de lever les indécisions restantes en appliquant les lois de la dynamique. Les choix sont malheureusement à courte vue puisque l'ensemble des mouvements possibles doit être redéfini à chaque instant en fonction de la configuration géométrique. De plus, les temps de calcul nécessaires à la décomposition spectrale de la matrice des contraintes sont prohibitifs pour des applications en temps réel. En fin de compte, cette approche "active" a été abandonnée au profit de méthodes "réactives" considérablement plus rapides.

Trois approches prometteuses pour la prise en compte des contraintes de temps réel ont été imaginées et concrétisées. Le calcul itératif des forces de contact est encore perfectible sur la question des frottements mais s'est montré applicable à des scènes comportant plusieurs centaines de contacts. La résolution itérative des chaînes fermées s'est montrée plus rapide que tous les autres algorithmes sur de nombreux exemples, avec une précision réglable. La boucle d'animation avec contraintes de déplacements et prise en compte différée des collisions s'est avérée efficace. Ces trois algorithmes sont combinables et peuvent permettre de créer un véritable système d'animation en temps réel. Aucune matrice dense autre que les matrices de masse des solides n'est utilisée. L'espace mémoire utilisé reste donc strictement proportionnel au nombre de solides et de contraintes, ce qui laisse entrevoir la possibilité de traiter des gros problèmes sur des machines modestes.

Ce système temps réel reste à construire. Cela passe par la réimplémentation soigneuse des algorithmes, et leur intégration dans une architecture logicielle apte à gérer les contraintes temporelles. Les travaux futurs devraient aussi inclure l'animation d'objets déformables. L'approche dynamique avec multiplicateurs de

Deuxième partie

Approche Photométrique pour la Modélisation de Surfaces à Base d'Images

Introduction

La synthèse d'images consiste à faire fabriquer par un ordinateur des images d'un monde plus ou moins imaginaire. L'utilisateur fournit des données qui décrivent les objets contenus dans une scène, le point de vue d'où l'on se place pour l'observer, ainsi que certaines spécifications comme par exemple la position et la nature des sources lumineuses. La visualisation d'une surface nécessite d'une part la connaissance géométrique de celle-ci, et d'autre part certains paramètres physiques qui décrivent sa façon d'interagir avec la lumière, qu'on regroupe dans ce qu'on appelle son *modèle de réflectance*.

Avec les techniques actuelles, la modélisation d'objets un tant soit peu complexes nécessite encore beaucoup de temps et de dextérité. Dans le cas où l'objet que l'on désire représenter existe dans la réalité ou ressemble fortement à un objet réel, il peut être extrêmement difficile de parvenir à un résultat ressemblant à l'original. On doit en effet construire une "maquette numérique" respectant la géométrie de l'objet ainsi que ses propriétés lumineuses.

La mesure directe de la réflectance est longue et délicate puisque quatre paramètres sont nécessaires pour décrire la direction incidente et la direction de vue. Même si des techniques ont vu le jour qui simplifient considérablement cette opération, elle nécessite toujours la mise en place d'un échantillon plan de matière dans un appareil spécial, ce qui ne va pas sans poser de nombreux problèmes pratiques.

Quant à la géométrie, la mesure directe est aussi coûteuse et compliquée puisqu'elle nécessite un télémetre laser, ou une machine à mesurer tridimensionnelle.

Nous nous intéresserons ici à fournir des outils permettant, étant donné un objet à représenter, de connaître facilement et rapidement sa géométrie et les caractéristiques de sa fonction de réflectance, pour l'inclure tel quel dans une scène.

La façon la plus simple de "capturer" un objet dans un ordinateur est d'en prendre des images à l'aide d'une caméra numérique, matériel désormais courant. C'est pourquoi on désire trouver une technique dont la mise en oeuvre soit simple et qui permette, à base d'images d'un objet, de reconstruire sa géométrie et modéliser sa fonction de réflectance.

Les différentes techniques actuelles de vision artificielle se répartissent approximativement entre deux paradigmes: stéréovision géométrique ou photométrie. Le premier s'appuie sur les relations géométriques entre les images d'un même point vu de différentes directions. Le résultat de l'analyse d'une scène est un nuage de points dans un repère spatial. La photométrie se base sur la quantité de lumière émise par un point en fonction des directions d'éclairage et de vue, et en déduit le vecteur normal à la surface en tout point de l'image. L'obtention de ce résultat nécessite d'inverser un système d'équations construit à partir de la description mathémathique de la réflectance.

Pour produire des images réalistes il est nécessaire de connaître en tout point la normale aux surfaces. Nous nous intéresserons donc aux approches photométriques. Le pre-

mier chapitre de ce document rassemble des rappels physiques utiles ainsi qu'un état de l'art du domaine. Le deuxième chapitre présente un nouvel algorithme de stéréovision photométrique. Le troisième présente une méthode de modélisation simultanée de la géométrie et de la réflectance d'une surface, et des tests sur des images de synthèses et des images réelles.

Chapitre 1

État de l'art

1.1 Rappels

Toutes les surfaces ne réfléchissent pas la lumière de la même façon. Intuitivement, la principale caractéristique est la couleur. L'aspect mat ou brillant est aussi extrêmement important pour le réalisme d'une image. Qualitativement, la réflection peut avoir deux types de comportement: spéculaire ou diffus. Nous allons d'abord caractériser ces deux modes de réflection, et montrer les modèles mathématiques correspondants. Puis nous verrons certaines variantes introduites pour affiner la modélisation. Nous terminerons ce chapitre par une approche physique du phénomène qui débouche sur des modèles encore plus perfectionnés.

1.1.1 Définitions

La lumière est une onde électromagnétique qui correspond à un dégagement d'énergie évacuée par des photons. Ce dégagement est caractérisé par une puissance qui s'exprime en $Watt$ (W). La répartition de cette puissance dans les directions de l'espace s'exprime en Watts par stéradians (W/st). La puissance reçue par élément de surface s'exprime en W/m^2. Ramenée à chaque direction de l'espace, cette puissance s'exprime en $W/m^2.st$, et cette grandeur que nous appellerons *intensité* correspond à la notion d'intensité lumineuse perçue par nos yeux en provenance d'une certaine direction, comme illustré sur la figure (1.1). C'est donc une valeur correspondant à cette grandeur qui doit être affichée en chaque pixel de l'image. Dans toute la suite nous désignerons l'intensité lumineuse par la lettre L. Nous distinguerons l'intensité incidente L_s provenant des sources de lumière de l'intensité réfléchie L_r qui repart de la surface.

1.1.2 Réflection spéculaire et diffuse

La réflection spéculaire a été étudiée géométriquement par Descartes. Elle correspond au comportement d'un miroir, et est illustrée sur la figure (1.2). Nous utiliserons les caractères gras pour les notations de vecteurs. Les vecteurs $\boldsymbol{n}$, $\boldsymbol{s}$ et $\boldsymbol{r}$ représentent respectivement la normale à la surface, la direction de provenance de la lumière et la direction de réflection spéculaire.

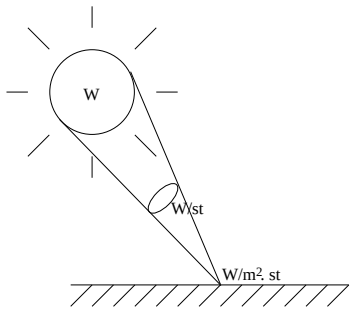


FIG. 1.1 - *Intensité lumineuse*

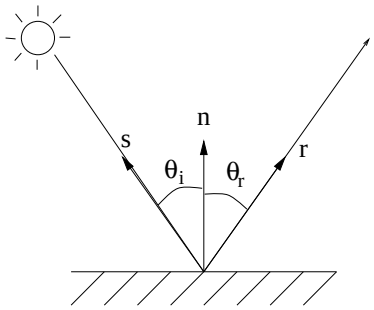


FIG. 1.2 - *Réflection spéculaire:* $\theta_i = \theta_r$

L'intensité réfléchie dans la direction $\mathbf{v}$ a pour expression:

$$L_r = \rho_s L_s \quad si \;\; \boldsymbol{v} = \boldsymbol{r}$$
$$L_r = 0 \quad sinon$$

où ρ_s est un coefficient de réflection inférieur ou égal à 1.

Au contraire, la réflection diffuse est caractéristique des surfaces mattes et correspond, dans le cas idéal décrit par Lambert en 1760 [Lam60], à la redistribtion uniforme de la lumière dans toutes les directions du demi-espace défini par la normale. Elle est illustrée sur la figure 1.3. L'intensité réfléchie a pour expression:

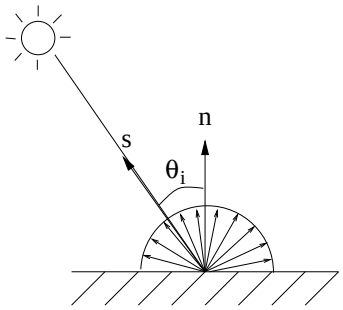


FIG. 1.3 - *Réflection lambertienne: redistribution uniforme*

$$L_r = \rho_l \; \boldsymbol{n}.\boldsymbol{s} \; L_s$$

où ρ_l est un coefficient de réflection parfois appelé *albedo*. Ces deux modèles, même combinés, sont toutefois insuffisants pour décrire tous les modes de réflection lumineuse. En

effet, la réflection diffuse n'est pas forcément uniforme et pour la plupart des surfaces elle semble s'effectuer majoritairement dans la direction spéculaire.

Dans le cas général de réflection diffuse, pour une normale et une direction incidente données, l'intensité réfléchie est fonction de la direction de réflection considérée. On peut décrire cette fonction par une surface centrée sur le point de réflection. Cette surface est souvent appelée un *lobe*. Dans le cas lambertien, la réflection est uniforme, la fonction de réflection est donc constante et le lobe correspondant est une demi-sphère.

Phong[Pho75] introduisit un lobe supplémentaire pour représenter le phénomène de réflection majoritaire dans la direction spéculaire. La fonction de réflectance devient alors la somme de trois termes correspondant à la réflection spéculaire, la réflection lambertienne et la réflection qualifiée de *spéculaire diffuse* comme l'illustre la figure 1.4.

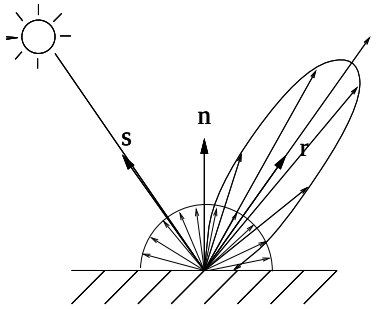


FIG. 1.4 - *Le modèle de Phong*

Le lobe supplémentaire de Phong a pour expression:

$$L_d = \rho_d \, \boldsymbol{v}.\boldsymbol{r} \, L_s \quad si \; \boldsymbol{v}.\boldsymbol{r} > 0$$
$$L_d = 0 \quad sinon$$

Ce modèle fut perfectionné par Blinn [Bli77] en remplaçant $\boldsymbol{v}.\boldsymbol{r}$ par $\boldsymbol{n}.\boldsymbol{h}$ (fig. 1.5) où $\boldsymbol{h}$ est le vecteur bissecteur de $\boldsymbol{s}$ et $\boldsymbol{v}$, i.e. $\boldsymbol{h} = (\boldsymbol{s}+\boldsymbol{v})/\|\boldsymbol{s}+\boldsymbol{v}\|$. Il n'est pas adapté à certains

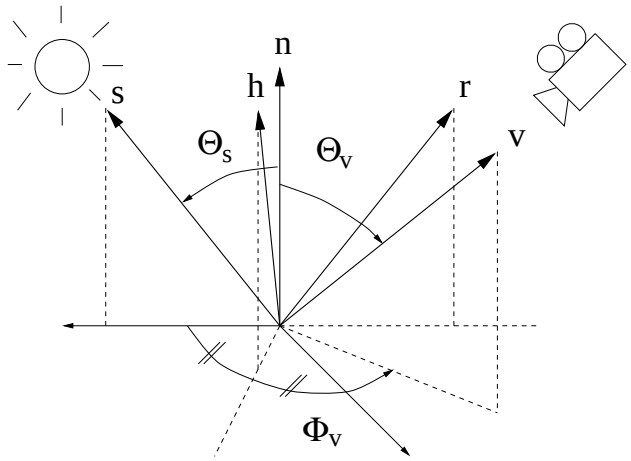


FIG. 1.5 - *Vecteurs et angles*

algorithmes de rendu[SP94] basés sur la quantification des échanges énergétiques. En effet, dans certaines conditions, la quantité totale d'énergie réfléchie peut être supérieure à la

quantité d'énergie incidente. Pour identifier et corriger ce problème, il est nécessaire de mener une étude plus approfondie du phénomène physique de réflection lumineuse.

1.1.3 Approche physique

L'approche physique du phénomène de réflection lumineuse peut se faire à deux niveaux: d'une part, la modélisation microscopique des surfaces permet de prévoir assez finement la répartition de lumière réfléchie; d'autre part, la prise en compte du principe de conservation de l'énergie induit des modifications à apporter à des modèles empiriques afin de les rendre physiquement plausibles. Notons au préalable que de par sa nature d'onde électromagnétique, la lumière peut se comporter différemment suivant la longueur d'onde considérée. Toutefois ces différences restent purement quantitatives tant qu'on se limite au spectre visible. Par souci de clarté, nous ne ferons pas intervenir les longueurs d'onde dans les formules qui suivent, tout en gardant à l'esprit que les valeurs des coefficients en dépendent.

Approche microscopique

Vues de très près, les surfaces ne sont pas lisses. Pour pouvoir les traiter géométriquement, on les considère comme des ensembles de microfacettes à différentes hauteurs dont les orientations obéissent à une répartition aléatoire gaussienne. En plus des réflections diffuse et directe, trois phénomènes entrent alors en jeu: les réflections multiples, l'auto-ombrage, et l'auto-masquage. Ces trois phénomènes sont illustrés sur la figure 1.6.

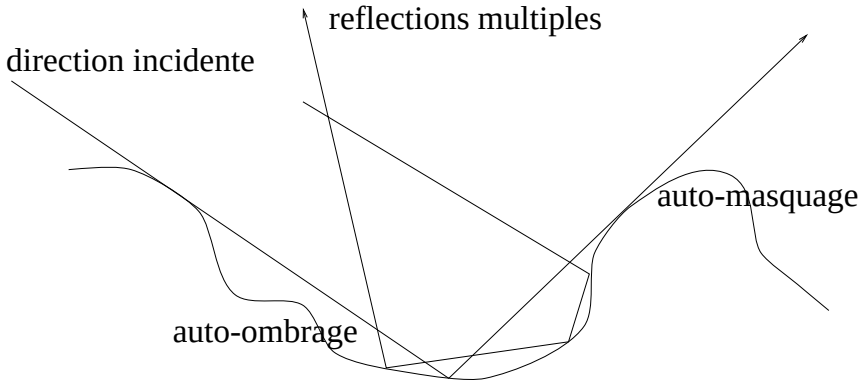


FIG. 1.6 - *Phénomènes dus à la rugosité des surfaces*

La prise en compte de ces phénomènes débouche sur des modèles de réflection tels que [TS67] et [HTSG91]. Signalons que:

- Contrairement au modèle de Phong/Blinn, la réflection diffuse maximum peut être légèrement décalée par rapport à la direction de réflection spéculaire.

- Plusieurs lobes peuvent apparaître, dont notamment un dans la direction incidente *(backscatter lobe)*. Cet effet est explicitement recherché lors de l'élaboration de matériaux réfléchissants comme par exemple les peintures de signalisation routière.

– Certaines surfaces présentent une répartition hétérogène des orientations des microfacettes qui se traduit par des effets d'anisotropie. On peut imaginer, par exemple, un ensemble de cylindres parallèles. Il faut alors tenir compte de l'azimuth pour définir la direction incidente[War92].

Approche énergétique

L'approche énergétique consiste à faire le bilan de l'énergie incidente et de l'énergie réfléchie pour chaque élément de surface. Conformément aux définitions données au paragraphe 1.1.1, la puissance surfacique provenant de la direction $\boldsymbol{s}$ vaut $L_s\ d\omega_s$, en W/m^2. La portion de surface dA de normale $\boldsymbol{n}$ représentée sur la figure 1.7 reçoit donc une puissance de: $L_s.dS.d\omega_s$ (W). L'élément de surface dA étant la projection de dS sur la surface parallèlement à $\boldsymbol{s}$, on a:

$$dS = \cos\theta_s dA = \boldsymbol{n}.\boldsymbol{s}dA$$

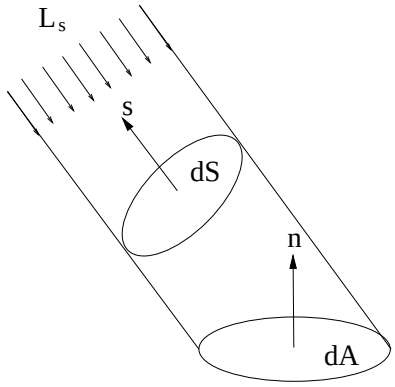


FIG. 1.7 -

L'énergie d^2E_s reçue d'une direction $\boldsymbol{s}$ par un élément de surface dA de normale $\boldsymbol{n}$ est donc donnée par la relation:

$$d^2E_s = \boldsymbol{n}.\boldsymbol{s}.dA.L_s.d\omega_s\ \ (W)$$

où L_s est la luminosité dans l'angle solide $d\omega_s$ centré sur la direction de $\boldsymbol{s}$.

En terme de flux énergétique, la relation s'exprime sous la forme:

$$d^2E_s = d\Phi_s.dA$$

où $\Phi_s = \frac{dE_s}{dA}$ est le flux énergétique (W/m^2). Une partie de ce flux est absorbé et chauffe la matière, tandis que le reste est réfléchi de façon spéculaire et/ou diffuse comme nous l'avons vu précédemment.

Pour un observateur situé dans une direction $\boldsymbol{v}$, la réflection de ce flux contribue à l'intensité lumineuse de la surface d'une valeur dL_r proportionnelle à la valeur de ce flux et fonction des directions d'éclairement et d'observation:

$$dL_r(\boldsymbol{s}, ,v) = f_r(\boldsymbol{s}, \boldsymbol{v}).d\Phi_s$$

La fonction f_r caractérise la redistribution de l'énergie incidente dans toutes les directions, elle est communément appelée *Bidirectionnal Reflection Distribution Function*

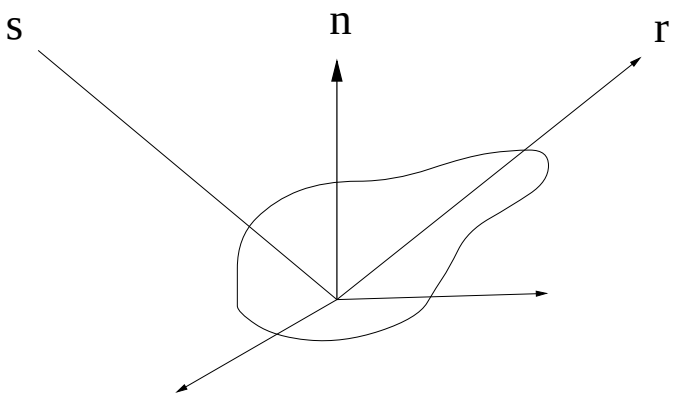


FIG. 1.8 - *Répartion de l'énergie réfléchie*

(BRDF) et s'exprime en st^{-1}. A toute direction incidente $\boldsymbol{s}$, elle fait correspondre une répartition d'énergie réfléchie comme l'illustre la figure 1.8.

La luminosité d'un point vu d'une direction $\boldsymbol{v}$ est la somme des contributions de toutes les directions incidentes de l'espace en ce point, on a donc:

$$L_r(\boldsymbol{v}) = \int_{\Omega_n} f_r(\boldsymbol{s}, \boldsymbol{v}) \boldsymbol{n}.\boldsymbol{s} L_s(\boldsymbol{s}).d\omega_s$$

où Ω_n est le demi-espace centré sur la direction $\boldsymbol{n}$.

Dans le cas particulier d'une source unique à l'infini, toute la lumière provient d'une même direction et la relation devient:

$$L_r(\boldsymbol{v}) = \phi_s.(\boldsymbol{n}.\boldsymbol{s}) f_r(\boldsymbol{s}, \boldsymbol{v}) \tag{1.1}$$

De manière similaire à la puissance surfacique incidente, on peut quantifier la puissance surfacique $d\Phi_r$ réfléchie dans une direction $\boldsymbol{v}$ par:

$$d\Phi_r = \boldsymbol{n}.\boldsymbol{v}\ L_r(\boldsymbol{v}) d\omega_r$$

La conservation de l'énergie nécessite que l'énergie réfléchie soit toujours inférieure à l'énergie incidente quelle que soit la direction d'éclairage. On a donc à satisfaire:

$$\int_{\Omega_n} f_r(\boldsymbol{s}, \boldsymbol{v}) \boldsymbol{n}.\boldsymbol{v} d\omega_r \leq 1$$

pour toute direction incidente $\boldsymbol{s}$, où $\boldsymbol{v}$ est la variable d'intégration. Cette valeur est généralement appelée *réflectance directionnelle-hémisphérique.*
Les contraintes de plausibilité physique de la BRDF portent sur deux aspects:

- Réciprocité: $\boldsymbol{s}$ et $\boldsymbol{v}$ doivent être interchangeables dans la BRDF ([SH81]). Le modèle de Blinn est supérieur au modèle de Phong pour cette raison;

- Conservation de l'énergie: le rapport entre énergie incidente et énergie réfléchie doit rester inférieur à 1 pour toute configuration géométrique. Lewis[Lew93] propose des modifications à apporter à la plupart des modèles classiques de BRDF dans ce but.

1.1.4 Conclusion

Cette brève description du phénomène de réflection lumineuse n'est pas une revue exhaustive mais dégage les éléments qui doivent intervenir lors du choix d'un modèle:

- Pour représenter un large éventail de surfaces, il est important de prendre en compte la réflection spéculaire et la réflection diffuse. La réflection diffuse doit pouvoir comporter plusieurs lobes centrés sur des directions différentes.

- Pour être utilisable par tous les algorithmes de rendu, la réflection doit satisfaire la conservation de l'énergie.

1.2 Shape from Shading

1.2.1 Principe

Le *Shape from Shading*, qui pourrait se traduire par *La Forme par l'Ombrage* consiste à reconstruire la forme d'une surface à partir des quantités de lumière réfléchies par celle-ci. Il n'existe quasiment pas de littérature française sur le sujet bien que ce soit un des problèmes les plus captivants de la vision artificielle. C'est pourquoi on trouvera dans la suite de ce chapitre un certain nombre de termes anglais, difficiles à traduire en français de manière compacte et élégante.

Intuitivement, il paraît assez simple de déduire une forme à partir d'une image comme celle de la figure 1.9. Pourtant, l'expérience semble montrer que notre cerveau met en oeuvre des procédés beaucoup plus subtils et heuristiques que tous les algorithmes jamais développés pour résoudre le problème.

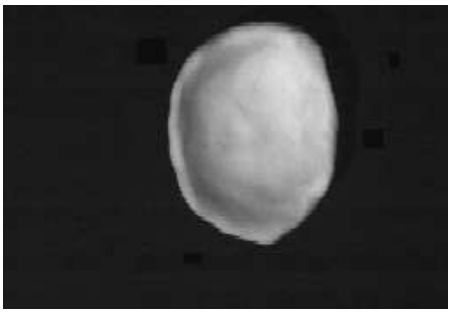


FIG. 1.9 - *Une forme ombrée*

Formellement, le problème se pose de la façon suivante: étant donné un champ de luminosité $L(x_i, y_j)$ défini en chaque pixel d'une image, retrouver le champ de hauteur $z(x_i, y_j)$ correspondant à la surface représentée sur l'image.

Pour ceci, notre seul outil est la connaissance du phénomène de réflection lumineuse tel qu'il est présenté au paragraphe 1.1. En tout point il fait intervenir la normale locale, la réflectance de la surface ainsi que les directions d'éclairage et de vue. Nous allons voir par la suite que de fortes hypothèses sur tous ces paramètres sont nécessaires afin de pouvoir obtenir un résultat, et qu'il est utile de disposer de plusieurs images de la même surface prises dans des conditions différentes.

En pratique, on trouve schématiquement deux classes d'approches dans l'état de l'art: les premières utilisent une seule image et ne peuvent traiter que des surfaces lisses et lambertiennes; les autres utilisent plusieurs images, peuvent traiter des modèles de réflectance plus évolués et nécessitent moins d'hypothèses sur la surface. Toutes utilisent de fortes hypothèses sur la réflectance de la surface.

Les premières applications portèrent sur l'analyse de photos aériennes ou astronomiques [Rin66]. La plupart des travaux portent sur l'analyse d'une image unique. En effet, la mise en correspondance d'images prises de différents points de vue est un problème encore largement ouvert, et il est rare de maîtriser les conditions d'éclairage. Pour la présentation de cette approche, nous nous bornerons à formaliser le problème et en présenter les principales pistes de recherche. On peut trouver plus d'approfondissement dans [Hor90, HB89, Hor86]. Nous examinerons ensuite les approches multi-images.

Hypothèses simplificatrices

Par la suite, on considèrera que:

- la surface est suffisamment lisse pour être considérée dérivable;
- la réflectance est la même en tout point de la surface;
- la formation de l'image se fait selon les règles de projection orthographique;
- par défaut, la surface est isotrope.

La première hypothèse est nécessaire pour utiliser l'idée de normale et pouvoir en déduire les hauteurs. La deuxième est nécessaire uniquement dans le cas où l'on ne dispose que d'une seule image. La troisième n'est pas indispensable mais simplifie les équations et leur traitement. On l'adopte pour clarifier le discours. Le lecteur intéressé par le cas de projection perspective peut se reporter à [Rin66].

Images

Une image est une grille rectangulaire de valeurs généralement entières comprises entre 0 et 255, telle que fournie par une caméra CCD. Les valeurs correspondent à des luminosités et le champ de luminosité ainsi représenté apparait souvent dans la littérature sous la notation $E(x, y)$.

Surfaces

La grande variété des surfaces représentées sur des images empêche généralement de choisir un modèle analytique prédéfini dont on chercherait à adapter les coefficients afin d'obtenir la meilleure reconstruction. C'est pourquoi on se limite à représenter la surface comme un ensemble de points $z(x_i, y_j)$ répartis sur une grille régulière correspondant aux pixels de l'image.

Le modèle de réflectance faisant intervenir la normale à la surface, il faut faire l'hypothèse d'une surface dérivable ou d'un ensemble de facettes indépendantes. Ce dernier modèle ne permettant pas de reconstruire les hauteurs à partir des orientations, on garde la première hypothèse, la surface est considérée dérivable et la normale en tout point peut

prendre différentes formulations [Hor90]. Par la suite nous adopterons la plus classique qui est de représenter la normale par un vecteur $\boldsymbol{n}$:

$$\boldsymbol{n} = \frac{1}{\sqrt{1+p^2+q^2}}.(-p,-q,1)^t$$
$$avec\ \ p = \frac{\delta z}{\delta x}\ \ et\ \ q = \frac{\delta z}{\delta y}$$

où l'exposant t exprime la transposition matricielle.

Cartes de réflectance

La carte de réflectance rassemble la connaissance que l'on a de la réflectance de la surface et des conditions d'éclairage. Elle traduit la correspondance entre luminosité et normales correspondantes possibles, c'est donc une fonction de p et q et on la note généralement $R(p,q)$. Les valeurs correspondent au rapport entre intensité réfléchie et incidente.

Visuellement, on la représente par les courbes de niveau du champ $R(p,q)$ comme illustré sur la figure 1.10. Ces courbes de niveaux correspondent à des valeurs constantes de luminosité, on les appelle donc des *isophotes*.

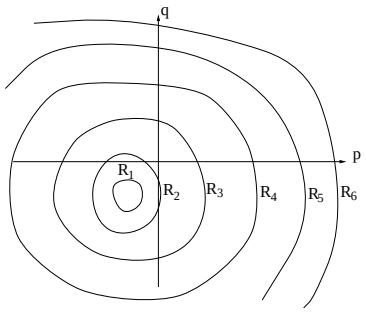


FIG. 1.10 - *Une carte de réflectance*

Elle peut être déterminée de trois façons:

- expérimentalement à l'aide d'un échantillon de surface et d'un photogoniomètre;
- par calibration à l'aide d'un objet de forme connue (sphère);
- par intégration de la BRDF dans des conditions d'éclairage bien spécifiées.

Pour associer les valeurs des pixels d'une image à une carte de réflectance, il faut préalablement multiplier ces valeurs par un facteur qui dépend de la puissance d'éclairage et des caractéristiques de la caméra (réponse des capteurs, ouverture du diaphragme). Ce coefficient étant inconnu en général, il convient de le déterminer expérimentalement. On peut par exemple faire l'hypothèse que le point le plus lumineux de la surface correspond à la valeur maximale de la carte de réflectance.

Les cartes de réflectance sont utiles pour accélérer les calculs car elles sont des fonctions de transfert entre normale et intensité réfléchie qui prennent en compte la BRDF, l'éclairage et le point de vue.

Problème à résoudre

Conformément à toutes les hypothèses et définitions qui précèdent, le problème est de trouver les valeurs $z(i,j)$ telles que:

$$E(i,j) = R(p(i,j), q(i,j)) \tag{1.2}$$

Selon les algorithmes, la détermination des z s'effectue simultanément aux orientations, ou après par intégration.

1.2.2 Image unique

Dans le cas d'une image unique, le problème est *a priori* sous-contraint car à une valeur de luminosité donnée peuvent correspondre une infinité de normales comme on peut le voir sur une carte de réflectance ou sur la figure 1.11.

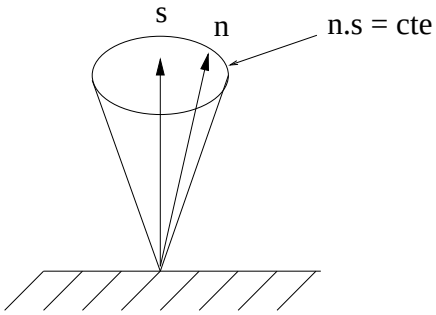


FIG. 1.11 - *Pour une certaine luminosité réémise par une surface lambertienne, l'ensemble des normales possibles repose sur un cône de révolution.*

La première solution proposée [Hor75] (repris dans [HB89]) consiste à intégrer en suivant des chemins particuliers appelés *bandes caractéristiques*. Partant d'un point, on en déduit un nouveau par:

$$\dot{x} = R_p, \quad \dot{y} = R_q, \quad \dot{z} = pR_p + qR_q, \quad \dot{p} = E_x, \quad \dot{q} = E_y$$

où une variable en indice désigne la dérivation par rapport à cette variable et le point la dérivation par rapport à l'abcisse curviligne sur une courbe dont on connaît en tout point la position et l'orientation.

Outre qu'elle nécessite de connaître au départ les points et orientations d'une courbe appartenant à la surface, cette méthode s'avère peu robuste en présence de bruit, ce qui arrive presque toujours en travaillant sur des données réelles. Dans le cas où la source de lumière provient de la direction d'observation, les chemins d'intégration correspondent aux lignes de plus grande pente et l'unicité de la solution a pu être prouvée [Oli89], en dehors de l'ambiguité entre convexe et concave.

Les meilleurs résultats ont été obtenus à partir de formulations variationnelles du problème. L'idée est de minimiser la fonctionnelle

$$\int\int (E(x,y) - R(p,q))^2 dxdy$$

Toutefois l'unicité de la solution ne peut être assurée qu'en ajoutant des contraintes supplémentaires [Bru82, Sax88]. Ces contraintes peuvent porter sur les conditions aux limites ou sur les points singuliers (maxima). La minimisation de fonctionnelles débouche assez naturellement sur des algorithmes itératifs [Hor90]. D'autres fonctionnelles ont été proposées pour régulariser le problème. On peut par exemple chercher la surface la plus lisse [IH81] en minimisant

$$\int\int (E(x,y) - R(p,q))^2 + \lambda(p_x^2 + p_y^2 + q_x^2 + q_y^2)dxdy$$

Toutefois, cet algorithme ne fournit pas une solution au problème initial mais à une variante imposant une solution plus lisse. Il s'avéra [Hor90] que même en fournissant la surface recherchée comme point de départ des itérations, l'algorithme s'en éloigne sensiblement sauf si celle-ci est déja très lisse.

Tous ces algorithmes portent uniquement sur le calcul de valeurs pour p et q en tout point. Il reste à en déduire les valeurs de z. La sommation des différences finies sur les valeurs de gradient obtenues par ces algorithmes donnent des résultats très décevants. Une meilleure solution consiste, une fois encore, à minimiser itérativement une fonctionnelle:

$$\int\int ((z_x - p)^2 + (z_y - q)^2)dxdy$$

Horn propose[Hor90] une fonctionnelle qui mène à de meilleures reconstructions de normales tout en fournissant simultanément les valeurs de z:

$$\int\int ((E(x,y) - R(p,q))^2 + \mu((z_x - p)^2 + (z_y - q)^2))dxdy$$

Cette approche peut aussi être combinée avec la contrainte de surface lisse, et en faisant varier interactivement les paramètres de pénalisation:

$$\int\int ((E(x,y) - R(p,q))^2 + \mu((z_x - p)^2 + (z_y - q)^2) + \lambda(p_x^2 + p_y^2 + q_x^2 + q_y^2))dxdy$$

Frankot et Chellapa[FC88] proposent de traiter le problème d'intégrabilité en projetant en tout point la solution courante sur le champ intégrable le plus proche.

Le problème du Shape from Shading à une seule image reste ouvert car tous les algorithmes proposés restent très sensibles au bruit, et la résolution itérative assez lente. Les algorithmes utilisant plusieurs images, quand ils sont utilisables, fournissent de bien meilleurs résultats et c'est vers eux que nous allons nous tourner à présent.

1.2.3 Stéréovision Photométrique

La stéréovision photométrique consiste à utiliser plusieurs images d'une même surface sous plusieurs éclairages de directions différentes pour calculer les normales, comme sur la figure 1.12.

Cette méthode fut introduite par Woodham [Woo80] pour détecter des défauts de surface sur des pièces métalliques moulées. La surface étant rugueuse, elle est considérée lambertienne et sa BRDF constante vaut ν. On dispose en chaque point de trois valeurs correspondant chacune à une direction d'éclairage différente, on peut donc écrire le

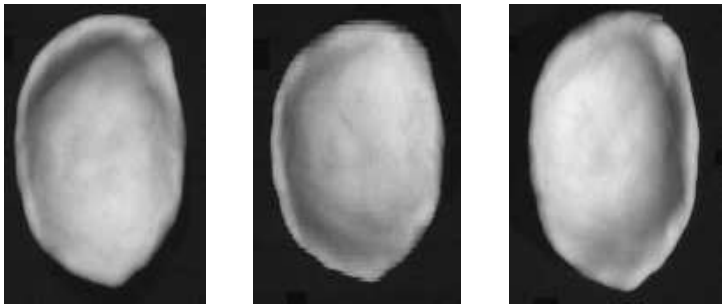


FIG. 1.12 - *Une même surface vue sous trois éclairages différents*

système:

$$L_1 = \nu L_1^* \boldsymbol{n}.\boldsymbol{s_1}$$
$$L_2 = \nu L_2^* \boldsymbol{n}.\boldsymbol{s_2}$$
$$L_3 = \nu L_3^* \boldsymbol{n}.\boldsymbol{s_3}$$

où L_i^* est la valeur de la luminosité maximale sur l'image i correspondant à $\boldsymbol{n}.\boldsymbol{s_i} = 1$. Il ne reste qu'à en déduire les composantes de la normale en résolvant un système linéaire de trois équations à trois inconnues. Une direction de l'espace pouvant être représentée par deux composantes indépendantes, on pourrait penser que deux équations suffisent mais certaines ambiguités subsisteraient comme l'illustre la figure 1.13.

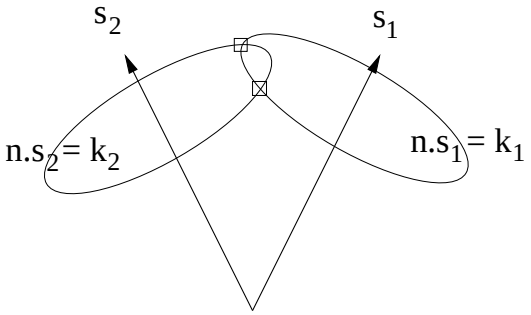


FIG. 1.13 - *Dans le cas général, trois équations sont nécessaires pour lever toute ambiguité.*

Pour les surfaces non lambertiennes, le système à résoudre est non linéaire. On peut procéder par intersection entre les cartes de réflectance de chaque image, comme illustré sur la figure 1.14.

Ikeuchi [Ike91] utilise des cartes de réflectance correspondant à des illuminations diffuses. Coleman et Jain [CJ82] utilisent pour calibrer leur dispositif une sphère dont la réflectance est identique au matériau de la surface à reconstruire. Ils montrent que la reconstruction de surfaces texturées est possible pourvu que la réflectance soit la même à un facteur de proportionnalité près. Tagare et deFiguereido [Td91] ont montré que pour une grande classe de BRDF, trois images suffisent pour calculer les normales. Drew [Dre93] utilise une seule image couleur analysée suivant ses trois composantes spectrales, avec des éclairages de différentes couleurs et différentes orientations. Christensen et Shapiro [CS94] utilisent aussi la couleur, avec deux images et deux éclairages en lumière blanche.

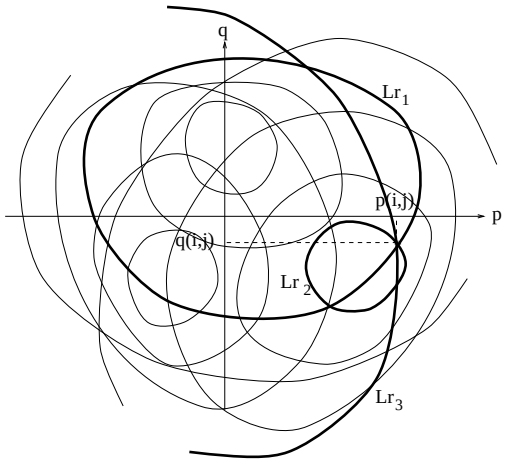


FIG. 1.14 - *intersection de trois cartes de réflectance. La normale locale à une surface est déterminée par l'intersection des isophotes correspondant aux différentes luminosités du point pour chaque éclairage.*

Les algorithmes de shading butent souvent sur des problèmes de spécularité. Rares sont les matériaux qui présentent une réflectance purement diffuse, et les reflets conduisent à des valeurs erronnées. L'utilisation d'un modèle comprenant à la fois la réflection spéculaire et la réflection diffuse est extrêmement délicate d'un point de vue pratique car la caméra a tendance à saturer sur les reflets. Si l'on ferme suffisamment le diaphragme pour éviter ce problème, les réflections diffuses conduisent à des niveaux trop faibles pour être exploitables.

Coleman [CJ82] propose d'utiliser quatre sources lumineuses pour éviter ce problème. Toutefois, les réflections spéculaires fournissent aussi des informations utiles et d'autres auteurs ont développé des algorithmes qui les utilisent explicitement [KSK88, LB92].

Application

La stéréovision photométrique, quand on peut l'employer, est largement préférable à l'image unique car elle permet de calculer la normale en chaque point indépendamment des voisins par la résolution d'un système d'équations. On peut même traiter des surfaces texturées si l'on connaît la BRDF en tout point de la surface.

Toutefois, la nécessité de disposer de plusieurs images prises du même point de vue sous des éclairages différents limite les applications à des objets assez petits pout être placés sur un banc de mesure. Le principal problème qui demeure est de connaître la BRDF de la surface. En effet, il n'existe pas de base de données digne de ce nom sur le sujet. On doit donc se limiter à des matières connues avec des états de surface bien contrôlés. C'est pourquoi certains travaux portent actuellement sur la mesure de la BRDF, comme nous allons le voir maintenant.

1.3 Mesure de la BRDF

Nous avons vu en 1.1.3 que la BRDF quantifie la redistribution de lumière dans les différentes directions de l'espace en fonction de la direction incidente et de la longueur

d'onde. C'est donc une fonction de cinq variables. Si la surface est isotrope, on peut aligner le repère d'expression de la BRDF sur la direction incidente, et la fonction n'a plus que quatre variables comme sur la figure (1.5): $f_r(\lambda, \theta_s, \theta_v, \phi_v)$. On peut la mesurer à l'aide d'un dispositif analogue à celui schématisé sur la figure (1.15).

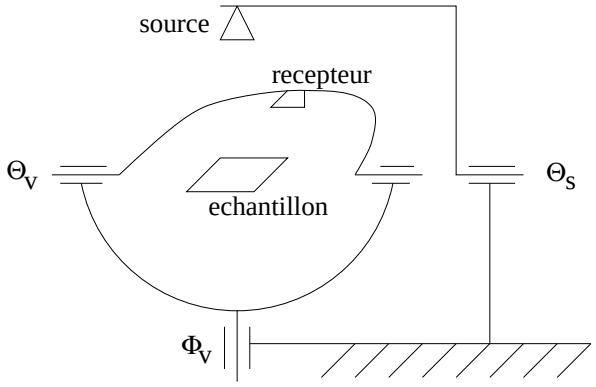


FIG. 1.15 - *Ce dispositif permet, en faisant varier les angles θ_s, ϕ_v et θ_v, de cartographier des BRDF de surfaces isotropes.*

Pour les surfaces anisotropes, il faut en plus pouvoir faire pivoter la source ou l'échantillon autour d'un axe vertical. Dans tous les cas, il faut se procurer un tel appareil, et la mesure de la BRDF est longue et fastidieuse.

Le paramètre longueur d'onde est le plus difficile à contrôler. En effet, on ne dispose pas de lampes dont on puisse facilement faire varier la longueur d'onde à puissance constante. On est généralement réduit à éclairer en lumière blanche et recueillir la lumière réfléchie, soit avec une caméra couleur, soit avec une caméra noir et blanc avec différents filtres devant l'objectif. On peut ainsi, faute de mieux, déterminer la BRDF pour les trois couleurs fondamentales, ce qui est raisonnable pour la synthèse d'images.

Un appareil presenté par Ward [War92] représenté sur la figure 1.16 permet une mesure beaucoup plus rapide car on peut saisir d'un seul coup l'ensemble des valeurs selon θ_v et ϕ_v.

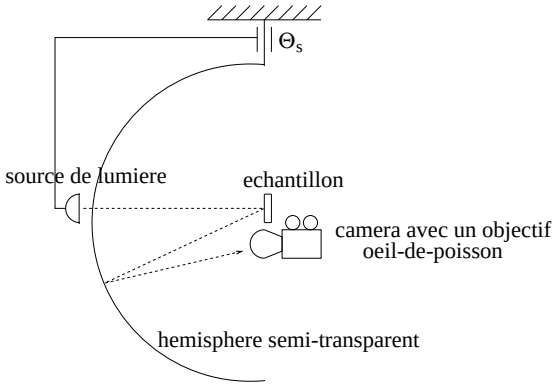


FIG. 1.16 - *La caméra recueille, par réflection sur l'hémissphère semi-transparent, la luminosité réfléchie dans toutes les directions.*

La lumière incidente pénètre dans l'hémisphère semi-transparent et vient frapper l'échantillon avec un angle réglable θ_s. La lumière réfléchie par l'échantillon se réfléchit

une fois encore à l'intérieur de la demi-sphère et est recueillie par l'objectif de la caméra. Connaissant la transformation $i, j \rightarrow \theta, \phi$, on peut cartographier d'un seul coup la BRDF dans toutes les directions réfléchies.

Ces dispositifs relativement complexes nécessitent de disposer d'un petit échantillon plan de la surface à analyser, ce qui n'est pas toujours possible.

Chapitre 2

Stéréovision photométrique rapide

Nous avons vu que les méthodes de stéréovision photométrique nécessitent, pour chaque normale à calculer, soit de résoudre un système d'équations non linéaires, soit de calculer l'intersection de cartes de réflectance préalablement établies. Dans ce chapitre, nous allons voir que pour une grande classe de BRDF, il est possible de calculer les normales par une recherche dichotomique, qui présente des avantages importants sur les méthodes habituellement employées, tant au niveau de la rapidité que de la précision.

Ce chapitre est organisé comme suit: nous allons tout d'abord choisir une famille de modèles de réflectance physiquement plausibles, suffisamment évolués pour représenter la réflectance de nombreuses surfaces, et assez simples pour être rapidement résolus. Puis nous présenterons la méthode dichotomique de résolution. Enfin nous discuterons des résultats.

2.1 Choix de modèle

2.1.1 Modèles classiques

En vision, il faut un modèle simple pour pouvoir inverser les équations. La plupart des auteurs ont utilisé le modèle lambertien qui pour une direction incidente unique donne:

$$L_r = \rho_s \, L_s \boldsymbol{n}.\boldsymbol{s}$$

Tagare et deFiguereido [Td93] proposent une simplification du modèle de Torrance-Sparrow, suivie d'une extension à plus de lobes, qu'ils appellent la Carte de Réflectance. L'intérêt du modèle de Torrance-Sparrow est d'être issu de considérations physiques sur le phénomène de réflection et transmission lumineuse. Dans sa version simplifiée, la carte de réflectance définit la luminosité provenant d'un point recevant un flux énergétique unitaire ($\phi = 1$) comme suit:

$$L_r(\boldsymbol{n}, \boldsymbol{s}, \boldsymbol{v}) = \rho_{fsc} \exp(-c^2 \arccos^2(\boldsymbol{n}, \boldsymbol{h})) + \rho_{norm} \boldsymbol{n}.\boldsymbol{s} + \rho_{bsc}$$

Cette formule a le mérite de correspondre assez bien aux mesures pour des angles d'incidence entre 0 et 40° de la normale. Elle appelle toutefois plusieurs remarques:

- Le terme ρ_{bsc} (bsc pour backscatter) est censé exprimer le mode de réflection directe, qui renvoie une partie de l'énergie autour de la direction incidente. A force de simplifications, ce terme n'est plus relié à la géométrie de la surface ni à celle de l'éclairage. Il reste

toutefois utile pour représenter l'illumination parasite toujours présente sur des images réelles.

- Le flux énergétique au point considéré vaut $\boldsymbol{n}.\boldsymbol{s}$, la BRDF correspondant à ce modèle est donc:

$$f_r = \rho_{norm} + \frac{\rho_{fsc} \exp(-c^2 \arccos^2(\boldsymbol{n}, \boldsymbol{h})) + \rho_{bsc}}{\boldsymbol{n}.\boldsymbol{s}}$$

et elle tend vers l'infini pour des incidences rasantes. Dans de telles conditions, la surface réémet plus d'énergie qu'elle n'en reçoit.

Ce modèle présente donc peu d'intéret pour le rendu, en particulier avec les méthodes de radiosité qui sont basées sur les transferts d'énergie. Les autres modèles issus de la physique [HTSG91, ON94, Sch93] sont beaucoup trop compliqués pour pouvoir inverser les équations.

2.1.2 Créer de nouveaux modèles pour la vision

Quelques règles à respecter

Nous avons vu en 1.1.3 que pour être physiquement plausible, les lois de réflection doivent s'exprimer à l'aide de la fonction f_r qui doit respecter certaines conditions:

- Pour respecter la conservation de l'énergie, le flux réémis doit être inférieur au flux incident, et donc la réflectance directionnelle-hémisphérique inférieure a 1:

$$\int_\Omega f_r(\boldsymbol{n}, \boldsymbol{s}, \boldsymbol{v})\ \boldsymbol{n}.\boldsymbol{s}\ d\omega_v \leq 1 \tag{2.1}$$

- La condition de symétrie de la BRDF entre les variables $\boldsymbol{s}$ et $\boldsymbol{v}$ impose de pouvoir définir la BRDF sous la forme:

$$f_r(\boldsymbol{n}, \boldsymbol{s}, \boldsymbol{v}) = f(\boldsymbol{n}, \boldsymbol{s}, \boldsymbol{v}) + f(\boldsymbol{n}, \boldsymbol{v}, \boldsymbol{s}) \tag{2.2}$$

  Tous les modèles physiquement plausibles peuvent s'exprimer sous cette forme.

- Les modèles anisotropes prennent en compte des directions privilégiées sur la surface du matériau. On peut alors utiliser la forme:

$$f_r(\boldsymbol{n}, \boldsymbol{t}, \boldsymbol{s}, \boldsymbol{v}) = f(\boldsymbol{n}, \boldsymbol{t}, \boldsymbol{s}, \boldsymbol{v}) + f(\boldsymbol{n}, \boldsymbol{t}, \boldsymbol{v}, \boldsymbol{s})$$

  où $\boldsymbol{t}$ représente une direction particulière de la surface.

- La réflection purement spéculaire rentre dans le cadre des fonctions symétriques, en s'exprimant sous la forme:

$$\begin{array}{l} f_{r,spec}(\boldsymbol{n}, \boldsymbol{s}, \boldsymbol{v}) = \delta(\boldsymbol{n}, \boldsymbol{h}).g(\boldsymbol{n}, \boldsymbol{s}) \\ g(\boldsymbol{n}, \boldsymbol{s}) \leq 1 \\ \delta(\boldsymbol{n}, \boldsymbol{h}) = 1\ si\ \boldsymbol{n} = \boldsymbol{h},\ 0\ sinon \end{array} \tag{2.3}$$

Comment créer des modèles physiquement plausibles

Quelles que soient les fonctions f et g, la BRDF:

$$f_r(\boldsymbol{n}, \boldsymbol{s}, \boldsymbol{v}) = f(\boldsymbol{n}, \boldsymbol{s}, \boldsymbol{v}) + f(\boldsymbol{n}, \boldsymbol{v}, \boldsymbol{s}) + \delta(\boldsymbol{n}, \boldsymbol{h}) g(\boldsymbol{n}, \boldsymbol{s})$$

respecte la condition de réciprocité. Il en va de même pour les fonctions qui font intervenir le vecteur $\boldsymbol{h}$ au lieu de $\boldsymbol{s}$ et $\boldsymbol{v}$. Pour respecter la conservation de l'énergie, il convient de multiplier la BRDF par un coefficient approprié afin que l'équation 2.1 soit respectée pour toute direction incidente.

2.1.3 Modèle choisi

Expression

On se trouve confronté à un dilemme: utiliser un modèle simple pour l'inverser facilement, mais trop limité pour modéliser la plupart des surfaces réelles, ou un modèle physique plus conforme à la réalité mais impossible à utiliser en pratique.

On utilisera donc une variante du modèle de Blinn, proposé par Lewis[Lew93] et défini par la BRDF:

$$f_r(\boldsymbol{n}, \boldsymbol{s}, \boldsymbol{v}) = \sum_i K_i (\boldsymbol{n}.\boldsymbol{h})^{n_i} \tag{2.4}$$

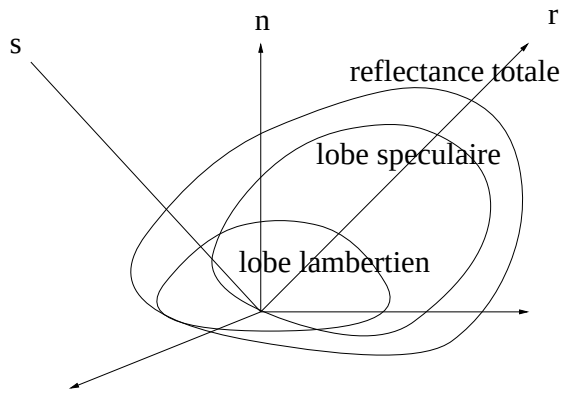


FIG. 2.1 - *Une BRDF définie par la somme de deux lobes.*

Les différents K_i et n_i sont des réels. Pour une direction incidente unique d'intensité L_i, l'intensité réfléchie vaut donc $L_r = L_i \boldsymbol{n}.\boldsymbol{s}. \sum_i K_i (\boldsymbol{n}.\boldsymbol{h})^{n_i}$ Cette formulation est suffisamment générale pour représenter la réflection lambertienne ($n_i = 0$) ou des réflections très concentrées (n_i grand). De plus, rien n'empêche *a priori* d'utiliser aussi des valeurs négatives pour certains lobes. Il faut veiller toutefois à ce que la BRDF reste positive sur tout son domaine de définition.

Performances et limitations

Ce modèle n'inclut pas de réflection spéculaire. La raison est que d'un point de vue technique on ne peut mesurer simultanément la réflection diffuse et la réflection spéculaire.

En effet, les reflets spéculaires sont beaucoup plus lumineux que les reflets diffus. Ce phénomène entraîne la saturation des capteurs de la caméra. Les surfaces devront donc être mesurées en utilisant plus de trois images pour assurer qu'en chaque point, au moins trois d'entre elles restent dans le domaine diffus. Toutefois, on peut représenter des réflections très concentrées en utilisant un lobe au coefficient n_i élevé dans l'équation 2.4.

La figure 2.2 représente les lobes de BRDF correspondant une direction incidente à trente degrés de l'horizontale ($\theta_s = 60^o$).

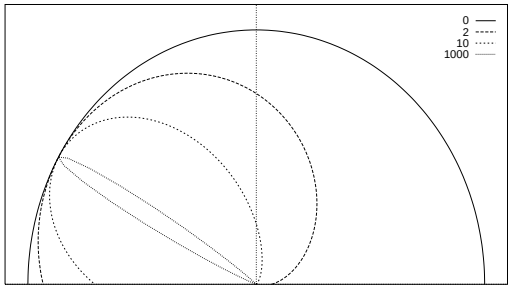


FIG. 2.2 - *Le modèle choisi permet de représenter aussi bien la réflection diffuse que la réflection quasi-spéculaire.*

Les lobes sont tous "centrés" sur la direction de réflection spéculaire, ce qui ne permet pas de représenter certaines réflectances atypiques comme les peintures réfléchissantes qui présentent un lobe dans la direction incidente.

Le fait de négliger la réflection spéculaire lors de la modélisation n'empêche pas d'en introduire lors du rendu, en utilisant une fonction arbitraire. Il suffit de vérifier que la réflectance directionnelle-hémisphérique reste inférieure à un. On peut d'ailleurs noter que pour le modèle proposé celle-ci décroit à mesure qu'on s'approche des incidences rasantes (voir les courbes dans [Lew93]). En compensant cette diminution de la réflection diffuse par un accroissement de la réflection spéculaire, on pourrait retrouver ce phénomène naturel bien connu: les surfaces tendent à devenir spéculaires quand on approche des incidences rasantes. Le lecteur curieux peut vérifier en utilisant n'importe quelle surface plane comme ce document par exemple, de la manière décrite par la figure (2.3).

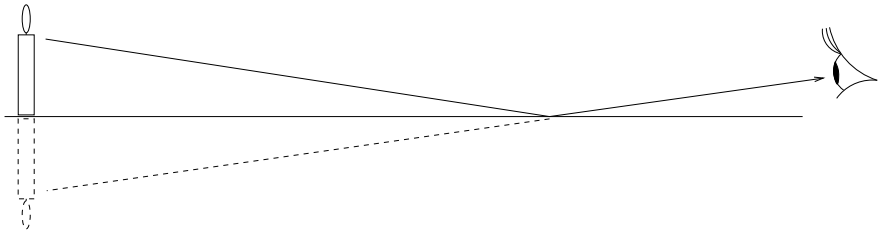


FIG. 2.3 - *Les surfaces tendent à devenir spéculaires pour les incidences rasantes.*

2.2 Reconstruction des normales

Nous allons chercher des conditions suffisantes pour que la reconstruction puisse s'effectuer, et voir dans quelles conditions on peut espérer une solution unique.

2.2.1 Hypothèses

- Par souci de simplicité, nous nous placerons dans le cas d'une caméra fixe et d'éclairages provenant successivement de différentes directions. L'hypothèse de surface lisse n'est pas nécessaire car la reconstruction s'effectue point par point.
- Pour chaque image, le champ lumineux est parfaitement caractérisé par la donnée d'une direction dans un repère lié à la caméra et d'une intensité lumineuse. C'est le cas quand la lumière provient d'une source ponctuelle située à l'infini.
- La fonction de réflectance est unimodale: il existe un vecteur $\boldsymbol{h} = \alpha\boldsymbol{s} + \beta\boldsymbol{v}$ tel que $R(\boldsymbol{n}, \boldsymbol{s}, \boldsymbol{v})$ soit strictement croissant en $\boldsymbol{n}.\boldsymbol{s}$ et $\boldsymbol{n}.\boldsymbol{h}$. Cette hypothèse est valable pour tous les modèles bilobes diffus, et peut même inclure la réflection spéculaire si le maximum diffus coincide avec cette direction.
- Les vecteurs interviennent dans $f(\boldsymbol{n}, \boldsymbol{s}, \boldsymbol{v})$ sous forme de produits scalaires: $f(\boldsymbol{n}, \boldsymbol{s}, \boldsymbol{v}) = f^*(\boldsymbol{n}.\boldsymbol{s}, \boldsymbol{n}.\boldsymbol{v})$. Le produit $\boldsymbol{s}.\boldsymbol{v}$ est considéré comme une donnée au vu des hypothèses précédentes. La luminosité réfléchie est donc une fonction $L(\boldsymbol{n}.\boldsymbol{s}, \boldsymbol{n}.\boldsymbol{v})$.

Dans ces conditions, il existe une fonction monotone

$$\begin{aligned} G :]0,\infty[\times]0,\infty[&\rightarrow]0,\infty[\\ L, \boldsymbol{n}.\boldsymbol{s} &\rightarrow \boldsymbol{n}.\boldsymbol{h} \end{aligned}$$

telle que pour une luminosité L, $G(L, \boldsymbol{n}.\boldsymbol{s}) = \boldsymbol{n}.\boldsymbol{h}$. Pour une valeur donnée de $\boldsymbol{n}.\boldsymbol{v}$, l'intersection de G avec la droite: $\boldsymbol{n}.\boldsymbol{s} = \|\boldsymbol{s} + \boldsymbol{v}\|\boldsymbol{n}.\boldsymbol{h} - \boldsymbol{n}.\boldsymbol{v}$ donne le couple$(\boldsymbol{n}.\boldsymbol{s}, \boldsymbol{n}.\boldsymbol{h})$ correspondant à L. Il existe donc une application

$$\begin{aligned} H :]0,\infty[\times]0,\infty[&\rightarrow]0,\infty[\times]0,\infty[\\ L, \boldsymbol{n}.\boldsymbol{v} &\rightarrow \boldsymbol{n}.\boldsymbol{s}, \boldsymbol{n}.\boldsymbol{h} \end{aligned}$$

avec $\boldsymbol{n}.\boldsymbol{s}, \boldsymbol{n}.\boldsymbol{h}, \boldsymbol{n}.\boldsymbol{v}$ satisfaisant l'équation de réflectance pour une source à l'infini (eq. 1.1) (voir figure 2.4). La valeur qui nous intéresse est $\boldsymbol{n}.\boldsymbol{s}$, qui est décroissant en $\boldsymbol{n}.\boldsymbol{v}$.

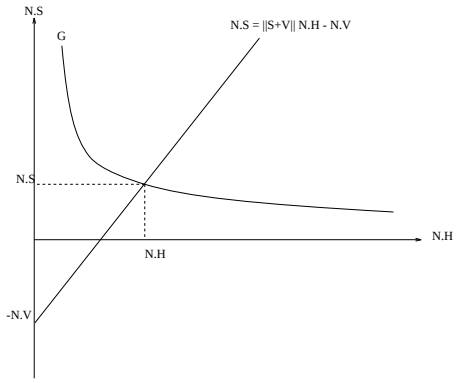


FIG. 2.4 - *La fonction G permet, connaissant les valeurs L_i et $\boldsymbol{n}.\boldsymbol{v}$, de déterminer $\boldsymbol{n}.\boldsymbol{s}_i$ et $\boldsymbol{n}.\boldsymbol{h}_i$*

Notons que $\boldsymbol{n}.\boldsymbol{v}$ ne dépend pas de la direction d'éclairage, et que pour trois sources le vecteur $\boldsymbol{v}$ s'exprime de façon unique par la relation: $\boldsymbol{v} = a_1\boldsymbol{s}_1 + a_2\boldsymbol{s}_2 + a_3\boldsymbol{s}_3$. Pour trois

valeurs de $\boldsymbol{n}.\boldsymbol{s}$ correspondant à trois éclairages différents, on recherche maintenant les valeurs de $\boldsymbol{n}.\boldsymbol{v}$ telle que

$$\boldsymbol{n}.\boldsymbol{v} = a_1\boldsymbol{n}.\boldsymbol{s}_1 + a_2\boldsymbol{n}.\boldsymbol{s}_2 + a_3\boldsymbol{n}.\boldsymbol{s}_3 \tag{2.5}$$

Nous appellerons la relation (2.5) *équation de compatibilité géométrique* entre la direction de vue et les directions d'éclairage. Ces valeurs existent si la fonction

$$S(\boldsymbol{n}.\boldsymbol{v}) = a_1\boldsymbol{n}.\boldsymbol{s}_1 + a_2\boldsymbol{n}.\boldsymbol{s}_2 + a_3\boldsymbol{n}.\boldsymbol{s}_3 - \boldsymbol{n}.\boldsymbol{v}$$

s'annule en certaines valeurs de $\boldsymbol{n}.\boldsymbol{v}$. La solution est unique si $S(\boldsymbol{n}.\boldsymbol{v})$ a un seul zéro. Si les a_i sont tous positifs alors $S(\boldsymbol{n}.\boldsymbol{v})$ est décroissante. La solution, si elle existe, est alors unique. Or S est continue et:

$$\lim_{\boldsymbol{n}.\boldsymbol{s}\to 0} S(\boldsymbol{n}.\boldsymbol{v}) = \infty$$
$$\lim_{\boldsymbol{n}.\boldsymbol{v}\to \infty} S(\boldsymbol{n}.\boldsymbol{v}) = -\infty$$

L'équation $S(\boldsymbol{n}.\boldsymbol{v}) = 0$ permet donc de trouver $\boldsymbol{n}.\boldsymbol{v}$ par dichotomie.

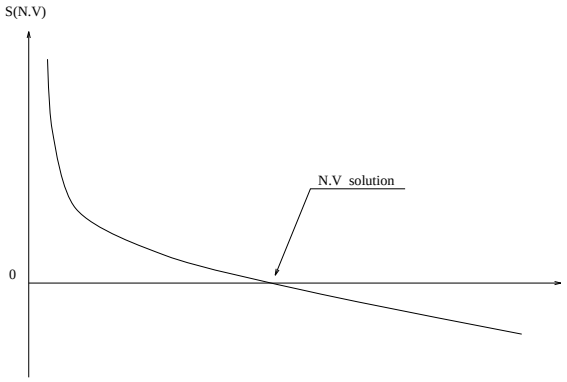


FIG. 2.5 - *La monotonie de S permet une recherche de* $\boldsymbol{n}.\boldsymbol{v}$ *par dichotomie*

Conditions d'unicité pour les fonctions unimodales

Pour tout triplet de valeurs, le système d'équations a donc une solution unique sur l'intervalle $]0, \infty[$ si:

- $R(\boldsymbol{s}, \boldsymbol{n}, \boldsymbol{v})$ est croissante en $\boldsymbol{n}.\boldsymbol{s}$ et $\boldsymbol{n}.\boldsymbol{h}$

- $\boldsymbol{v} = a_1\boldsymbol{n}.\boldsymbol{s}_1 + a_2\boldsymbol{n}.\boldsymbol{s}_2 + a_3\boldsymbol{n}.\boldsymbol{s}_3$ avec tous les a_i positifs. Cette configuration correspond au cas où $\boldsymbol{v}$ est contenu dans le trièdre positif formé par $\boldsymbol{s}_1$, $\boldsymbol{s}_2$ et $\boldsymbol{s}_3$. Ces deux conditions sont suffisantes, mais elles ne sont pas forcément nécessaires. D'autre part, pour un triplet aberrant de valeurs, le produit scalaire $\boldsymbol{n}.\boldsymbol{v}$ trouvé pourra être supérieur à 1, ce qui correspond à $\|\boldsymbol{n}\| > 1$. En effet, on n'a nulle part imposé la condition de normalité du vecteur $\boldsymbol{n}$ recherché.

Intérêt de l'équation de compatibilité géométrique

La relation 2.5 sur les vecteurs source et points de vue permet de définir la fonction $S(\boldsymbol{n}.\boldsymbol{v})$, et le problème de la reconstruction se réduit alors à trouver la ou les racines

de cette fonction. On peut donc utiliser des méthodes beaucoup plus rapides et efficaces qu'en résolvant le problème sous sa forme classique:

$$\begin{cases} R(\boldsymbol{s}_1, \boldsymbol{n}, \boldsymbol{v}) = R_1 \\ R(\boldsymbol{s}_2, \boldsymbol{n}, \boldsymbol{v}) = R_2 \\ R(\boldsymbol{s}_3, \boldsymbol{n}, \boldsymbol{v}) = R_3 \end{cases} \tag{2.6}$$

par minimisation d'erreur sur les deux angles qui définissent $\boldsymbol{n}$. En effet,

- l'espace de recherche est réduit une dimension,
- et on n'a pas de problème de minima locaux car la fonction est strictement décroissante.

Les avantages de cette méthode seront chiffrés dans l'exemple d'application traité plus loin.

2.2.2 Application à un modèle bilobe

Les essais ont été réalisés en utilisant un modèle bilobe, c'est-à-dire que la BRDF est la somme de deux termes. Le premier correspond à une réflection lambertienne, l'autre à une réflection spéculaire diffuse dont on peut moduler la concentration autour de la direction spéculaire. La surface est éclairée successivement par trois sources correspondant à des directions non coplanaires, et vue par une caméra fixe. Par la suite et par souci de simplicité, on supposera tous ces dispositifs à l'infini, hypothèse non indispensable si en tout point on connait les paramètres géométriques locaux de l'éclairage.
La radiance en tout point est exprimée par le modèle choisi, réduit à un lobe Lambertien et un seul lobe spéculaire diffus:

$$L_{ri}(\boldsymbol{n}, W_i, \boldsymbol{s}_i, \boldsymbol{h}_i) = W_i \boldsymbol{n}.\boldsymbol{s}_i (K_d + K_s (\boldsymbol{n}.\boldsymbol{h}_i)^n) \tag{2.7}$$

La quantité $W_i \boldsymbol{n}.\boldsymbol{s}_i$ correspond à l'énergie reçue localement par unité de surface. On peut utiliser l'équation sous la forme:

$$\boldsymbol{n}.\boldsymbol{v} = \|\boldsymbol{v} + \boldsymbol{s}_i\| \sqrt[n]{\frac{\frac{L_{ri}}{W \boldsymbol{n}.\boldsymbol{s}_i} - K_d}{K_s}} - \boldsymbol{n}.\boldsymbol{s}_i \tag{2.8}$$

Cette fonction est monotone en $\boldsymbol{n}.\boldsymbol{s}_i$, on peut donc facilement l'inverser par dichotomie. Notons que cette façon de procéder est une variante de la méthode générale, due au fait que grâce à la forme analytique du modele, on utilise :$L_i, \boldsymbol{n}.\boldsymbol{v} \rightarrow \boldsymbol{n}.\boldsymbol{s}$.

2.2.3 Évaluation de la qualité du résultat

Erreur objective

Quand les normales de la surface sont connues, comme dans le cas d'une surface simulée numériquement, ou très particulière (sphère, cylindre...), nous pouvons utiliser la fonction d'erreur:

$$E = \sqrt{\frac{1}{N} \sum_{i,j} (\boldsymbol{n}_{i,j} - \boldsymbol{n}*)^2}$$

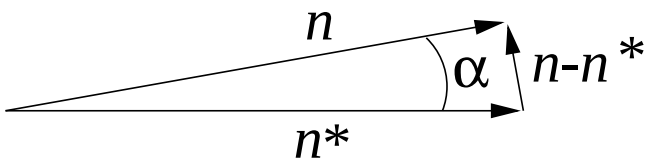


FIG. 2.6 - $\|\boldsymbol{n}\| \simeq 1$, $\|\boldsymbol{n} - \boldsymbol{n}*\| \simeq \alpha$

avec $\boldsymbol{n}_{i,j}$ la normale reconstituée, $\boldsymbol{n}*$ la normale d'origine et N le nombre de termes de la somme. Cette erreur est équivalente à un angle moyen de déviation comme le montre la figure (2.6).

Dans la suite, nous appellerons cette valeur l'*erreur objective* E_o, car elle est une mesure de l'écart entre les "vraies" normales et les normales reconstruites.

Erreur évaluée

Dans le cas d'une application concrète, les normales de la surface à reconstituer sont inconnues, on ne peut donc pas calculer l'erreur objective. Une bonne méthode de reconstruction doit donc être capable de se faire elle-même une idée de la qualité des résultats, c'est ce que nous appellerons l'*erreur évaluée* E_e.

Celle-ci dépend de la méthode utilisée, nous la définirons par la suite. Un exemple d'erreur évaluée est la fonction qu'on cherche à minimiser pour résoudre un système non linéaire.

2.2.4 Comparaison avec une méthode classique

Les essais ont été réalisés sur une surface de synthèse (figure 2.7) composée de 1600 polygones, dont on a simulé un comportement conforme au modèle de Blinn-Lewis, soumise à des éclairages fictifs parfaitement définis. Les coefficients utilisés pour la réflectance sont: $K_d = 0.3, K_s = 0.6, n = 3.0$

Nous appellerons méthode des racines la méthode dichotomique présentée dans ce document, et méthode de minimisation la résolution classique de système non linéaire. Les recherches de racines sont effectuées par un algorithme de Newton-Raphson unidimensionnel [PTVF92] utilisant les dérivées premières. L'erreur évaluée porte sur la norme des vecteurs reconstruits. En effet, aucune équation utilisée n'implique qu'elle doit valoir 1; les résultats nous diront si l'expression suivante est un critère d'erreur pertinent. Soit N le nombre de points traités,

$$E_e = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{n}_i^2 - 1)^2$$

Une telle reconstruction de 500×500 normales nécessite environ vingt minutes de calcul sur une station SGI INDY R4000 100 MHz, pour une précision telle que $E_o < 0.000001$.

On effectue la comparaison avec une méthode qui consiste à optimiser les deux angles (θ, ϕ) qui définissent l'orientation de la normale pour minimiser l'erreur évaluée. L'expression de cette erreur est:

$$E_e = \frac{1}{N}\sum_{i=1}^{N}\sum_{vue=1}^{3}(R_i^* - R_i(\theta_i, \phi_i, \boldsymbol{s}_{vue}, \boldsymbol{v}))^2$$

bruit		0	0.01	0.02	0.05
Minimisation	E_o	0.000	0.009	0.018	0.046
Racines	E_o	0.000	0.009	0.019	0.048
	E_e	0.000	0.004	0.008	0.019

TAB. 2.1 - *Erreur de reconstruction en fonction du bruitage des données.*

où le signe * désigne une valeur mesurée. La recherche du minimum dans l'espace (θ, ϕ) est effectuée par un algorithme de Powell[PTVF92].

Notons que cette dernière méthode consiste expressément à réduire l'erreur évaluée, alors que la première ne l'utilise que pour tirer un bilan du travail effectué.

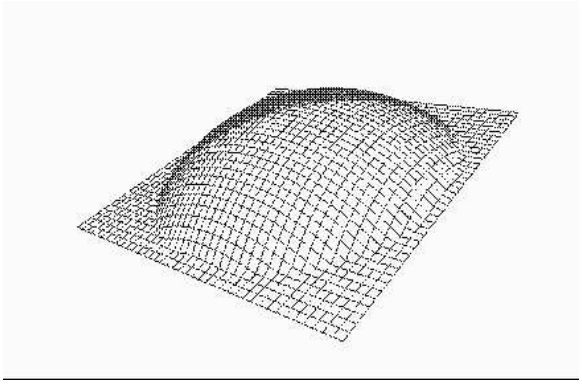


FIG. 2.7 - *Vue filaire de la surface-test*

Le tableau 2.1 montre les résultats obtenus pour différents bruitages. Le bruit appliqué est gaussien, l'écart-type imposé est proportionnel à la valeur à bruiter, les taux sont ceux figurant en première ligne.

Deux constats:

- les deux méthodes présentent la même robustesse au bruit, qui est bonne puisque l'erreur lui est proportionnelle;
- l'erreur évaluée est pertinente puisqu'elle présente le même caractère proportionnel.

Les figures 2.8 à 2.17 montrent le comportement de l'erreur objective et de l'erreur évaluée pour des coefficients de réflectance légèrement faux. Les essais ont été faits pour deux niveaux de bruit différents, l'un nul, et l'autre de 0.02 (écart-type de la Gaussienne), qui correspond au bruit habituel d'une caméra classiquement utilisée en vision.

Commentaires

On peut dégager plusieurs enseignements de ce comparatif:

– L'erreur objective est beaucoup plus faible par la méthode des racines pour des coefficients légèrement faux (fig. 2.8, 2.9, 2.10). On peut supposer, en termes imagés,

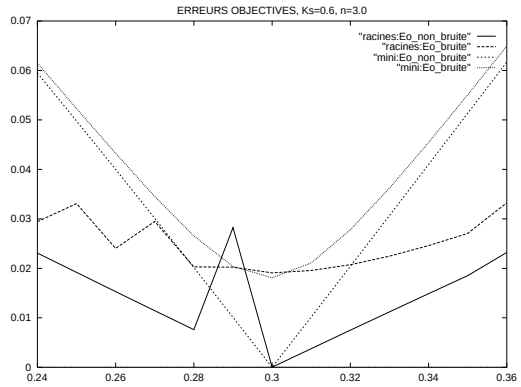


FIG. 2.8 - *Erreur objective en fonction de K_d*

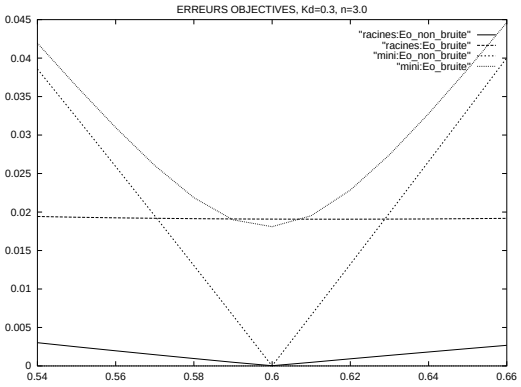


FIG. 2.9 - *Erreur objective en fonction de K_s*

qu'une partie de l'incohérence est "évacuée"dans la norme du vecteur reconstruit;

- L'erreur évaluée semble lisse et ne pas présenter de minima locaux indésirables (fig. 2.11, 2.12, 2.13, 2.14, 2.15, 2.16). Bien qu'elle traduise mal l'erreur objective, elle est utile pour évaluer la justesse des coefficients du modèle.

- Les minima locaux de l'erreur évaluée sont aussi bien, ou mieux placés, par les racines que par la minimisation. La figure 2.17 en montre un exemple. La méthode des racines semble donc bien se prêter à une optimisation itérative des paramètres du modèle;

- Le temps de calcul des normales est constant par la méthode des racines, et variable par minimisation suivant la justesse des coefficients utilisés. Il est entre trois et dix fois inférieur par la méthode des racines. Un autre algorithme d'optimisation donnerait peut-être de meilleurs résultats. Toutefois, la recherche portera dans tous les cas sur deux valeurs, au lieu d'une seule pour la méthode dichotomique.

2.2.5 Retrouver les sources d'éclairage

Les directions et intensités d'éclairage dans un repère dont un axe est confondu avec celui de la caméra peuvent être retrouvées encore plus facilement. Si l'on connaît les normales en trois points frappés par les mêmes éclairages, il suffit d'exprimer L_{i1}/L_{i2} et

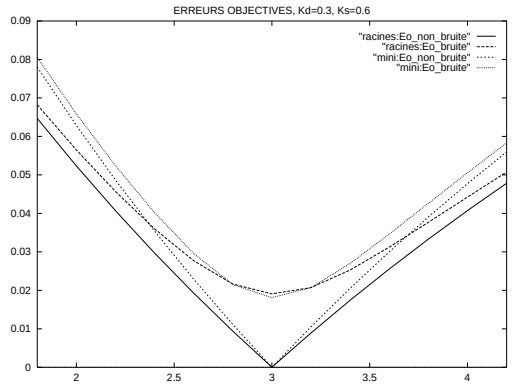


FIG. 2.10 - *Erreur objective en fonction de n*

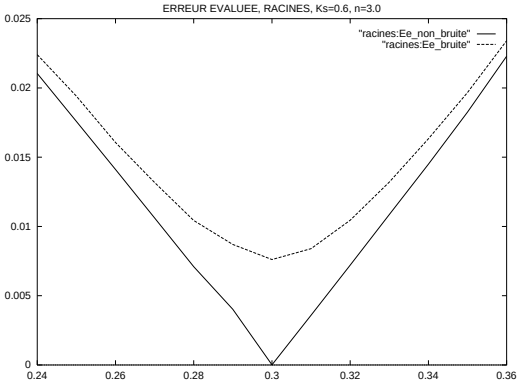


FIG. 2.11 - *Erreur évaluée par la méthode des racines en fonction de K_d*

L_{i1}/L_{i3} à l'aide de l'équation (2.7) pour déterminer le vecteur $\boldsymbol{s}_i$, puis d'en déduire W_i. En pratique, on peut utiliser plus de points et une méthode de minimisation d'erreur. Utiliser une sphère dont la BRDF est connue précisément permet de connaître en tout point l'ensemble des paramètres requis: normale et réflectance. Le temps nécessaire pour optimiser sur cent points avec une INDY est d'environ une seconde. Cela peut permettre de calibrer très facilement le dispositif, après quoi on peut mesurer des surfaces inconnues.

2.2.6 Application à une caméra mobile

La réciprocité de la BRDF permet d'adapter très simplement l'algorithme à un éclairage fixe avec caméra mobile, sur le bras d'un robot par exemple. La recherche portera alors sur $\boldsymbol{n}.\boldsymbol{s}$, et la relation (2.5) devient:

$$b_1\boldsymbol{n}.\boldsymbol{v}_1 + b_2\boldsymbol{n}.\boldsymbol{v}_2 + b_3\boldsymbol{n}.\boldsymbol{v}_3 = \boldsymbol{n}.\boldsymbol{s}$$

Ceci suppose tout de même que la correspondance entre les images soit effectuée.

2.3 Conclusion

La méthode des cartes de réflectance nécessite une calibration du banc de mesure qui rend son application concrète extrêmement problématique. Pour s'affranchir de ce

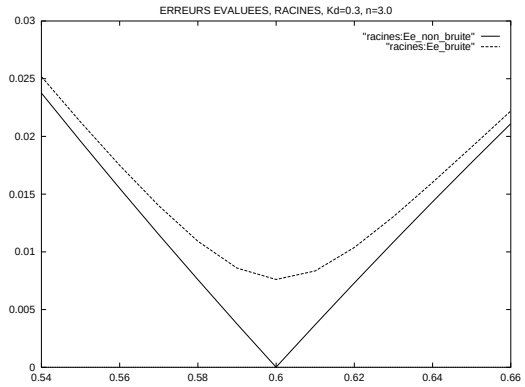


FIG. 2.12 - *Erreur évaluée par la méthode des racines en fonction de K_s*

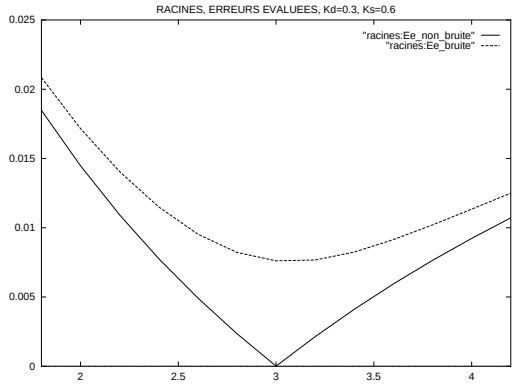


FIG. 2.13 - *Erreur évaluée par la méthode des racines en fonction de n*

problème, il faut être capable de résoudre un système d'équations non linéaires qui traduisent le phénomène physique de réflection lumineuse. Ces équations doivent satisfaire des critères de plausibilité physique pour être applicables dans des situations variées.

Pour résoudre le système d'équations, il est très utile d'introduire l'équation de compatibilité géométrique (2.5) car elle permet de réduire l'espace de recherche à une seule dimension. Pour une certaine classe très répandue de modèles de réflectance, il existe des conditions suffisantes d'unicité de la solution, qui sont facilement vérifiables.

La robustesse au bruit sur les valeurs mesurées est à peu près la même, mais les erreurs de reconstruction sont très nettement inférieures en cas d'imprécision sur les paramètres du modèle.

La fonction d'auto-évaluation de l'erreur est aussi plus efficace, car son minimum est mieux placé.

Cette nouvelle méthode, par sa rapidité et sa précision accrues et sa fonction d'évaluation plus précise, laisse entrevoir la possibilité d'une optimisation itérative efficace des paramètres d'un modèle complexe et physiquement plausible, prochaine étape à franchir avant de pouvoir utiliser concrètement la stéréovision photométrique non Lambertienne.

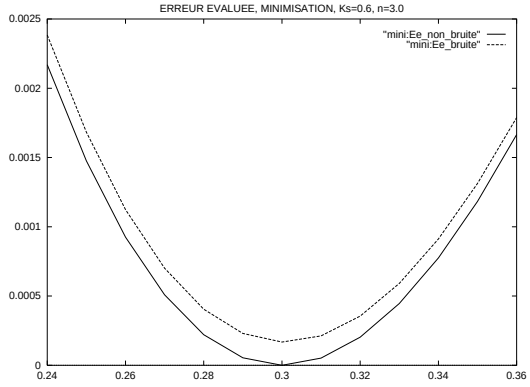


FIG. 2.14 - *Erreur évaluée par la méthode de minimisation en fonction de K_d*

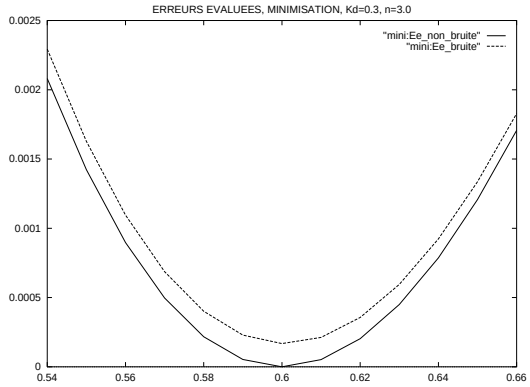


FIG. 2.15 - *Erreur évaluée par la méthode de minimisation en fonction de K_s*

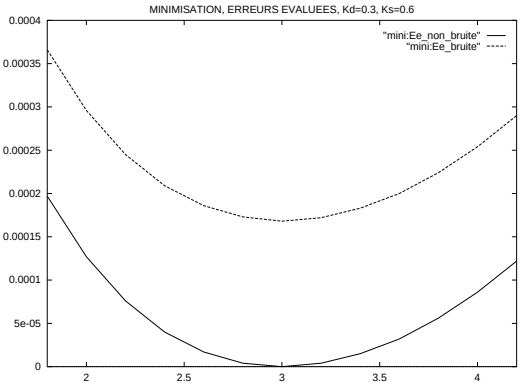


FIG. 2.16 - *Erreur évaluée par la méthode de minimisation en fonction de n*

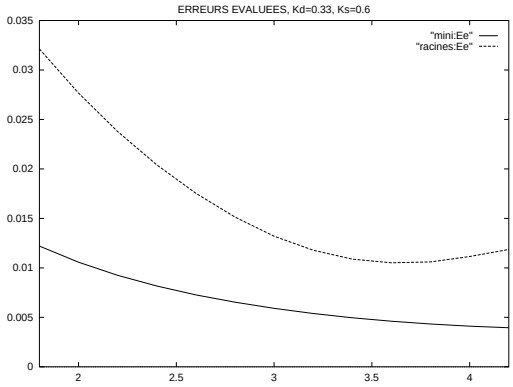


FIG. 2.17 - *Comparaison d'erreurs évaluées*

Chapitre 3

Mesure de la BRDF

3.1 Reconstruction du modèle

3.1.1 Détermination des paramètres

Avec un dispositif calibré et une surface dont on connaît la géométrie, il est possible de trouver le modèle de Blinn-Lewis à deux lobes s'approchant le mieux du comportement de la matière dont la surface est constituée. En effet, en posant l'équation (2.8) sous la forme:

$$\frac{L_{ri}}{\phi \boldsymbol{n}.\boldsymbol{s}_i} - K_d = K_s(\boldsymbol{n}.\boldsymbol{h}_i)^n$$

elle devient dans l'espace logarithmique:

$$log(\frac{L_{ri}}{\phi \boldsymbol{n}.\boldsymbol{s}_i} - K_d) = \log K_s + n \log \boldsymbol{n}.\boldsymbol{h}_i \tag{3.1}$$

A l'ensemble des normales correspond donc un nuage de points qui est une droite si l'on a choisi la bonne valeur de K_d. Le meilleur jeu de paramètres correspond à la droite dont la somme des carrés des écarts est minimale. On a alors les valeurs de K_s et de n qui sont déduites de l'ordonnée à l'origine et de la pente de cette droite (Figure 3.1). Le temps de calcul est de quelques millisecondes pour une centaine de points.

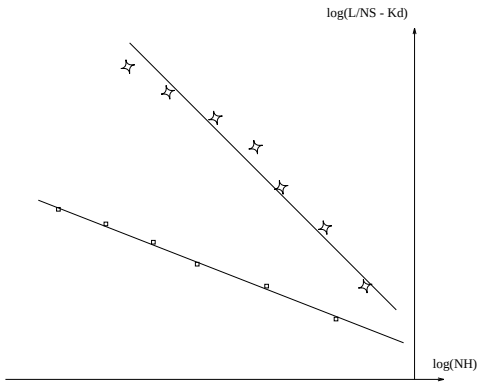


FIG. 3.1 - *La bonne valeur de K_d produit la meilleure droite*

3.1.2 Évaluation du résultat

Une fois la meilleure droite trouvée, l'erreur quadratique moyenne quantifie l'imprécision du résultat. Les sources d'erreur sont:

- La mesure de radiance: bruit, gamma-correction;
- l'éclairage non parallèle ou mal connu;
- les normales locales mal évaluées: surface imprécise, distorsions optiques;
- l'inadéquation du modèle à la réflectance réelle de la matière.

3.1.3 Expérimentation

Les essais ont été menés en retrouvant les meilleurs paramètres pour des surfaces de synthèse dont on avait simulé différents modèles de réflectance. Les BRDF reconstituées sont présentées par "tranches" correspondant à φ_s et φ_v constants, en faisant varier θ_v (voir Figure 3.2).

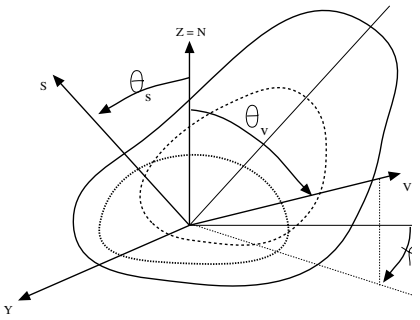


FIG. 3.2 - *Les BRDF seront représentées à φ_s et φ_v constants*

Les figures suivantes (3.3,3.4,3.5) représentent les BRDF retrouvées à partir d'une BRDF de Blinn-Lewis bruitée à différents niveaux. Le bruit est Gaussien avec un écart-type de respectivement 0.01, 0.02, 0.05 fois le niveau mesuré.

bruit	K_d	K_s	n	Erreur
0	0.300	0.600	3.003	0.0001
0.01	0.280	0.621	2.870	0.010
0.02	0.266	0.638	3.800	0.019
0.05	0.260	0.660	2.939	0.048

Il est intéressant de constater que même si les coefficients retrouvés diffèrent de ceux de départ, l'erreur qu'ils engendrent lors de la reconstruction est pratiquement égale à celle obtenue avec les bons coefficients.

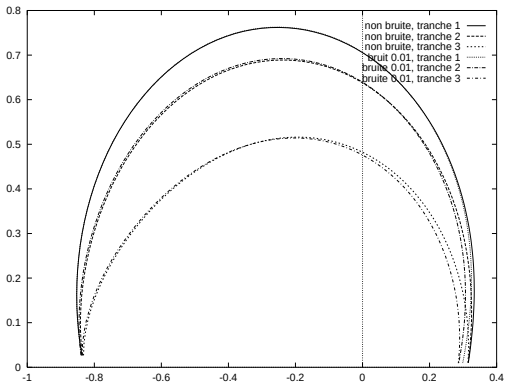


FIG. 3.3 - $\sigma_{bruit} = 0.01$

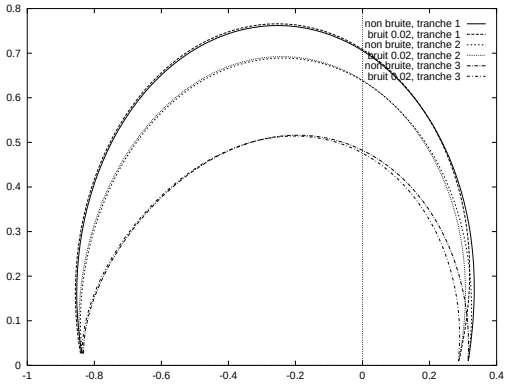


FIG. 3.4 - $\sigma_{bruit} = 0.02$

3.2 Détermination simultanée du modèle et de la géométrie

3.2.1 Optimisation itérative des paramètres

Sur fond d'images recueillies par les caméras, il existe une dualité entre réflectance et géométrie, car l'une peut être déduite à partir de l'autre. Ne connaissant ni l'une ni l'autre, on sera amené à exploiter cette dualité.

L'approche choisie est un système bouclé qui reçoit en entrée les intensités des points de l'image, fournit en sortie un modèle et un ensemble de normales, et qui tend à se stabiliser dans l'état où l'ensemble {images, éclairage, modèle, normales} forme un tout cohérent.

Le point crucial est la mise en oeuvre d'une fonction d'erreur efficace. L'optimisation de l'erreur quadratique entre intensités mesurées et recalculées s'avérant extrêmement lente, et peu fiable à cause de la multiplicité des minima locaux, on peut se tourner vers l'utilisation de l'écart-type sur la régression parabolique précédemment décrite.

L'initialisation est effectuée en donnant un jeu de paramètres du modèle, et les normales sont calculées d'après ceux-ci. La reconstruction s'effectue selon la méthode décrite précédemment, la synthèse selon la formule 2.7, et la mise à jour du modèle en utilisant une méthode classique d'optimisation. A chaque itération, on optimise successivement K_d

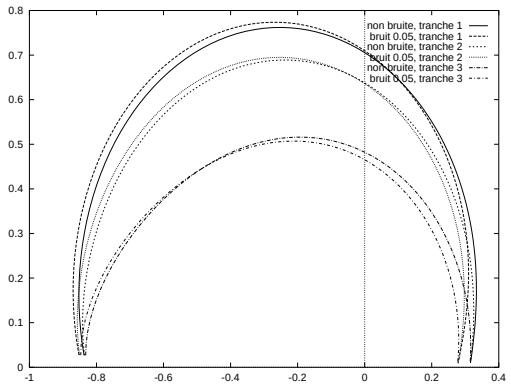


FIG. 3.5 - $\sigma_{bruit} = 0.05$

fonction Erreur_Kd(NouveauKd)
Kd=NouveauKd
reconstruire les normales
créer le nuage de points
calculer la droite des moindres carrés
retourner sigma

Algorithme Principal
tant que *sigma* > *tolerance*:
optimiser Erreur_Kd()
réactualiser Ks
optimiser Erreur_n()

FIG. 3.6 - *Reconstruction simultanée de la géométrie et de la réflectance.*

et n. La fonction Erreur_n ne figure pas dans la description de l'algorithme car elle suit exactement le même principe que pour K_d. K_s est réévalué après chaque optimisation de K_d par $K_s = L_{max} - K_d$. La valeur finale de l'erreur indique la qualité du résultat obtenu.

3.2.2 Expérimentation

Les essais ont été menés de la même façon que pour la reconstruction des normales. Un échantillon d'une centaine de points représentatifs suffit pour obtenir une bonne précision, en un temps acceptable. Pour un lobe spéculaire peu marqué ($n < 10$), les valeurs d'initialisation des paramètres n'ont pratiquement pas d'influence sur la convergence, et le temps nécessaire pour déterminer la BRDF avec une erreur relative inférieure à 0.001 varie de une à dix minutes, suivant que le modèle est plus ou moins proche d'un comportement Lambertien. La reconstruction des normales va de pair, avec une précision équivalente.

Sur des données bruitées les résultats sont bons puisque les erreurs sont à peine supé-

rieures à celles qu'on obtenait par reconstruction en connaissant les coefficients.

bruit	K_d	K_s	n	Erreur
0	0.298	0.602	2.993	0.0006
0.01	0.299	0.601	3.045	0.012
0.02	0.307	0.595	3.141	0.024
0.05	0.314	0.595	3.446	0.059

Pour de grandes valeurs de n, la convergence est très sensible aux valeurs initiales et il faut envisager une autre méthode d'optimisation.

3.3 Essai sur des images réelles

Les méthode de modélisation simultanée de la géométrie et de la réflectance a été testée sur les images présentées sur la figure (3.7). La surface est un modèle en pâte à modeler qui représente vaguement une forme de visage humain.

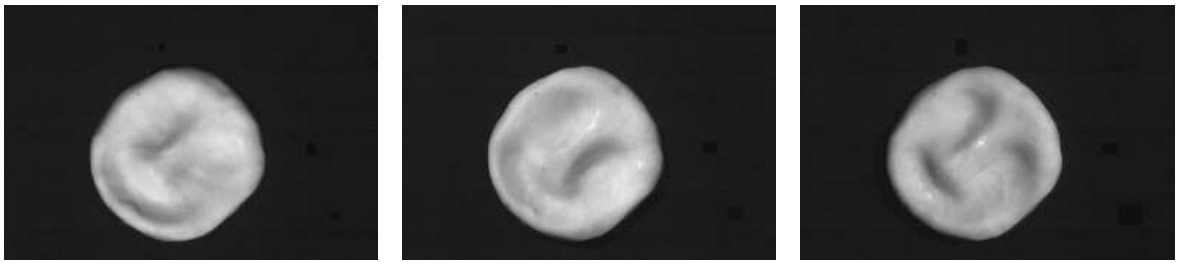


FIG. 3.7 - *Trois images utilisées pour reconstruire une surface,*

Le matériel d'éclairage est composé de deux lampadaires halogènes de jardin et d'un lampadaire à ampoule incandescente, posés à deux mètres de l'objet. D'ores et déjà, deux hypothèses simplificatrices ne sont donc pas vérifiées:

- les rayons incidents ne sont pas parallèles mais proviennent de directions variant de 0 à 5 degrés d'angle autour de la direction moyenne;
- l'influence des longueurs d'onde peut perturber les résultats car une ampoule incandescente n'a pas le même spectre lumineux qu'une ampoule halogène.

Ce dispositif inadéquat fut le seul disponible pour des raisons matérielles. De plus, la matière n'étant pas parfaitement matte, de petites zones de reflets amenant la caméra à saturation sont présentes sur les images. Il est de surcroît difficile de décider si certains ombrages proviennent de la lumière rasante ou d'un auto-ombrage de la surface. La surface étant faiblement concave, les phénomènes d'inter-réflection ont été ignorés. On a donc choisi de ne retenir que les points dont les trois luminosités (une pour chaque image) étaient compris entre un seuil mini et un seuil maxi.

Les résultats de l'algorithme de modélisation automatique de réflectance furent décevants, avec une lente convergence vers des valeurs qui dépendaient très fortement des seuils maxi et mini retenus pour les intensités lumineuses. Ce n'est pas très étonnant vu

les mauvaises conditions expérimentales, mais indique une réelle sensibilité de l'algorithme aux problèmes pratiques.

On a ensuite effectué différentes reconstructions en utilisant des paramètres arbitraires pour la BRDF. La surprise est venue du fait que les reconstructions dépendent qualitativement très peu de la BRDF utilisée. Le relief apparaît plus ou moins prononcé, mais un simple changement d'échelle permet d'obtenir une surface quasiment identique pour toutes les BRDF "raisonnables" testées. La figure (3.8) présente la surface reconstruite.

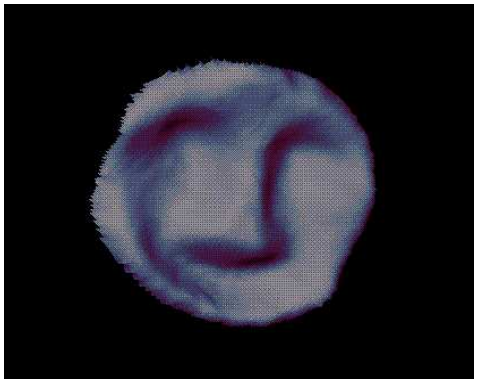


FIG. 3.8 - *surface reconstruite*

3.4 Bilan

- On a donné un exemple de méthode de reconstruction non Lambertienne avec un modèle physiquement plausible. Il reste à vérifier expérimentalement que ce modèle approche correctement le comportement d'un grand nombre de matières.
- Cet algorithme peut être appliqué à une caméra en mouvement avec source d'éclairage unique si l'on résoud la correspondance entre les images.
- On a montré un moyen de déterminer très simplement les meilleurs paramètres de Blinn-Lewis pour une surface à géométrie connue.
- Une boucle analyse/synthèse permet, dans des conditions idéales, de reconstituer simultanément la géométrie et la réflectance d'une surface non Lambertienne.
- Ce système fournit, parallèlement aux résultats, une quantification de leur fiabilité.
- Il n'utilise que des images comme données d'entrée, sans connaissance préalable de la géométrie ni de la réflectance des surfaces.
- L'application à des images réelles n'a pas permis de modéliser la BRDF de façon convaincante.
- La surface reconstruite dépend finalement assez peu de la BRDF utilisée.

Conclusion

Un nouvel algorithme pour la stéréovision photométrique a été proposé. Il permet, pour une vaste classe de BRDF physiquement plausibles, de reconstruire les normales locales à l'aide d'une recherche dichotomique. Cette approche est plus robuste au bruit gaussien qu'une résolution plus classique, et semble plus rapide.

Une nouvelle approche pour la reconstruction simultanée de la géométrie et de la réflectance des surfaces a été proposée. Elle donne de bons résultats sur des images de synthèse mais n'a pas fait ses preuves sur des images réelles. De meilleures conditions expérimentales pourraient peut-être permettre d'obtenir des résultats plus satisfaisants.

Dans tous les cas, la modélisation automatique de surfaces à base d'images repose sur des hypothèses très restrictives. Le travail et les résultats présentés dans ce document montrent qu'il est dejà difficile d'obtenir de bons résultats pour des surfaces mattes et unies. La supériorité des méthodes utilisant plusieurs images sur celles n'en utilisant qu'une suggère l'utilité d'une certaine redondance des données. Des modeleurs automatiques plus généraux et performants pourraient voir le jour en traitant des informations provenant de sources multiples (caméras, télémètres, bases de données).

Annexe A

Principaux symboles physiques utilisés

On notera s le nombre de solides multiplié par six, a le nombre de contraintes cinématiques scalaires. Le tableau se poursuit à la page suivante.

symbole	*dimension*	*apparition*	*commentaires*
q	1	1.1.1	variable cinématique
$\boldsymbol{c}$	a	1.1.1	valeurs des contraintes cinématiques
$\boldsymbol{J}$	a, s	1.1.1	jacobienne, influence des variables cinématiques sur les valeurs contraintes (dépend du contexte)
m	1	1.2	masse d'un solide
$\boldsymbol{a}$	3	1.2	accélération d'un solide
$\boldsymbol{I}_g$	$3, 3$	1.2	matrice d'inertie d'un solide en son centre de masse
$\dot{\boldsymbol{\omega}}$	3	1.2	accélération angulaire absolue d'un solide
$\dot{x}$	1		dérivée par rapport au temps de la variable x
$\ddot{x}$	1		dérivée seconde par rapport au temps de la variable x
$(\dot{\boldsymbol{u}})_i$	3	2.1.3	dérivée par rapport au temps du vecteur u considéré fixe dans le repère i
$\boldsymbol{f}$	f	1.2.2,5.1.2	forces aux contraintes
$\boldsymbol{a}$	a	1.2.2	mouvements dans les directions contraintes
$\boldsymbol{D}$	a, a	1.2.2,5.1.4	matrice dynamique, mouvements aux contraintes en fonction des forces aux contraintes $\boldsymbol{Df} = \boldsymbol{a}$

symbole	*dimension*	*apparition*	*commentaires*
$\boldsymbol{V}_i^P$	3	2.1.1	vitesse en P du solide i
$\boldsymbol{\Gamma}_i^P$	3	2.1.1	accélération en P du solide i
$\boldsymbol{\Omega}_{ij}$	3	2.1.1	vitesse angulaire du solide i par rapport à j
$\boldsymbol{V}_{ij}^P$	3	2.1.2	vitesse en P du solide i par rapport à j
$\boldsymbol{v}_{ij}^P$	6	2.1.3	vitesse généralisée en P du solide i par rapport à j
$\boldsymbol{\nabla}$	$6,6$	2.1.3	opérateur de changement de point d'expression des vitesses généralisées
$\boldsymbol{u}^\wedge$	$3,3$	2.1.3	matrice adjointe du vecteur $\boldsymbol{u}$
$\boldsymbol{\gamma}_{ij}^P$	6	2.1.3	accélération d'entraînement généralisée en P du solide i par rapport à j
$\boldsymbol{\delta}_{ij}^u$	6	2.1.3	additif de changement de point d'expression relatif aux accélérations centripètes
$\boldsymbol{\gamma}_{ijk}^P$	6	2.1.3	additif de composition des accélérations relatif aux accélérations de Coriolis
$\boldsymbol{f}^P$	6	2.2.2	force généralisée exprimée au point P
$\boldsymbol{\Delta}$	$6,6$	2.2.2	opérateur de changement de point d'expression des forces généralisées
$\boldsymbol{f}_{\bar{i}i}^P$	6	2.2.2	force généralisée exercée par l'extérieur de i sur i, exprimée au point P
$\boldsymbol{M}_i^P$	$6,6$	2.2.3	matrice de masse généralisée du solide i au point P
$\boldsymbol{d}_{ij}^p$	6	2.2.5	quantité de mouvement généralisée du solide i dans son mouvement par rapport à j, exprimée au point P
$\ddot{\boldsymbol{d}}_{ij}^p$	6	2.2.5	quantité d'accélération généralisée du solide i dans son mouvement par rapport à j, exprimée au point P
$\boldsymbol{P}_{ji}$	$4,4$	3.2	matrice de passage de j à i
A_{jk}		3.5.1	articulation entre les solides j et k
L_{jk}	3	3.5.1	centre de la liaison A_{jk}
q_{jkl}	1	3.5.1	l-ième coordonnée relative de l'articulation A_{jk}
$\boldsymbol{a}_{jkl}$	6	3.5.1	axe généralisé de la l-ième direction indépendante de la liaison A_{jk}
a? b: c		3.5.1	expression valant b si a est vrai, c sinon
$\boldsymbol{F}_i^P$	$6,a$	3.5.2	influence des forces articulaires sur le solide i, exprimée au point P
$\hat{\boldsymbol{\omega}}_{ij0}$	6	3.5.3	accélération relative induite par la rotation du repère de dérivation
$\boldsymbol{e}_{ij}$	6	5.1.1	mouvement relatif au point de liaison entre i et j
$\boldsymbol{e}$	a	5.1.4	vecteur rassemblant tous les mouvements relatifs aux points de contraintes dans les directions des contraintes
$\boldsymbol{e}[ijk]$	6	5.1.1	mouvement relatif au point de liaison entre i et j, projeté sur $\boldsymbol{a}_{ijk}$

Annexe B

Algorithmes de dynamique

Nous fournissons ici des pseudocodes concernant l'implémentation de la méthode itérative rapide: la résolution linéaire des contraintes acycliques de Baraff, puis le produit matriciel restreint aux chaînes fermées.

B.1 Résolution acyclique

Le graphe cinématique considéré ici est défini conformément à l'exemple de la figure (5.16). Seule la partie acyclique est prise en compte. De plus, la racine n'est pas le repère absolu. Si la structure est reliée au repère absolu par une articulation comportant au moins une contrainte, la racine est cette articulation. Dans le cas contraire, la racine est le plus "haut" solide du graphe. L'implémentation nécessite la définition d'une classe générique regroupant solides et articulations. Nous appelons cette classe *noeud* et sa strucure est donnée dans le pseudocode B.1.

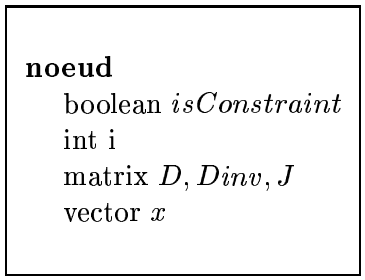


FIG. B.1 - *classe noeud*

À chaque noeud du graphe cinématique correspond un elément de la classe *noeud*, repéré individuellement par son identificateur i. Le booléen $isConstraint$ est vrai pour les articulations et faux pour les solides. Les articulations ont leurs matrices D et $Dinv$ carrées de dimension le nombre de contraintes cinématiques de l'articulation. Nous supposons disposer d'une fonction **parent(i)** qui renvoie l'identificateur du noeud immédiatement supérieur au noeud i dans le graphe cinématique, et 0 si i est la racine du graphe. Nous disposons aussi d'une fonction **child(i)** qui renvoie la liste de tous les éléments immédiatement inférieurs à i dans le graphe.

Nous définissons deux listes de noeuds appelées $Forward$ et $Backward$, l'une étant l'inverse de l'autre. La boucle "pour $n \in Forward$" indique un parcours des feuilles vers la

racine, alors que "pour $n \in Backward$" indique un parcours de la racine vers les feuilles.

La procédure **linearFactor** (fig. B.2) effectue un précalcul dont les résultats permettent d'utiliser à volonté la procédure **linearSolve**. Au début de **linearSolve** (fig. B.3), les erreurs cinématiques sont chargées dans les vecteurs x des articulations. Après exécution de la procédure, les vecteurs x contiennent, pour les solides, les déplacements, et pour les articulations, les forces articulaires qui annulent l'erreur initiale. À chaque noeud est associée une matrice $\boldsymbol{J}_n$ définie comme suit. Soit une articulation A_{ij}, qui relie dans le graphe acyclique le solide i au solide supérieur j. La matrice $\boldsymbol{J}_n$ de l'articulation A_{ij} est $\boldsymbol{J}_{ijj}$, conformément aux notations de la formule (5.5), et la matrice $\boldsymbol{J}_n$ du solide i est $\boldsymbol{J}_{iji}$.

```
procedure linearFactor
pour n ∈ Forward
   if n.isConstraint
      n.D = 0
   else
      n.D = M_{n.i}
   if parent(n) ≠ 0
      int p = n.i, q = parent(i)
      n.J = n.isConstraint? J_n:J_n^t
pour n ∈ Forward
   pour c ∈ child(n)
      n.D -= (c.J)^t (c.D)(c.J)
   n.Dinv = n.D^{-1}
   if parent(n) ≠ 0
      n.J = (n.Dinv)(n.J)
```

FIG. B.2 - *procedure linearFactor*

```
procedure linearSolve
pour n ∈ Forward
   n.x = n.isConstraint ? e_n : 0
   pour c ∈ child(n)
      n.x -= (c.J)^t (c.x)
pour n ∈ Backward
   n.x = (n.Dinv)(n.x)
   if parent(n) ≠ 0
      n.x -= (n.J)(parent(n).x)
```

FIG. B.3 - *procedure linearSolve*

Notons pour finir que cette méthode ne traite pas que les arbres mais aussi les forêts. On peut lui confier la résolution de plusieurs graphes acycliques indépendants sans rien

avoir à modifier dans l'algorithme.

B.2 Produit restreint

Le produit restreint aux chaînes fermées utilisé dans la résolution mixte itérative est présenté sur le pseudocode de la figure (B.6). Pour un vecteur $\boldsymbol{f}$ représentant un ensemble de forces articulaires dans les fermetures de boucles, la procedure **produitRestreint**($\boldsymbol{f}$, $\boldsymbol{x}$) calcule le vecteur $\boldsymbol{x}$ qui rassemble les mouvements résultants selon les directions des contraintes de fermeture. La résolution linéaire acyclique est utilisée afin de maintenir satisfaites les contraintes acycliques. Au départ, nous transférons les forces dans les vecteurs x des articulations de fermetures chaînes au moyen de la procédure **lireVecteur** décrite sur la figure (B.5). À la fin, nous transférons dans le vecteur $\boldsymbol{f}$ les déplacements calculés, au moyen de la procédure **ecrireVecteur**. Ces deux procédures nécessitent une numérotation préalable des liaisons effectuée par la procédure **numeroteFermetures** fournie sur la figure (B.4). La liste $s.Closed$ désigne l'ensemble des articulations de fermetures de boucle sur le solide s. L'entier $n.nbConstraints$ désigne le nombre de contraintes cinématiques contenues dans la liaison n. Nous supposons diposer d'une liste d'articulations appelée *Closed* qui contient toutes les liaisons de fermetures de chaînes. Nous avons aussi une liste *Solid* qui contient tous les solides. L'ordre de ces listes n'a pas d'importance.

```
procedure numeroteFermetures
int nb = 0
pour s ∈ Solid
   pour n ∈ s.Closed
      n.num = nb
      nb = nb + n.nbConstraints
```

FIG. B.4 - *procedure numeroteFermetures*

Pour une articulation habituellement notée A_{ij} où j est le père et i le fils, nous notons J_p la matrice $\boldsymbol{J}_{ijj}$ et J_f la matrice $\boldsymbol{J}_{iji}$.

```
procedure lireVecteur( vecteur u )
pour n ∈ Closed
   pour i = 0; i < n.nbConstraints; i++
      n.x[i] = u[n.num + i]


procedure ecrireVecteur( vecteur u )
pour n ∈ Closed
   pour i = 0; i < n.nbConstraints; i++
      u[n.num + i] = n.x[i]
```

FIG. B.5 - *procedures lireVecteur et ecrireVecteur*

```
procedure produitRestreint( vecteur f, vecteur x)
lireVecteur(x)
pour n ∈ Solid
   n.x = 0
pour n ∈ Closed
   n.pere.x += (n.J_p^t)(n.x)
   n.fils.x += (n.J_f^t)(n.x)
pour n ∈ Solid
   n.x = (n.M^-1)(n.x)
pour n ∈ Closed
   n.x = (n.J_p)(n.pere.x) + (n.J_f)(n.fils.x)
pour n ∈ Forward
   n.x = n.isConstraint ? (n.J_p)(n.pere.x) + (n.J_f)(n.fils.x) : 0
   pour c ∈ child(n)
      n.x -= (c.J)^t(c.x)
pour n ∈ Backward
   n.x = (n.Dinv)(n.x)
   if parent(n) ≠ 0
      n.x -= (n.J)(parent(n).x)
pour n ∈ Closed
   n.x -= (n.J_p)(n.pere.x) + (n.J_f)(n.fils.x)
ecrireVecteur(f)
```

FIG. B.6 - *procedure produitRestreint*

Annexe C

Coordonnées homogènes

On utilisera le formalisme des coordonnées homogènes. Un vecteur tridimensionnel est représenté par un quadruplet de valeurs, les trois premières correspondant aux habituelles projections du vecteur sur trois directions orthogonales. La quatrième valeur vaut un ou zéro selon que le vecteur représente un point ou un bipoint.

Les points $\boldsymbol{A}$ et $\boldsymbol{B}$ sur la figure C.1 sont représentés respectivement par les quadruplets $(x_A, y_A, z_A, 1)^t$ et $(x_B, y_b, z_B, 1)^t$. Le vecteur (ou bipoint) $\boldsymbol{AB}$ est alors naturellement représenté par la différence des deux quadruplets, soit $(x_B - x_A, y_B - y_A, z_B - z_A, 0)^t$.

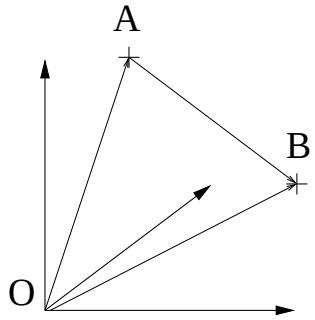


FIG. C.1 - *Deux points, un bipoint*

Cette convention surprenante permet une formalisation simple des changements de repères. En coordonnées cartésiennes, pour un changement de base défini par la matrice 3×3 $\boldsymbol{R}$, le nouveau vecteur se calculerait par la formule: $\boldsymbol{u'} = \boldsymbol{R} \cdot \boldsymbol{u}$. Un changement de repère, qui correspond simultanément à un changement de base et un changement d'origine, s'exprime en coordonnées homogènes par: $\boldsymbol{u'} = \boldsymbol{C} \cdot \boldsymbol{u}$, où $\boldsymbol{C}$ est une matrice 4×4 contenant à la fois le changement de base et le changement d'origine:

$$\boldsymbol{C} = \begin{pmatrix} & & & x \\ & \boldsymbol{R} & & y \\ & & & z \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{C.1}$$

Le quadruplet $(x, y, z, 1)$ représente les coordonnées de l'origine de l'ancien repère dans le nouveau, comme représenté sur la figure C.2. Ceci explique l'intéret du quatrième paramètre du quadruplet: pour changer de repère un point, qui représente une position, il convient de tenir compte du décalge d'origine des repères, d'où la valeur 1 du quatrième

paramètre, qui sert à ajouter le décalage. Au contraire, un vecteur représentant un écart entre deux points n'a pas à subir le déclage d'origine lors d'un changement de repère, on fixe donc le quatrième paramètre à zéro.

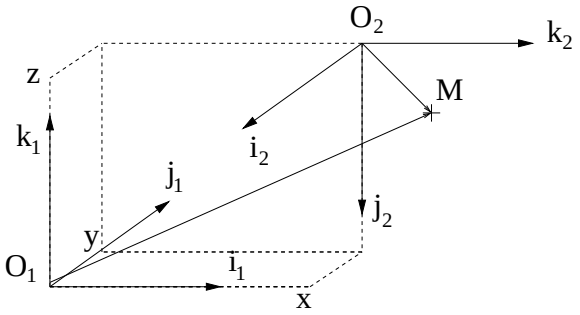


FIG. C.2 - *Un point* $\boldsymbol{M}$ *dans deux repères différents*

Examinons l'exemple représenté sur la figure C.2. Soit le point $\boldsymbol{M}$ défini par le vecteur $\boldsymbol{M}_1(x_{M1}, y_{M1}, z_{M1}, 1)^t$ dans le repère 1 et par le vecteur $\boldsymbol{M}_2(x_{M2}, y_{M2}, z_{M2}, 1)^t$ dans le repère 2. La matrice de changement de repère s'écrit:

$$\begin{pmatrix} 0 & 0 & 1 & x \\ -1 & 0 & 0 & y \\ 0 & -1 & 0 & z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Les trois premières colonnes de la matrice $\boldsymbol{C}$ correspondent respectivement aux projections de $\boldsymbol{i_2}$,$\boldsymbol{j_2}$ et $\boldsymbol{k_2}$ dans le repère 1. Le point M projeté dans le repère 1 vaut:

$$\boldsymbol{M}_1 = \boldsymbol{C} \cdot \boldsymbol{M}_2$$

tandis que le vecteur $\boldsymbol{O_2M}$ projeté dans 1 vaut:

$$(\boldsymbol{O_2M})_1 = \boldsymbol{C} \cdot (x_{M2}, y_{M2}, z_{M2}, 0)^t$$

Annexe D

Centres et axes de liaisons

Une articulation entre deux objets sert à restreindre leur mouvement relatif. Les contraintes s'expriment alors préférentiellement en un certain point, qu'on appelle généralement *centre de liaison*, ou sur une certaine droite de l'espace affine, généralement appelée*axe de liaison*. Par exemple, le centre d'une liaison rotule, réalisable par deux sphères concentriques, est le centre commun de ces deux sphères. C'est un point cinématiquement privilégié car la vitesse relative y reste constamment nulle. Une liaison pivot, réalisable par deux cylindres de révolution coaxiaux, possède un axe, qui n'est autre que l'axe commun des deux cylindres. La vitesse relative est nulle en tout point de cet axe.

Toutefois, certaines liaisons semblent ne pas comporter d'axe. C'est le cas de la liaison "appui plan", dont un exemple est un cube glissant sur un marbre, comme représenté sur la figure D.1. Aucune contrainte n'oblige le cube à tourner autour d'un axe précis.

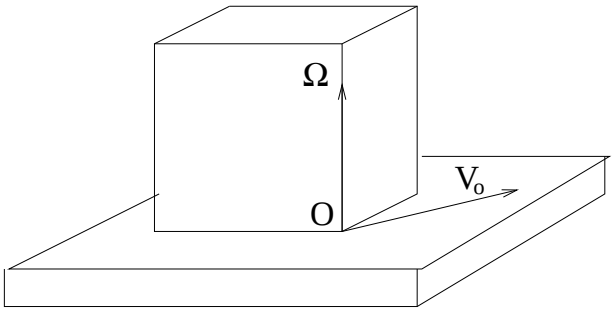


FIG. D.1 - *Une liaison sans centre fixe*

Une étude plus poussée des mouvements relatifs instantannés est donc nécessaire avant de pouvoir utiliser les centres ou axes de liaisons comme outils privilégiés de cinématique du solide. Examinons donc la notion d'axe instantané de rotation.

D.1 Axe instantané de rotation en 2D

En deux dimensions, les solides bougent dans un plan et le mouvement peut être décomposé en deux translations dans le plan et une rotation selon un axe perpendiculaire au plan. Connaissant ces paramètres à l'origine d'un repère, on peut connaître le mouvement en tout point de ce repère en utilisant la relation (2.7). On voit sur la figure D.2 que dans

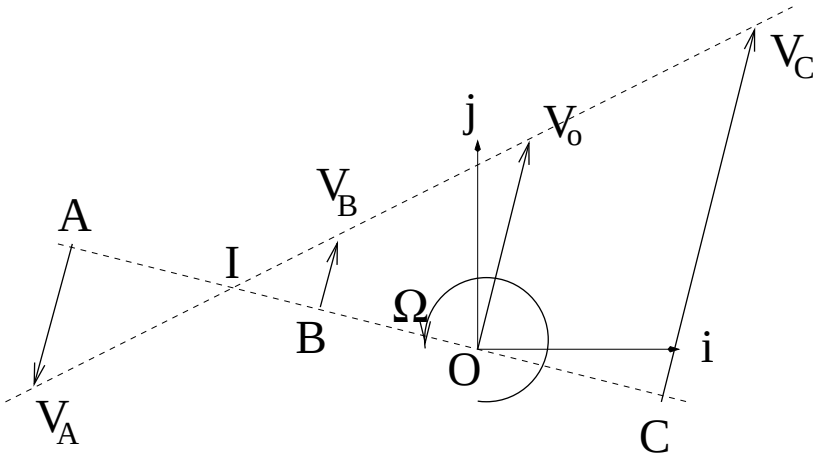


FIG. D.2 - *Centre instantané de rotation*

le cas où la rotation est non nulle, il existe un point I dont la vitesse est nulle à l'instant considéré. Ce point est appelé *centre instantané de rotation* du mouvement solide défini par V_0 et Ω.

Ce mouvement est donc équivalent à une rotation de centre I. Toutefois, dans le cas général, le point I peut varier au cours du temps. Dans le cas d'une rotation nulle, le centre instantané de rotation n'est pas défini, mais le mouvement est alors une translation et la vitesse est la même en tout point. Nous pouvons donc conclure que tout mouvement plan, à un instant donné, est assimilable à une rotation autour d'un point du plan. Dans le cas d'une translation, ce point est rejeté à l'infini.

D.2 Cas tridimensionnel

Pour décomposer un mouvement en trois dimensions, il suffit de se placer dans un plan orthogonal à la direction de rotation. Le problème est alors similaire au cas bidimensionnel, avec en plus une composante de la vitesse V_0 colinéaire à la direction de rotation. Le centre instantané de rotation devient l'axe instantané de rotation. La vitesse des points de cet axe peut ne pas être nulle, mais simplement colinéaire à l'axe. On parle alors de mouvement de vissage. Tout mouvement solide tridimensionnel peut donc être assimilé, à un instant donné, à un vissage autour d'une droite appelée*axe instantané de rotation.* Dans le cas d'une translation pure, cet axe est rejeté à l'infini.

Annexe E

Exemple de modélisation

```
    ———————————————————————————
————————- Constantes du Bonhomme ————————-
———————————————————————————

set Taille 1.8
set Masse 80

set CouleurR 0.8
set CouleurV 0.8
set CouleurB 0.6

set inclinaison -0.1

set Ltete [expr 0.10*$Taille]
set Mtete [expr 0.04*$Masse]
set Rtete [expr sqrt($Mtete/($Densite*$Ltete))]

set Lcorps [expr 0.287*$Taille]
set Mcorps [expr 0.621*$Masse]
set Rcorps [expr sqrt($Mcorps/($Densite*$Lcorps))]

⋮


———————————————————————————
———————— Declaration des Solides ————————
———————————————————————————

set tete [FaArticulatedGenerator new]
$tete setMass $Mtete
$tete setCenterOfMass  0 0 0
$tete setInertia  1 1 1 0 0 0
```

```
set brasG [FaArticulatedGenerator new]
$brasG setMass $Mbras
$brasG setCenterOfMass  0 0 0
$brasG setInertia  1 1 1 0 0 0

set brasD [FaArticulatedGenerator new]
$brasD setMass $Mbras
$brasD setCenterOfMass  0 0 0
$brasD setInertia  1 1 1 0 0 0

⋮

_____________________________
_____- Declaration des Articulations _____-
_____________________________

set Cou [FaArtStill new]
$Cou setFather $corps
$Cou setSon $tete
$Cou setFatherPoint "0 [expr -$Lcorps/2] 0"
$Cou setSonPoint "0 [expr $Ltete/2] 0"

set EpauleG [FaArtPivotDriven new]
$EpauleG setFather $corps
$EpauleG setSon $brasG
$EpauleG setFatherPoint "$Dbras [expr -$Lcorps/2] 0"
$EpauleG setSonPoint "0 [expr -$Lbras/2] 0"

⋮ set Crochet [FaArtPlaneTranslation new]
$Crochet setFather $Ground
$Crochet setSon $corps
$Crochet setFatherPoint  0.0 0.0 0.0
$Crochet setSonPoint  0 0 0
$agroup moveDOF $Crochet "0 0 0 $inclinaison 0 0"

set HancheG [FaArtPivotDriven new]
$HancheG setFather $corps
$HancheG setSon $cuisseG
$HancheG setFatherAxis  1 0 0
$HancheG setSonAxis  1 0 0
$HancheG setFatherPoint "-$Dcuisse [expr $Lcorps/2] 0"
$HancheG setSonPoint "0 [expr -$Lcuisse/2] 0"

⋮
```

```
————————————————————————————
———- FaSurfaces pour les contacts ———-
————————————————————————————

set talonG [FaSurfSphere new]
$talonG setColor "$CouleurR $CouleurV $CouleurB"
$talonG setTransparency 0
$talonG setRadius $Rpied
$talonG setCenter "0 [expr -$Lpied/2] 0"
$talonG setArtGen $piedG

set talonD [FaSurfSphere new]
$talonD setColor "$CouleurR $CouleurV $CouleurB"
$talonD setTransparency 0
$talonD setRadius $Rpied
$talonD setCenter "0 [expr -$Lpied/2] 0"
$talonD setArtGen $piedD

set planteG [FaSurfSphere new]
$planteG setColor "$CouleurR $CouleurV $CouleurB"
$planteG setTransparency 0
$planteG setRadius $Rpied
$planteG setCenter "0 [expr $ArtPied] 0"
$planteG setArtGen $piedG
$planteG setFrictionCoeff $frictionCoef

⋮

set plan [FaSurfRectangle new]
$plan setColor  0 1 0
$plan setPlan  0 0 0 0 -1 0
$plan setDim  2 2
$plan setFrictionCoeff 1
$plan setArtGen $sol

set contactD [FaArtSurfaceContact new]
$contactD setMother $sol
$contactD setDaughter $talonD
$contactD setFatherPoint  0 0 0.51
$contactD setFatherAxis  0 -1 0
$contactD setSonPoint  0 0 0
$contactD setSonAxis  0 0 0
```

Bibliographie

[ADH89] B. Arnaldi, G. Dumont, and G. Hégron. Dynamics and unification of animation control. *The Visual Computer*, 5:22–31, 1989.

[AG85] W. W. Armstrong and M. Green. The dynamics of articulated rigid bodies for purposes of animation. In M. Wein and E. M. Kidd, editors, *Graphics Interface '85 Proceedings*, pages 407–415. Canadian Inf. Process. Soc., 1985.

[APRB90] T. Alameldin, M. Palis, S. Rajasekaran, and N. Badler. On the complexity of computing reachable workspaces for redundant manipulators. In *OE/Boston'90 Symposium on Advances in Intelligent Systems. Intelligent Robots and Computer Vision IX: Algorithms and Complexity*, SPIE, Bellingham, Wash., 1990.

[Arm79] W. W. Armstrong. Recursive solution to the equations of motions of an n-link manipulator. In *Proceedings of the 5th World Congress on Theory of Machines and Mechanisms*, volume 2, pages 1343–1346. Canadian Inf. Process. Soc., 1979.

[Bar89] David Baraff. Analytical methods for dynamic simulation of non-penetrating rigid bodies. In Jeffrey Lane, editor, *Computer Graphics (SIGGRAPH '89 Proceedings)*, volume 23, pages 223–232, July 1989.

[Bar90] David Baraff. Curved surfaces and coherence for non-penetrating rigid body simulation. In Forest Baskett, editor, *Computer Graphics (SIGGRAPH '90 Proceedings)*, volume 24, pages 19–28, August 1990.

[Bar91] David Baraff. Coping with friction for non-penetrating rigid body simulation. In Thomas W. Sederberg, editor, *Computer Graphics (SIGGRAPH '91 Proceedings)*, volume 25, pages 31–40, July 1991.

[Bar94] David Baraff. Fast contact force computation for nonpenetrating rigid bodies. In Andrew Glassner, editor, *Proceedings of SIGGRAPH '94 (Orlando, Florida, July 24–29, 1994)*, Computer Graphics Proceedings, Annual Conference Series, pages 23–34. ACM SIGGRAPH, ACM Press, July 1994. ISBN 0-89791-667-0.

[Bar96] David Baraff. Linear-time dynamics using lagrange multipliers. In *SIGGRAPH 96 Conference Proceedings*, Computer Graphics Proceedings, Annual Conference Series, pages 137–146. ACM SIGGRAPH, Addison Wesley, August 1996. ISBN 0-201-94800-1.

[BB88] Ronen Barzel and Alan H. Barr. A modeling system based on dynamic constraints. In John Dill, editor, *Computer Graphics (SIGGRAPH '88 Proceedings)*, volume 22, pages 179–188, August 1988.

[BBG+94] Norman I. Badler, Ramamani Bindiganavale, John P. Granieri, Susanna Wei, and Xinmin Zhao. Posture interpolation with collision avoidance. In IEEE Computer Society Press, editor, *Proceedings of Computer Animation'94*, pages 13–20, May 1994.

[BC89] Armin Bruderlin and Thomas W. Calvert. Goal-directed, dynamic animation of human walking. In Jeffrey Lane, editor, *Computer Graphics (SIGGRAPH '89 Proceedings)*, volume 23, pages 233–242, July 1989.

[Bli77] James F. Blinn. Models of light reflection for computer synthesized pictures. In James George, editor, *Computer Graphics (SIGGRAPH '77 Proceedings)*, volume 11, pages 192–198, July 1977.

[BMT95] Ronan Boulic, Ramon Mas, and Daniel Thalmann. Position control of the center of mass for articulated figures in multiple support. In Dimitri Terzopoulos and Daniel Thalmann, editors, *Computer Animation and Simulation '95*, pages 130–143. Eurographics, Springer-Verlag, September 1995. ISBN 3-211-82738-2.

[BN88] Lynne Shapiro Brotman and Arun N. Netravali. Motion interpolation by optimal control. In John Dill, editor, *Computer Graphics (SIGGRAPH '88 Proceedings)*, volume 22, pages 309–315, August 1988.

[BPW93] Norman I. Badler, Cary B. Phillips, and Bonnie Lynn Webber. *Simulating Humans: Computer Graphics Animation and Control*. Oxford University Press, New York, 1993. ISBN 0-19-507359-2.

[Bru82] A.R. Bruss. The eikonal equation: Some results applicable to computer vision. *J. Physics*, 23(5):890–896, 1982.

[BTC94] Armin Bruderlin, Chor Guan Teo, and Tom Calvert. Procedural movement for articulated figure animation. *Computers and Graphics*, 18(4):453–462, 1994. ISSN 0097-8493.

[BTT90] Ronan Boulic, Nadia Magnenat Thalmann, and Daniel Thalmann. A global human walking model with real-time kinematic personification. *The Visual Computer*, 6(6):344–358, December 1990.

[BVBS87] A.H. Barr, B. Von Herzen, R. Barzel, and S. Snyder. Computational techniques for the self assembly of large structures. In *Proceedings of the 8th Princeton/SSI Conference on Space Manufacturing*, May 1987.

[BW95] Armin Bruderlin and Lance Williams. Motion signal processing. In Robert Cook, editor, *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series, pages 97–104. ACM SIGGRAPH, Addison Wesley, August 1995. held in Los Angeles, California, 06-11 August 1995.

[CA95] Rémi Cozot and Bruno Arnaldi. A language for multibody systems modelling and simulation. In *Proceedings of the European Simulation Symposium*, Erlangen-Nuremberg, 1995.

[CC96] Rémi Cozot and Laurent Cogné. Efficient distributed parallel simulation of multibody systems. In *Proceedings of the 10th European Simulation Multiconference*, Budapest, 1996.

[CCS93] Nak Young Chong, Dongoon Choi, and Il Hong Suh. A generalized motion/force planning strategy for multifingered hands using both rolling and sliding contacts. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Yokohama, Japan, 1993.

[CD95] Rémi Cozot and Stéphane Donikian. General animation and simulation platform. In *Proceedings of the European Simulation Symposium*, Erlangen-Nuremberg, 1995.

[CJ82] E.N. Coleman and Ramesh Jain. Obtaining three-dimensional shape of textured and specular surfaces from four-source photometry. *CGIP*, 18:309–328, April 1982.

[Coh92] Michael F. Cohen. Interactive spacetime control for animation. In Edwin E. Catmull, editor, *Computer Graphics (SIGGRAPH '92 Proceedings)*, volume 26, pages 293–302, July 1992.

[Coz96] Rémi Cozot. *Environnement de simulation de systèmes physiques: modèle et langage.* PhD thesis, Université de Rennes I, 1996.

[CPA95] B. Cloutier, D.K. Pai, and U.M. Ascher. The formulation stiffness of forward dynamics algorithms and implications for robot simulation. In *Proceedings of the IEEE Conference on Robotics and Automation*, 1995.

[CS94] Per H. Christensen and Linda G. Shapiro. Three-dimensional shape from color photometric stereo. *IJCV*, 13(2):213–227, 1994.

[CV96] Rémi Cozot and Dominique Villard. Formal computation of non-rigid multibody systems dynamics. In *Proceedings of the 10th European Simulation Multiconference*, Budapest, 1996.

[DG95] Mathieu Desbrun and Marie-Paule Gascuel. Animating soft substances with implicit surfaces. In Robert Cook, editor, *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series, pages 287–290. ACM SIGGRAPH, Addison Wesley, August 1995. held in Los Angeles, California, 06-11 August 1995.

[DG96] Mathieu Desbrun and Marie-Paule Gascuel. Smoothed particules: a new paradigm for animating highly deformable bodies. In *Proceedings of the Eurographics Workshop on Animation and Simulation*, Poitiers, 1996.

[Dre93] Mark S. Drew. Shape and specularity from color. Technical Report CSS/LCCR TR93-01, Simon Fraser University, Burnaby, B.C., Canada, 1993.

[Fau95] François Faure. Modélisation cinématique du contact pour la dynamique inverse. In *Troisièmes journées de l'Association Française d'Informatique Graphique*, pages 15–22, November 1995.

[Fau96] François Faure. An energy-based method for contact force computation. In *Computer Graphics Forum (Proceedings of EUROGRAPHICS'96)*, volume 15, pages 357–366, August 1996.

[FC88] R.T. Frankot and R. Chellapa. A method for enforcing integrablity in shape from shading algorithms. *IEEE PAMI*, 10(64):439–451, July 1988.

[FDCGM97] François Faure, Gilles Debunne, Marie-Paule Cani-Gascuel, and Franck Multon. Dynamic analysis of human walking. In *Proceedings of the Eurographics Workshop on Animation and Simulation*, 1997.

[Fea83] R. Featherstone. The calculation of robot dynamics using articulated-body inertias. *International Journal of Robotics Research*, 2(1):13–30, 1983.

[Fea87] R. Featherstone. *Robot Dynamics Algorithms*. Kluwer, 1987.

[FRMTT86] L. Forest, D. Rambaud, N. Magnenat-Thalmann, and D. Thalmann. Keyframe-based subactors. In M. Green, editor, *Proceedings of Graphics Interface '86*, pages 213–216, May 1986.

[Gas93] Marie-Paule Gascuel. An implicit formulation for precise contact modeling between flexible solids. In James T. Kajiya, editor, *Computer Graphics (SIGGRAPH '93 Proceedings)*, volume 27, pages 313–320, August 1993.

[GG94] M.P. Gascuel and J.D. Gascuel. Displacement constraints for interactive modeling and animation of articulated structures. *The Visual Computer*, 10(4):191–204, March 1994. An early version of this paper appeared in the *Third Eurographics Workshop on Animation and Simulation*, Cambridge, UK, Sept 92.

[Gle94] M. Gleicher. *A Differential Approach to Graphical Manipulation*. PhD thesis, Carnegie Mellon University, 1994.

[GM85] Michael Girard and Anthony A. Maciejewski. Computational modeling for the computer animation of legged figures. In B. A. Barsky, editor, *Computer Graphics (SIGGRAPH '85 Proceedings)*, volume 19, pages 263–270, July 1985.

[GRBC96] Brian Guenter, Charles F. Rose, Bobby Bodenheimer, and Michael F. Cohen. Efficient generation of motion transitions using spacetime constraints. In Holly Rushmeier, editor, *SIGGRAPH 96 Conference Proceedings*, Annual Conference Series, pages 147–154. ACM SIGGRAPH, Addison Wesley, August 1996. held in New Orleans, Louisiana, 04-09 August 1996.

[HA95] Matthew Holton and Simon Alexander. Soft cellular modelling: A technique for the simulation of non-rigid materials. In *Computer Graphics International '95*, June 1995.

[Hah88] James K. Hahn. Realistic animation of rigid bodies. In John Dill, editor, *Computer Graphics (SIGGRAPH '88 Proceedings)*, volume 22, pages 299–308, August 1988.

[HB89] B.K.P Horn and M.J. Brooks. *Shape from Shading.* MIT Press, 1989.

[HLV96] A. Habibi, A. Luciani, and A. Vapillon. A physically-based model for the simulation of reactive turbulent objects. In *Winter School of Computer Graphics 1996*, February 1996. held at University of West Bohemia, Plzen, Czech Republic, 12-16 February 1996.

[Hor75] B.K.P Horn. Obtaining shape from shading information. In McGraw Hill, editor, *The Psychology of Computer Vision*, pages 115–155, 1975.

[Hor86] B.K.P Horn. *Robot Vision.* MIT Press, 1986.

[Hor90] B.K.P Horn. Height and gradient from shading. *IJCV*, 5(1):37–75, 1990.

[HTSG91] Xiao D. He, Kenneth E. Torrance, Francois X. Sillion, and Donald P. Greenberg. A comprehensive physical model for light reflection. In Thomas W. Sederberg, editor, *Computer Graphics (SIGGRAPH '91 Proceedings)*, volume 25, pages 175–186, July 1991.

[HW90] Charles Herr and Brian Wyvill. Towards generalised motion dynamics for animation. In *Proceedings of Graphics Interface '90*, pages 49–59, May 1990.

[HWBO95] Jessica K. Hodgins, Wayne L. Wooten, David C. Brogan, and James F. O'Brien. Animating human athletics. In Robert Cook, editor, *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series, pages 71–78. ACM SIGGRAPH, Addison Wesley, August 1995. held in Los Angeles, California, 06-11 August 1995.

[IC87] Paul M. Isaacs and Michael F. Cohen. Controlling dynamic simulation with kinematic constraints, behavior functions and inverse dynamics. In Maureen C. Stone, editor, *Computer Graphics (SIGGRAPH '87 Proceedings)*, volume 21, pages 215–224, July 1987.

[IC88] Paul M. Isaacs and Michael F. Cohen. Mixed methods for complex kinematic constraints in dynamic figure animation. *The Visual Computer*, 4(6):296–305, December 1988.

[IH81] K. Ikeuchi and B.K.P Horn. Numerical shape from shading and occluding boundaries. *Artificial Intelligence*, 17(3):141–184, 1981.

[Ike91] Katsushi Ikeuchi. Determining surface orientations of specular surfaces by using the photometric stereo method. *IEEE PAMI*, 3(6), November 1991.

[Jim93] Stéphane Jimenez. *Modélisation et simulation physique d'objets volumiques déformables*. PhD thesis, Institut National Polytechnique de Grenoble, 1993.

[KB82] J. U. Korein and N. I. Badler. Techniques for generating the goal-directed animation of articulated structures. *IEEE Computer Graphics and Applications*, 2(9):71–81, November 1982.

[KB93] Hyeongseok Ko and Norman I. Badler. Straight line walking animation based on kinematic generalization that preserves the original characteristics. In *Proceedings of Graphics Interface '93*, pages 9–16, Toronto, Ontario, Canada, May 1993. Canadian Information Processing Society.

[KSK88] G.J. Klinker, S.A. Shafer, and T. Kanade. The measurement of highlight in color images. *IJCV*, 7(32), 1988.

[Lam60] J.H. Lambert. *Photometria sive de Mensure de Gratibus Luminis et Colorum Umbrae*. Eberhardt Klett, 1760.

[Las87] John Lasseter. Principles of traditional animation applied to 3D computer animation. In Maureen C. Stone, editor, *Computer Graphics (SIGGRAPH '87 Proceedings)*, volume 21, pages 35–44, July 1987.

[Lat86] R.H. Lathrop. Constrained (closed-loop) robot simulation by local constraint propagation. In *Proceedings of the IEEE Conference on Robotics and Automation*, pages 689–694, 1986.

[LB92] Sang Wook Lee and Ruzena Bajcsy. Detection of specularity using color and multiple views. *Image and Vision Computing*, 10:643–653, 1992.

[Lew93] Robert Lewis. Making shaders more physically plausible. In Michael F. Cohen, Claude Puech, and François Sillion, editors, *Fourth Eurographics Workshop on Rendering*, pages 47–62, June 1993.

[LFA75] Jacqueline Lelong-Ferrand and Jean-Marie Arnaudiès. *Cours de Mathématiques (tome 3): Géométrie et Cinématique*. Dunod Université, Paris, 1975.

[LGC94] Zicheng Liu, Steven J. Gortler, and Michael F. Cohen. Hierarchical space-time control. In Andrew Glassner, editor, *Proceedings of SIGGRAPH '94 (Orlando, Florida, July 24–29, 1994)*, Computer Graphics Proceedings, Annual Conference Series, pages 35–42. ACM SIGGRAPH, ACM Press, July 1994. ISBN 0-89791-667-0.

[LHVD95] A. Luciani, A. Habibi, A. Vapillon, and Y. Duroc. A physical model of turbulent fluids. In Dimitri Terzopoulos and Daniel Thalmann, editors, *Computer Animation and Simulation '95*, pages 16–29. Eurographics, Springer-Verlag, September 1995. ISBN 3-211-82738-2.

[Lom95] Jean-Christophe Lombardo. Oriented particles: A tool for shape memory objects modeling. In *Proceedings of Graphics Interface'95*, 1995.

[Lom96] Jean-Christophe Lombardo. *Modélisation d'objets déformables avec un système de particules orientées.* PhD thesis, Université Joseph Fourier, Grenoble, 1996.

[Löt84] P. Lötstedt. Numerical simulation of time-dependent contact friction problems in rigid body mechanics. *SIAM Journal of Scientific Statistical computing*, 5(2):370–393, 1984.

[Luc90] Annie Luciani. Physical models in animation: towards a modular approach. In *Proceedings of the Eurographics Workshop on Animation and Simulation*, 1990.

[LvdP96] Alexis Lamouret and Michiel van de Panne. Motion synthesis by example. In *Proceedings of the Eurographics Workshop on Animation and Simulation*, 1996.

[LvF96] Joseph F. Laszlo, Michiel van de Panne, and Eugene Fiume. Limit cycle control and its application to the animation of balancing and walking. In Holly Rushmeier, editor, *SIGGRAPH 96 Conference Proceedings*, Annual Conference Series, pages 155–162. ACM SIGGRAPH, Addison Wesley, August 1996. held in New Orleans, Louisiana, 04-09 August 1996.

[MD82] M.W.Walker and D.E.Orin. Efficient dynamic computer simulation of robotic mechanisms. *Journal of Dynamic Systems*, (104):205–211, 1982.

[Mil88] Gavin S. P. Miller. The motion dynamics of snakes and worms. In John Dill, editor, *Computer Graphics (SIGGRAPH '88 Proceedings)*, volume 22, pages 169–178, August 1988.

[Mil96] Victor J. Milenkovic. Position-based physics: simulating the motion of many highly interacting spheres and polyhedra. In *SIGGRAPH 96 Conference Proceedings*, pages 129–136. ACM SIGGRAPH, 1996.

[Mon88] David J. Montana. The kinematics of contact and grasp. *International Journal of Robotics Research*, 7(3):17–31, June 1988.

[MP89] Gavin Miller and Andrew Pearce. Globular dynamics: A connected particle system for animating viscous fluids. *Computers and Graphics*, 13(3):305–309, 1989.

[MTT91] Nadia Magnenat-Thalmann and Daniel Thalmann. Complex models for animating synthetic actors. *IEEE Computer Graphics and Applications*, 11(5):32–44, September 1991.

[MW88] Matthew Moore and Jane Wilhelms. Collision detection and response for computer animation. In John Dill, editor, *Computer Graphics (SIGGRAPH '88 Proceedings)*, volume 22, pages 289–298, August 1988.

[MYYT95] S. Miyazaki, T. Yasuda, S. Yokoi, and J. Toriwaki. A study on virtual manipulation of elastic objects. In *Computer Graphics International '95*, June 1995.

[MZ90] Michael McKenna and David Zeltzer. Dynamic simulation of autonomous legged locomotion. In Forest Baskett, editor, *Computer Graphics (SIGGRAPH '90 Proceedings)*, volume 24, pages 29–38, August 1990.

[Nak91] Yoshiro Nakamura. *Advanced Robotics: Redundancy and Optimization.* Addison-Wesley, 1991.

[Oli89] J. Oliensis. Existence and uniqueness in shape from shading. Technical Report COINS TR 89-109, University of Massachussets at Amherst, 1989.

[ON94] Michael Oren and Shree K. Nayar. Seeing beyond lambert's law. In *Proceedings of the DARPA-Esprit Workshop on applications of invariants in computer vision*, 1994.

[Pau81] Richard P. Paul. *Robot Manipulators: Mathematics, Programming, and Control.* MIT Press, Cambridge, MA, 1981.

[Pho75] Bui-T. Phong. Illumination for computer generated pictures. *Communications of the ACM*, 18(6):311–317, June 1975.

[PTVF92] Press, Teukolski, Vetterling, and Flannery. *Numerical Recipes in C.* Cambridge University Press, 1992.

[RH91] Marc H. Raibert and Jessica K. Hodgins. Animation of dynamic legged locomotion. In Thomas W. Sederberg, editor, *Computer Graphics (SIGGRAPH '91 Proceedings)*, volume 25, pages 349–358, July 1991.

[Rin66] T. Rindfleisch. Photometric method for lunar topography. *Photogrammetric Engineering*, 32(2):262–277, March 1966.

[Ros60] J. B. Rosen. The gradient projection method for nonlinear programming. *SIAM Journal of Applied Mathemathics*, 8(1):181–217, 1960.

[Sax88] B.V.H. Saxberg. *A modern differential approach to shape from shading.* PhD thesis, Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA, 1988.

[Sch93] Christophe Schlick. A customizable reflectance model for everyday rendering. In Michael F. Cohen, Claude Puech, and Francois Sillion, editors, *Fourth Eurographics Workshop on Rendering*, pages 73–84. Eurographics, June 1993. held in Paris, France, 14–16 June 1993.

[SH81] Robert Siegel and John R. Howell. *Thermal Radiation Heat Transfer.* Hemisphere Publishing Corporation, 1981.

[SP94] François Sillion and Claude Puech. *Radiosity and Global Illumination.* Morgan Kaufmann, San Francisco, 1994. excellent coverage of radiosity and global illumination algorithms.

[Stu84] D. Sturman. Interactive keyframe animation of 3-D articulated models. In S. MacKay, editor, *Graphics Interface '84 Proceedings*, pages 35–40, 1984.

[SZ90] Peter Schröder and David Zeltzer. The virtual erector set: Dynamic simulation with linear recursive constraint propagation. In Rich Riesenfeld and Carlo Sequin, editors, *Computer Graphics (1990 Symposium on Interactive 3D Graphics)*, volume 24, pages 23–31, March 1990.

[Td91] Hermant D. Tagare and Rui J.P. deFigueiredo. A theory of photometric stereo for a class of diffuse non-lambertian surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(2), February 1991.

[Td93] Hermant D. Tagare and Rui J.P. deFigueiredo. A framework for the construction of reflectance maps for machine vision. *CVGIP: Image Understanding*, 57(3):265–282, May 1993.

[TF88] Demetri Terzopoulos and Kurt Fleischer. Modeling inelastic deformation: Viscoelasticity, plasticity, fracture. In John Dill, editor, *Computer Graphics (SIGGRAPH '88 Proceedings)*, volume 22, pages 269–278, August 1988.

[TPBF87] Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer. Elastically deformable models. In Maureen C. Stone, editor, *Computer Graphics (SIGGRAPH '87 Proceedings)*, volume 21, pages 205–214, July 1987.

[TS67] K. E. Torrance and E. M. Sparrow. Theory for off-specular reflection from roughened surfaces. *Journal of Optical Society of America*, 57(9), 1967.

[TW88] Demetri Terzopoulos and Andrew Witkin. Physically-based models with rigid and deformable components. In *Proceedings of Graphics Interface '88*, pages 146–154, June 1988.

[UAT95] Munetoshi Unuma, Ken Anjyo, and Ryozo Takeuchi. Fourier principles for emotion-based human figure animation. In Robert Cook, editor, *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series, pages 91–96. ACM SIGGRAPH, Addison Wesley, August 1995. held in Los Angeles, California, 06-11 August 1995.

[Ver74] A.F. Vereshchagin. Computer simulation of the dynamics of complicated mechanisms of robot manipulators. *Engineering Cybernetics*, (6):65–70, 1974.

[vF93] Michiel van de Panne and Eugene Fiume. Sensor-actuator networks. In James T. Kajiya, editor, *Computer Graphics (SIGGRAPH '93 Proceedings)*, volume 27, pages 335–342, August 1993.

[vKF94] Michael van de Panne, Ryan Kim, and Eugene Fiume. Virtual wind-up toys for animation. In *Proceedings of Graphics Interface '94*, pages 208–215, Banff, Alberta, Canada, May 1994. Canadian Information Processing Society.

[War92] Gregory J. Ward. Measuring and modeling anisotropic reflection. In Edwin E. Catmull, editor, *Computer Graphics (SIGGRAPH '92 Proceedings)*, volume 26, pages 265–272, July 1992.

[WB85] J. Wilhelms and B. A. Barsky. Using dynamic analysis to animate articulated bodies such as humans and robots. In M. Wein and E. M. Kidd, editors, *Graphics Interface '85 Proceedings*, pages 97–104. Canadian Inf. Process. Soc., 1985.

[WFB87] Andrew Witkin, Kurt Fleischer, and Alan Barr. Energy constraints on parameterized models. In Maureen C. Stone, editor, *Computer Graphics (SIGGRAPH '87 Proceedings)*, volume 21, pages 225–232, July 1987.

[Wil87] Jane Wilhelms. Using dynamic analysis for realistic animation of articulated bodies. *IEEE Computer Graphics and Applications*, 7(6):12–27, June 1987.

[Wit77] J. Wittenburg. *Dynamics of Systems of Rigid Bodies*. B.G. Teubner, Stuttgart, Germany, 1977.

[WK88] Andrew Witkin and Michael Kass. Spacetime constraints. In John Dill, editor, *Computer Graphics (SIGGRAPH '88 Proceedings)*, volume 22, pages 159–168, August 1988.

[WMS88] Jane Wilhelms, Matthew Moore, and Robert Skinner. Dynamic animation: interaction and control. *The Visual Computer*, 4(6):283–295, December 1988.

[Woo80] R.J. Woodham. Photometric method for determining surface orientation from multiple images. *Optical Engineering*, 19, 1980.

[WP95] Andrew Witkin and Zoran Popović. Motion warping. In Robert Cook, editor, *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series, pages 105–108. ACM SIGGRAPH, Addison Wesley, August 1995. held in Los Angeles, California, 06-11 August 1995.

[ZB94a] Jianmin Zhao and Norman I. Badler. Inverse kinematics positioning using nonlinear programming for highly articulated figures. *ACM Transactions on Graphics*, 13(4):313–336, October 1994.

[ZB94b] Jianmin Zhao and Norman I. Badler. Near real-time body awareness. *Computer Aided design*, 26(12):861–868, December 1994.