

David LESAGE

Un modèle dynamique de
spécifications d'ingénierie
basé sur une approche de
géométrie variationnelle

Thèse de doctorat



Éditions des Nik's News
www.niksnews.com/editions/

2003

L'œuvre appartient à son auteur.
L'auteur est seul responsable du contenu de son œuvre.
L'auteur autorise les Éditions des Nik's News à :

- ajouter à son œuvre des informations les concernant ;
- diffuser gratuitement son œuvre ;
- choisir le ou les formats de diffusion de son œuvre.

Les Éditions des Nik's News s'engagent à ne plus publier une œuvre si son auteur le désire.

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

N° attribué par la bibliothèque

--	--	--	--	--	--	--	--	--	--

THÈSE

pour obtenir le grade de

DOCTEUR DE L'INPG

Spécialité : “Mécanique : Conception, Géomécanique et Matériaux”

préparée au laboratoire : Sols, Solides, Structures de Grenoble – France

dans le cadre de l'École Doctorale : “Mécanique et Énergétique”

présentée et soutenue publiquement

par

David LESAGE

le 20 novembre 2002

Un modèle dynamique de spécifications d'ingénierie basé
sur une approche de géométrie variationnelle

Directeur de thèse : M. Jean-Claude LÉON

Composition du jury :

M. Marc DANIEL

M. Dominique MICHELUCCI

M. Bernard YANNOU

M. Jean-Claude LÉON

M. Mike PAYNE

M. Alain RIVIÈRE

M. Pascal SÉBAH

Président

Rapporteur

Rapporteur

Directeur de thèse

Examineur

Examineur

Examineur

À mes parents,
à Karen

Remerciements

Je tiens à remercier en premier lieu l'ensemble de l'équipe du laboratoire 3S et tout particulièrement l'équipe du pôle "Conception Intégrée" avec laquelle j'ai passé de très bon moment.

J'exprime ma gratitude au groupe Dassault Système pour avoir financé mes travaux et m'avoir aussi bien accueilli pendant mes séjours à Suresnes.

Que soient aussi remerciés tous les membres du jury pour l'attention qu'ils ont bien voulu porter à ce travail :

- Marc DANIEL, pour m'avoir fait l'honneur de présider ce jury ;
- Dominique MICHELUCCI et Bernard YANNOU, pour avoir accepté la charge de rapporteurs. Qu'ils trouvent ici toute ma reconnaissance pour avoir pris une partie de leur temps pour lire ce mémoire et en faire une critique constructive ;
- Alain RIVIÈRE, pour m'avoir fait l'honneur de l'examiner ;
- Pascal SÉBAH, pour les bons moments que j'ai pu passer en sa compagnie ;
- et surtout Mike PAYNE, envers qui je serai toujours redevable.

Je tiens également à remercier très sincèrement Jean-Claude LÉON qui a bien voulu diriger mes travaux pendant toutes ces années. Qu'il trouve ici ma gratitude et ma sympathie pour son suivi incessant et attentif, malgré les petits incidents qui sont venus perturber ce projet.

Je ne peux pas oublier de remercier Alice, ma première lectrice, pour son soutien et pour avoir trouver les quelques fautes du manuscrit.

Et enfin, tous mes amis qui ont su me soutenir pendant toutes ces années, en particulier Lionel, Jean-Étienne, Kim, David, Valentin, Nicolas, Guillaume, Nadia, Nico et Marion, Olivier, Jérémy, Guy, Gaby et tous les autres, pour leur aide et les bons moments passés ensemble durant ces quatre ans.

Table des matières

Notations	xv
Introduction	1
I Méthodes de description et de maintien de la cohérence de la géométrie d'un produit	7
1 Principes de description de la géométrie d'un composant	9
1.1 Modélisation géométrique déclarative	9
1.1.1 Une approche générale	9
1.1.2 Modélisation géométrique par caractéristiques	11
1.1.3 Les langages de la modélisation déclarative	12
1.1.4 Conclusion	15
1.2 Les modèles d'information	15
1.2.1 Introduction	15
1.2.2 La modélisation de flux	16
1.2.3 Système de Gestion des Connaissances	17
1.2.4 Un modèle produit multi-vues	17
1.2.5 Vers un modèle géométrique multi-représentation	18
1.2.6 Conclusion	19
1.3 Modeleur géométrique : description explicite de la géométrie	19
1.3.1 Une définition formelle d'un solide	20
1.3.2 Construction procédurale des solides	21
1.3.3 Les solides CSG	23
1.3.4 Les solides B-Rep	25
1.3.5 Méthode de construction B-Rep	28
1.3.6 Les modeleurs hybrides	28
1.4 Conclusion	29
2 Méthodes de mise à jour de la géométrie	31
2.1 Introduction	31
2.2 Modélisation géométrique paramétrée	32

2.2.1	Présentation	32
2.2.2	Analyse	33
2.3	Modélisation géométrique variationnelle	34
2.3.1	Présentation	34
2.3.2	Analyse	34
2.3.3	Problèmes de mise en place	35
2.3.4	Conclusion	36
2.4	Synthèse	36
3	Les solveurs géométriques	37
3.1	Les différentes applications	37
3.2	Approches algébriques	38
3.2.1	Mise en équations d'un problème	38
3.2.2	Méthodes formelles	41
3.2.3	Méthodes numériques	41
3.2.4	Méthodes de décomposition	42
3.2.5	Conclusion	44
3.3	Méthodes géométriques	45
3.3.1	Les systèmes à bases de connaissances	45
3.3.2	Décomposition réalisée par un graphe	48
3.3.3	Décomposition réalisée par une analyse des ddl	53
3.3.4	Conclusion	54
3.4	Résolution de problèmes d'ingénierie	55
3.4.1	Un exemple de problème d'ingénierie	55
3.4.2	Une résolution parallèle	56
3.4.3	Un algorithme itératif	57
3.4.4	Conclusion	58
3.5	Synthèse	58
II	Présentation de l'approche et du modèle d'information des spécifications d'un modèle mécanique	61
4	Description générale de l'approche	63
4.1	Introduction	63
4.2	Cahier des charges	64
4.2.1	Caractéristiques nécessaires	64
4.2.2	Caractéristiques différenciatrices	66
4.3	Description globale de l'approche	67
4.3.1	Architecture générale	68
4.3.2	Concepts du modèle d'information	70
4.3.3	Rôle du solveur	72

4.3.4	Rôle du modelleur	73
4.3.5	Conclusion	73
4.4	Les spécifications	73
4.4.1	Introduction	73
4.4.2	Les éléments géométriques	74
4.4.3	Les contraintes géométriques	74
4.4.4	Les contraintes d'ingénierie	76
4.4.5	Conclusion	76
4.5	Synthèse	77
5	Modèle géométrique	79
5.1	Introduction	79
5.2	La structure	79
5.2.1	Définitions	80
5.2.2	Structure 2D	81
5.2.3	Explications	82
5.2.4	Généralisation au 3D	83
5.3	La toile	84
5.3.1	Structure de données	84
5.3.2	Le support	85
5.4	Conclusion	86
6	Les éléments géométriques	87
6.1	Introduction	87
6.2	Les objets élémentaires	87
6.2.1	Le point	87
6.2.2	La droite, le segment de droite	88
6.2.3	Le plan	88
6.3	Les arcs de courbes	88
6.3.1	Structure et toile d'une courbe	90
6.3.2	Les fonctions $f_{i,m}$	91
6.3.3	Point courant	95
6.3.4	La tangente courante	95
6.3.5	Le rayon de courbure	96
6.4	Les arcs de coniques	98
6.4.1	Les coniques	98
6.4.2	Arc de parabole	99
6.4.3	Arc d'ellipse	101
6.4.4	Arc d'hyperbole	104
6.4.5	Arc de cercle	105
6.5	Conclusion	106

7	L'expression des contraintes géométriques	109
7.1	Introduction	109
7.2	Définitions	110
7.3	Contraintes angulaires 2D	110
7.3.1	Angle "droite – droite"	110
7.3.2	Parallélisme "droite – droite"	111
7.4	Contraintes de coïncidence 2D	111
7.4.1	Coïncidence "point – point"	111
7.4.2	Coïncidence "point – droite"	112
7.4.3	Coïncidence "droite – droite"	113
7.5	Contraintes dimensionnelles 2D	114
7.5.1	Distance "point – point"	114
7.5.2	Distance "point – droite"	114
7.5.3	Distance "droite – droite"	115
7.6	Contraintes de liberté	115
7.7	Les contraintes sur les courbes	117
7.7.1	Contraintes angulaires	117
7.7.2	Contrainte de coïncidence	118
7.7.3	Contrainte de distance	120
7.7.4	Contrainte de continuité "courbe – courbe"	122
7.7.5	Conclusion	122
7.8	Les Contraintes 3D	123
7.8.1	Contraintes angulaires 3D	123
7.8.2	Contraintes de coïncidence 3D	125
7.8.3	Contraintes dimensionnelles 3D	126
7.8.4	Conclusion	128
7.9	Synthèse	128
8	Le modèle algébrique	129
8.1	Introduction	129
8.2	Le modèle algébrique	130
8.2.1	L'estimateur	131
8.2.2	Le nœud "relation"	131
8.3	Du modèle géométrique au modèle algébrique	131
8.3.1	Les relations 2D	133
8.3.2	Ensemble nécessaire et suffisant de relations	135
8.3.3	Conclusion	139
8.4	Le générateur d'équations 2D	139
8.4.1	Qualité d'un système	139
8.4.2	Aspect heuristique	141
8.4.3	Stratégie du générateur d'équations	141

8.4.4	Prise en compte des multipoints	145
8.4.5	Génération dynamique des équations	145
8.5	Généralisation au 3D	146
8.5.1	La trigonométrie sphérique	147
8.5.2	Ensemble nécessaire et suffisant de relations 3D	150
8.5.3	Génération des relations de la trigonométrie sphérique	150
8.6	Des spécifications d'ingénierie au modèle algébrique	151
8.7	Conclusion	151
 III Le solveur d'ingénierie		153
 9 Le module de résolution		155
9.1	Introduction	155
9.2	Analyses et diagnostics	156
9.2.1	Analyse structurelle	156
9.2.2	Analyse qualitative	157
9.2.3	Diagnostics post- ou pré-résolution	158
9.3	Résolution	158
9.3.1	Plan de résolution	158
9.3.2	Critère de choix des algorithmes de résolution	159
9.4	Arbre de solutions	160
9.4.1	Approximation du nombre de solutions	162
9.4.2	Multiplicité des solutions	162
9.4.3	Existence d'une solution	162
9.5	Algorithme global de résolution	163
9.6	Conclusion	164
 10 Plan de résolution		167
10.1	Introduction	167
10.2	Analyse structurelle	168
10.2.1	Algorithme	168
10.2.2	Les PPSI sur-contrainants	171
10.2.3	Les PPSI sous-contrainants	173
10.2.4	Conclusion	176
10.3	Analyse qualitative	176
10.3.1	Introduction	176
10.3.2	Hypothèses autorisant l'analyse	177
10.3.3	Analyse qualitative	178
10.3.4	Les modifications du plan de résolution	179
10.4	Gestion dynamique du plan de résolution	180
10.4.1	Gestion dynamique d'un graphe biparti	180

10.4.2	Gestion du plan de résolution	181
10.5	Conclusion	182
11	La résolution	183
11.1	Algorithme de résolution	183
11.2	Les agents de résolution	184
11.2.1	Agents linéaires	186
11.2.2	Newton-Raphson	186
11.2.3	Homotopie	186
11.2.4	Agent de résolution multi-solutions	189
11.2.5	Solveur algébrique 2D	189
11.2.6	Solveur algébrique 3D	192
11.2.7	Algorithmes d'optimisation	197
11.2.8	Agent de résolution de zones comportant des courbes planes	197
11.3	Les points clés de l'algorithme	200
11.3.1	Choix du solveur	200
11.3.2	Existence d'une solution	201
11.3.3	Recherche de plusieurs solutions	202
11.4	Les méta-agents	202
11.4.1	Le méta-agent de résolution des problèmes sous-contraints	202
11.4.2	Le méta-agent de résolution des problèmes sur-contraints .	203
11.5	Conclusion	203
12	La génération de diagnostics	205
12.1	Introduction	205
12.2	Préparation d'un diagnostic en cas d'échec	206
12.2.1	Recherche des différentes composantes du problème	206
12.2.2	Isolement des différentes parties d'un système	206
12.3	Génération d'un diagnostic	207
12.3.1	Diagnostic pré-résolution	208
12.3.2	Diagnostic post-résolution	208
12.4	Conclusion	210
IV	Maquette, exemples, résultats et conclusion	213
13	Maquette informatique, exemples et résultats	215
13.1	Maquette informatique	215
13.2	Exemples et résultats	216
13.2.1	Un exemple élémentaire de géométrie contrainte	216
13.2.2	Deux arcs de cercles et une parabole	218
13.2.3	Un problème d'ingénierie	224

13.2.4	La spécification de courbes planes	227
13.2.5	Correction du plan de résolution	230
13.2.6	Un exemple “industriel”	234
Synthèse, perspectives et conclusion		237
Annexes et Bibliographie		243
A Rappels de la théorie des ensembles		245
B Compléments de géométrie et topologie des surfaces		247
C Quelques définitions et algorithmes de la théorie des graphes		249
C.1	Introduction	249
C.2	Définitions et concepts de base	250
C.2.1	Généralités	250
C.2.2	Principales définitions	250
C.2.3	Connexité	252
C.2.4	Quelques graphes particuliers	253
C.3	Problème du plus court chemin	255
C.3.1	Définition	255
C.3.2	Algorithme de Moore-Dijkstra	255
C.4	Recherche d’une forêt maximale d’un graphe	256
C.4.1	Algorithme élémentaire	256
C.4.2	Recherche d’un arbre de poids minimum	257
C.5	Base de cycles d’un graphe non orienté	258
C.5.1	Base de cycles quelconques	258
C.5.2	Base de cycles minimaux	259
C.5.3	Gestion dynamique d’une base de cycles	259
C.6	Graphe biparti	262
C.6.1	Couplage	262
C.6.2	Décomposition du graphe biparti	264
C.6.3	Gestion dynamique d’un graphe biparti.	268
C.7	Conclusion	270
D Les complexes cellulaires		271
D.1	Introduction	271
D.2	Complexe cellulaire : définition formelle	271
D.3	Graphe d’adhérence	272
E Complément de géométrie différentielle		273

Bibliographie

275

Table des figures

1	Coûts engagés et dépensés sur tout le cycle du produit	2
1.1	Exemples de scènes obtenues par Chauvat et al.	11
1.2	Un exemple de scènes obtenues par Desmontils et al.	11
1.3	Un exemple de pièce déclarée	12
1.4	Un objet localement dégénéré	21
1.5	Les différentes familles de solides élémentaires	22
1.6	Résultat d'une opération addition classique	22
1.7	Résultat d'opérations booléennes non régularisées	22
1.8	Réalisation d'une opération contextuelle	23
1.9	Exemple d'un arbre de construction	24
1.10	Une structure de données pour un modeleur volumique B-Rep	26
2.1	Un triangle spécifié par un ensemble de contraintes cycliques	33
3.1	Deux exemples de géométrie sous contraintes	38
3.2	Graphe biparti du système d'équations 3.1	40
3.3	Étapes de résolution de la figure 3.1-b	40
3.4	Bassins d'attraction	42
3.5	Graphe biparti du système d'équations 3.1	44
3.6	Illustration de quelques patterns	47
3.7	Trois premiers membres d'une famille d'irréductibles	47
3.8	Graphe de contraintes du problème de la figure 3.1-b	49
3.9	Décomposition suivant Owen	50
3.10	Exemples non-décomposables par l'algorithme d'Owen	51
3.11	Décomposition suivant Hoffmann	52
3.12	Quelques schémas d'assemblage d'Hoffmann	52
3.13	Un exemple de problème d'ingénierie	56
3.14	Architecture proposée par Chung	57
4.1	Concept de toile tendue sur une structure rigide	71
4.2	Représentation UML d'un objet géométrique	71
4.3	Les différentes étapes d'une mise à jour d'un modèle variationnel.	73

5.1	Un robot de manutention	80
5.2	diagramme de classes de la structure 2D	82
5.3	Quelques exemples simples de positionnement de points	83
5.4	Différentes structures pour un même objet	83
5.5	diagramme de classes de la structure 3D	85
5.6	Diagramme de classes partiel d'un modèle géométrique	85
6.1	Structure minimale et toile d'une droite et d'un segment	88
6.2	Structure minimale et toile d'un plan	89
6.3	Influence de la modification du polygone caractéristique	89
6.4	Polygone caractéristique d'une courbe de Bézier	90
6.5	Formulation conventionnelle d'une courbe de Bézier	91
6.6	Structure et toile d'une courbe NURBS	91
6.7	Structure d'un point courant d'une courbe	95
6.8	Structure du vecteur tangent à un point $\mathbf{P}(u)$ d'une courbe	97
6.9	Définition d'une conique par ces caractéristiques intrinsèques	98
6.10	Une parabole	100
6.11	Arc de parabole porté par une courbe de Bézier	100
6.12	Structure d'une parabole	102
6.13	Une ellipse	102
6.14	Structure d'une ellipse	103
6.15	Une hyperbole	104
6.16	Structure d'une hyperbole	105
6.17	Un arc de cercle défini par une courbe NURBS.	106
6.18	Modélisation d'un cercle avec trois courbes NURBS.	106
6.19	Modélisation d'un arc de cercle avec trois courbes NURBS	107
7.1	Les différentes façons de définir l'angle entre deux droites	111
7.2	Les différentes façons de définir la coïncidence entre deux points	112
7.3	Structure de la coïncidence "point – droite"	112
7.4	Exemple de structure contenant une coïncidence "point–droite".	113
7.5	différentes structures pour la coïncidence "droite – droite"	114
7.6	Structure de la contrainte de distance "point – droite"	115
7.7	Structure de la contrainte de distance "droite – droite"	116
7.8	Structure des contraintes de liberté	116
7.9	Structure de la contrainte angulaire	117
7.10	Angle entre deux points appartenant à deux courbes	118
7.11	Coïncidence entre un point \mathbf{A} et une courbe \mathcal{C}	118
7.12	Structure de la contrainte de tangence "courbe – droite"	119
7.13	Structure de la contrainte de tangence "courbe – courbe"	119
7.14	Contrainte de distance entre un point et une courbe	120

7.15	Structure de la contrainte de distance “courbe – droite”	121
7.16	Structure de la contrainte de distance “courbe – courbe”	121
7.17	Contrainte angulaire entre deux plans	123
7.18	Parallélisme entre deux plans	124
7.19	La structure de la contrainte angulaire plan – droite	125
7.20	Parallélisme entre un plan \mathcal{P} et une droite \mathcal{D}	125
7.21	Structure de la contrainte de distance d’un point à un plan	127
7.22	Structure de la contrainte de distance entre plans	127
7.23	La structure de la contrainte de distance d’une droite à un plan	128
8.1	Diagramme de classes d’une relation et de ses dépendances	130
8.2	Les éléments d’un triangle	132
8.3	Graphes affine et vectoriel du problème de la figure 3.1-b	141
8.4	Un triangle sphérique	148
9.1	Un plan de résolution représenté par un simple graphe orienté.	159
9.2	Un plan de résolution séquentiel et l’arbre de solutions associé	161
9.3	Conditions d’existence de deux problèmes géométriques simples.	163
9.4	Algorithme global de résolution.	165
10.1	Algorithme d’analyse structurelle	168
10.2	Plan de résolution	171
10.3	Plan de résolution d’un problème sur-contraint	172
10.4	Différents comportements d’un problème sous-contraint	175
10.5	Plan de résolution d’un système sous-contraint	175
11.1	Algorithme de résolution	185
11.2	État initial et état final d’un triangle	188
11.3	Une courbe homotopique	188
11.4	Un triangle sphérique	192
11.5	Réduction d’un polygone	194
11.6	Les différents polygones sphériques irréductibles	195
11.7	Illustration de la procédure de réduction d’un polygone	196
11.8	Modèle mécanique proposé par Schek	198
11.9	Équilibre d’un noeud du réseau de barre	199
12.1	Un plan de résolution.	206
12.2	Les PPSI nécessaire à la résolution du PPSI courant	207
12.3	Les PPSI dépendants de la résolution du PPSI courant.	208
12.4	Exemple de diagnostic sur l’état de contrainte.	209
12.5	Exemple de diagnostic pour l’échec de la résolution d’un PPSI.	211
13.1	Interface graphique de la maquette informatique	216

13.2	Un premier exemple simple de géométrie contrainte	216
13.3	La structure de l'objet de la figure 13.2	217
13.4	Graphe biparti de l'exemple de la figure 13.2.	218
13.5	Plan de résolution du problème de la figure 13.2.	218
13.6	Un exemple composé de cercles et d'une parabole	219
13.7	Structure de l'objet de la figure 13.6	219
13.8	Graphe vectoriel complété de la structure de l'objet 13.6	220
13.9	Graphe biparti du problème de la figure 13.6.	222
13.10	Plan de résolution du problème de la figure 13.2.	223
13.11	Une solution du problème de la figure 13.6	223
13.12(a)	Une hélice, (b) le pied d'une pale, (c) section à dimensionner .	224
13.13	Spécification géométrique de la section (c) du pied de pale	225
13.14	Structure du pied de pale	225
13.15	Module de définition des spécifications d'ingénierie	227
13.16	Plan de résolution du problème d'ingénierie	228
13.17	Une courbe plane dont un point courant est partiellement spécifié	229
13.18	Structure du problème de la figure 13.17	230
13.19	Résultat obtenu	231
13.20	Un plan de résolution	231
13.21	Identification d'une équation redondante	232
13.22	Recherche du plus petit sous-graphe	232
13.23	Recherche des chaînes alternées augmentantes	232
13.24	Sous plan de résolution	233
13.25	Plan de résolution corrigé	233
13.26	Définition d'une pièce estampée.	234
13.27	Plan de résolution du problème de la pièce estampée	235
13.28	Résultat du problème de la figure 13.26.	236
C.1	Exemple de graphe	250
C.2	Sous-graphe et graphe partiel	251
C.3	Une chaîne élémentaire	251
C.4	Une base de cycles	252
C.5	Composantes connexes et fortement connexes.	253
C.6	Illustration des points d'articulation et des isthmes.	253
C.7	Ce graphe simple est un arbre non orienté	254
C.8	Graphe aux arêtes	254
C.9	Graphe biparti	255
C.10	Deux composantes connexes chacune complétée d'une arête	261
C.11	Couplages complet, maximum et parfait	263
C.12	Recouvrement extérieur quelconque	264
C.13	Graphes réductible, semi-réductible et irréductible	265

C.14 Décomposition canonique d'un graphe 266

Notations

$\text{vol}(A)$ volume d'une partie A d'un espace vectoriel E de dimension finie ;
 $|x|$ valeur absolue de $x \in \mathbb{R}$;

Ensembles

\mathbb{N} ensemble des entiers ;
 \mathbb{R} ensemble des réels ;
 \mathbb{C} ensemble des complexes ;
 $\{a, b, c\}$ L'ensemble constitué des 3 éléments a, b, c ;
 $x \in X$ x est l'élément de l'ensemble X ;
 $x \notin X$ x n'est pas un élément de l'ensemble X ;
 $A \subset X$ A est contenu dans l'ensemble X ;
 $A \cup B$ Réunion des éléments de A et B ;
 $A \cap B$ Intersection de A et B ;
 $A - B$ Ensemble des éléments de A qui n'appartiennent pas à B ;
 $A \Delta B$ Différence symétrique de A par B : $A \Delta B = (A \cup B) - (A \cap B)$;
 $A \times B$ Produit cartésien de A par B : $A \times B = \{(a, b) \mid a \in A, b \in B\}$;
 $\overset{\circ}{A}$ l'intérieur de l'ensemble A ;
 \tilde{A} l'adhérence de A ;
 \overline{A} complémentaire de A ;
 ∂A le bord de l'ensemble A ;
 $|A|$ nombre d'éléments de l'ensemble A , cardinal de A ;

Espace vectoriel

$\dim(E)$ dimension de l'espace vectoriel E ;
 \mathbf{a} vecteur d'un espace vectoriel E de dimension n ;
 $\mathbf{a} \wedge \mathbf{b}$ produit vectoriel des vecteurs \mathbf{a} et \mathbf{b} ;
 $\mathbf{a} \cdot \mathbf{b}$ produit scalaire des vecteurs \mathbf{a} et \mathbf{b} ;
 $d(e_1, e_2)$ distance euclidienne entre les entités e_1 et e_2 ;

$\|\mathbf{a}\|$ norme euclidienne du vecteur \mathbf{a} . Afin d'alléger les équations lorsqu'il n'y a pas de confusion possible, la norme euclidienne du vecteur \mathbf{a} sera simplement notée \mathbf{a} ;

Espace affine

$a, \dots, z, \alpha, \dots, \zeta$ scalaires $\in \mathbb{R}$;
 $\mathbf{a}, \dots, \mathbf{z}$ vecteurs $\in \mathbb{R}^3$ (ou à \mathbb{R}^2);
 \mathbf{AB}, \dots bipoints $\in \mathbb{R}^3$ (ou à \mathbb{R}^2);
 $\widetilde{\mathbf{AB}}, \dots$ multipoint $\in \mathbb{R}^3$ (ou à \mathbb{R}^2);
 $\widehat{\mathbf{AB}}, \dots$ copoint $\in \mathbb{R}^3$ (ou à \mathbb{R}^2);
 $\mathbf{A}, \dots, \mathbf{Z}$ points $\in \mathbb{R}^3$ (ou à \mathbb{R}^2);
 $\left(\widehat{\mathbf{a}}, \mathbf{b}\right)$ angle orienté défini par les vecteurs \mathbf{a} et \mathbf{b} ;

Graphe

$G = [X, U]$ graphe dont l'ensemble des sommets est X , et dont l'ensemble des arcs (ou arêtes) est U ;
 $\nu(G)$ nombre cyclomatique du graphe G ;
 Γ_x ensemble des successeurs du sommet x ;
 $\omega(A)$ ensemble des arcs ayant une extrémité et une seule appartenant à un sous-ensemble de nœuds A (cocycle);
 $\omega^+(A), \omega^-(A)$ ensemble des arcs ayant respectivement seulement l'extrémité initiale dans A , seulement l'extrémité finale dans A ;
 $E(G)$ dimension extérieure d'un graphe G ;

Courbes et surfaces paramétriques

C_m^p coefficients du binôme de Newton pour un polynôme de degré m , nombre de combinaisons de m objets pris p à p :

$$C_m^p = \frac{m!}{p!(m-p)!};$$
 $B_{i,m}(u)$ $i^{\text{ème}}$ polynôme de Bernstein de degré m :

$$B_{i,m}(u) = \sum_{i=0}^m C_m^i u^i (1-u)^{m-i};$$
 $N_{i,m}(u)$ $i^{\text{ème}}$ fonction de base B-Spline constituée de polynômes de degré m ;
 $B_{i,m}^*$ $i^{\text{ème}}$ fonction de base pour le modèle Bézier rationnel constituée

$N_{i,m}^*(u)$ de polynômes de degré m ;
i^{ème} fonction de base pour le modèle NURBS constituée
de polynômes de degré m ;

Introduction

A une époque où réactivité et compétitivité sont les maîtres mots du monde industriel, les méthodes et les outils de travail sont en pleine évolution. Avec l'arrivée de l'informatique dans les années 1960, de nouveaux outils ont permis aux différents acteurs industriels de se concentrer sur le processus de création, laissant aux machines la charge de réaliser les tâches de bas niveau. C'est au début des années 1990, alors que le marché n'est plus en expansion, que les entreprises ont dû revoir leurs méthodes de travail afin de privilégier la créativité et l'innovation, tout en réduisant la durée du cycle de développement de leurs produits.

Dans le monde de la mécanique, le processus de conception est fortement remis en cause en raison de son caractère séquentiel. Ce processus, qui *“consiste à donner un ensemble de propositions permettant de décrire le produit (forme, dimensions, moyens d'obtention...) et répondant globalement à un cahier des charges (fonctions à assurer, conditions de fonctionnement, durée de vie souhaitée, environnement,...)”* [Tichkiewitch *et al.* 1993], regroupe toutes les étapes : de la définition des besoins, à la destruction et au recyclage d'un produit.

Le modèle de conception séquentiel, né pendant la révolution industrielle avec les débuts de la production en série, est calqué sur une logique d'organisation des métiers. Les projets sont découpés en tâches. La coordination temporelle des différentes tâches est assurée par une organisation de gestion de projet. Un des points faibles de cette approche est qu'il n'est pas prévu de coopération entre les acteurs des différents services. Les contraintes de chaque service sont alors résolues localement, sans marge de manœuvre, ce qui conduit à une juxtaposition d'optimum locaux n'aboutissant pas à un compromis global efficace [Prudhomme 1999].

Ce modèle fait place à un modèle d'ingénierie concurrente. Les moyens pour mettre en œuvre ces nouvelles logiques d'organisation coopérative de la conception sont les *plateaux projets* et les *groupes fonctions*. L'objectif de ces modèles est de *“prendre en compte **au plus tôt** les contraintes de différents métiers du cycle de vie du produit”* [Prudhomme 1999], ceci afin de répondre aux besoins de compétitivité et de réactivité. Il est maintenant reconnu que les coûts d'un produit sont engagés dès les premières phases de conception (Figure 1). C'est pendant ces phases que les actions sur la réduction des coûts donnent des résultats optimum

alors que ces actions n'étaient jusqu'alors attachées qu'à la phase de fabrication du produit, et par conséquent, ne concernaient que 20% du coût d'un produit.

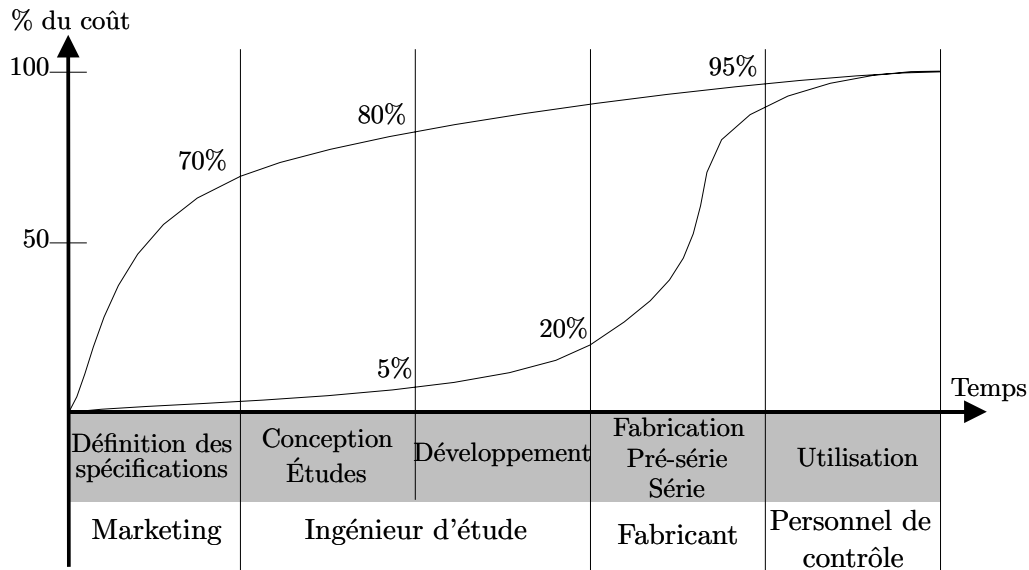


FIG. 1: Coûts engagés et dépensés sur tout le cycle du produit (secteur spatial) [Clautrier 1991]

“L’idée maîtresse” des plateaux projets “réside dans la mise en place d’une collégialité de la conception, où tous les acteurs potentiels sont présents à tous moments, et où chacun doit pouvoir réagir afin de donner un avis motivé, envisageant les conséquences des prises de décisions collectives. Ce mode de fonctionnement permet, tout au moins sur le principe, de résoudre une grande part des problèmes posés par la conception linéaire” [Tichkiewitch 1995].

Les acteurs intervenant dans les processus de conception d’un produit industriel ont en général des compétences, des responsabilités et des objectifs différents, mais ils ont tous une tâche commune : définir un maximum de spécifications. Ces informations sont de différentes natures et portent sur différents aspects du produit (fonctionnels, structurels) mais également sur toutes les données relatives à sa fabrication, à son usage, à sa destruction... Il est alors nécessaire de pouvoir disposer de plusieurs modèles d’information d’un produit afin de fournir à chacun une représentation adaptée à sa tâche et à ses compétences.

L’enjeu est donc d’intégrer au mieux, dans un environnement de conception, cette notion de multi-représentation. Ceci ne pourra être réalisé de façon pertinente que si des outils de gestion et de maintien de la cohérence des spécifications sont associés. En effet, les éléments décrivant un produit sont de nature différente mais sont liés. La modification d’une spécification doit pouvoir être répercutée sur les différents ensembles d’informations. De même, l’ajout de spécifications doit laisser cet ensemble cohérent.

Dans le monde de la conception mécanique, les outils informatiques essayent de répondre à ces nouveaux besoins. En effet, les logiciels de CAO¹ intègrent de plus en plus d'applications permettant aux acteurs d'avoir plusieurs représentations et des outils adaptés à leurs tâches. Ainsi, les différents systèmes ont commencé par intégrer des outils de FAO² puis des outils d'analyse et de simulation pour devenir de nos jours de véritables solutions PLM³ permettant ainsi la création et la simulation du cycle de vie intégral du produit, de la préconception à la mise en fabrication.

Les différentes solutions PLM ne sont pas de simples regroupements d'applications. Les modèles produits sont communs à tous les modules. Ces modèles intègrent la notion de *feature*⁴. Cette notion permet de donner un sens aux différentes entités créées par une application. Pour les applications de conception, l'objet manipulé n'est plus une entité purement géométrique définie par un ensemble d'opérations booléennes et d'entités géométriques élémentaires, mais devient un objet défini par des caractéristiques comme, par exemple, le matériau. Ces caractéristiques peuvent être fonction des métiers : un trou peut être une portée de roulement pour le concepteur, alors que pour la fabrication ce trou sera un alésage avec un état de surface et une certaine tolérance sur le diamètre. Ainsi, chaque acteur va pouvoir enrichir la définition du produit et réagir au plus tôt en fonction des choix réalisés par les autres acteurs.

Chaque application d'une solution PLM représentera l'ensemble des informations de plusieurs façons afin de faciliter la lecture par l'utilisateur. La vue la plus courante, mais aussi la plus importante pour le concepteur, est la vue graphique 3D lui offrant une représentation virtuelle de son sujet de travail.

Dans le monde de la mécanique, des solutions de multi-représentations informationnelles commencent à être proposées. Malheureusement, le maintien de la cohérence des spécifications n'est pas encore une fonctionnalité présente à tous les niveaux de ces solutions. Bien au contraire, seuls quelques modules commencent à gérer ces problèmes de cohérence et pour un ensemble limité de spécifications. Le premier module est le *sketcher*⁵. La gestion de la cohérence à ce niveau est globalement de bonne qualité bien que l'ensemble des spécifications soit généralement limité aux spécifications géométriques et dimensionnelles. Les modules de conception 3D (généralement appelés *Part Design*) gèrent aussi le maintien de la cohérence de celles-ci avec leurs représentations. À ce niveau, les contraintes dimensionnelles sont beaucoup plus limitées. En effet, elles sont soit les paramètres de position d'un objet par rapport aux autres, soit imposées

¹Conception Assistée par Ordinateur.

²Fabrication assistée par ordinateur.

³Product Lifecycle Management : Gestion du Cycle de vie du Produit.

⁴caractéristique.

⁵module 2D d'un logiciel de CAO permettant de réaliser les contours définissant la section d'une pièce 3D.

par les méthodes de construction des objets géométriques 3D. Alors, le maintien de la cohérence ne consiste qu'à rejouer l'historique de construction avec les nouveaux paramètres.

Le thème de recherche, dans lequel s'inscrit le travail de thèse présenté dans ce manuscrit, a pour objectif de proposer des nouvelles méthodes de maintien de la cohérence des spécifications d'un composant mécanique pour les solutions PLM. Les travaux ont été centrés autour des spécifications géométriques, bien que des propositions de solutions au problème général soient présentées.

Afin d'introduire les différentes propositions, la première partie de ce manuscrit présentera une synthèse des travaux réalisés dans les différents domaines abordés :

- en amont de la géométrie, des réflexions sont portées sur des méthodes de description explicite du produit. Ces approches permettent à l'utilisateur de décrire sa pièce ou son mécanisme par un ensemble de spécifications d'une sémantique élaborée ;
- une introduction aux méthodes de description de la géométrie d'un composant permettra de familiariser le lecteur sur les moyens de définir un objet dans les environnements de CAO où deux courants d'idées essayent de cohabiter. D'une part, un courant de tendance formaliste, définissant la forme d'un objet par sa surface et, d'autre part, un courant de tendance constructiviste, définissant un objet par l'historique de construction. Finalement, la solution choisie par les éditeurs de logiciel est de plus en plus une solution intermédiaire entre ces deux courants ;
- l'évolution des spécifications rend le maintien de la cohérence de la géométrie de plus en plus difficile. Les différents travaux menés afin de répondre à ce problème peuvent être répartis dans deux familles qui seront présentées. La première, appelée *géométrie paramétrée* est présente essentiellement dans les modules de *part design*. Son principe général est de maintenir la géométrie en "rejouant" l'historique de construction. La seconde dite, *géométrie variationnelle*, doit être associée à un module de résolution. En effet, la nature des contraintes utilisables ne permet plus une résolution séquentielle du problème. Cette famille n'est actuellement présente que dans certains "*sketchers*";
- le dernier thème abordé par cet état de l'art porte sur les différents algorithmes de résolution. Les méthodes dédiées aux problèmes géométriques seront alors analysées en détails. Les méthodes de résolution d'un système d'équations seront présentées dans une autre partie de l'exposé.

La seconde partie de ce manuscrit commence par la présentation de l'approche proposée dans ses grandes lignes. Afin de préciser les objectifs, un "cahier des charges" commenté d'un modèle géométrique variationnel dédié à la CAO précède

l'introduction des propositions.

La solution choisie est obtenue à partir d'un modèle d'information dont la cohérence est déléguée à deux modules distincts mais complémentaires. Le premier module est un modeleur géométrique gérant la topologie du modèle géométrique alors que le second est un solveur dont le rôle est d'assurer la cohérence des différents paramètres du problème. La description de ce modèle termine la seconde partie.

La troisième partie présente le module de résolution. Ce dernier est un solveur qui aspire à être générique, c'est-à-dire qui est en mesure de résoudre une grande diversité de systèmes d'équations. En plus de résoudre, il offre à l'utilisateur des fonctionnalités d'analyse et de diagnostic d'échec et de degrés de liberté.

Cette dissertation s'achève par une quatrième partie dans laquelle la présentation détaillée de quelques exemples permettront d'illustrer l'ensemble des propositions. Une synthèse reprenant les différents résultats encourageants de cette approche, suivie des différentes perspectives, permettra de conclure.

Première partie

Méthodes de description et de maintien de la cohérence de la géométrie d'un produit

Chapitre 1

Principes de description de la géométrie d'un composant

Dans ce chapitre, les différentes possibilités de décrire la géométrie d'un composant seront abordées. La première partie présente la modélisation géométrique déclarative. L'objectif de cette approche est d'engendrer des formes ou des ensembles de formes par la donnée d'un ensemble de propriétés ou de caractéristiques. La deuxième partie s'attache aux modèles d'information. Ils permettent d'enregistrer et de traiter la totalité des données liées aux différents aspects d'un produit. Enfin, ce chapitre se termine par la description des modélisateurs géométriques destinés à la CAO.

1.1 Modélisation géométrique déclarative

Pendant l'apogée de la recherche sur l'intelligence artificielle, des actions ont été lancées sur ce thème dans les équipes de CAO. Un des sujets principaux était la géométrie déclarative, mais le principe de remplacer une réflexion humaine par de l'intelligence artificielle a progressivement été abandonné. De nos jours, les recherches portent donc plus sur des systèmes d'assistance.

Les premiers travaux à base d'intelligence artificielle sont rapidement présentés. Ensuite, l'introduction de la modélisation par caractéristiques présentera les spécifications dimensionnelles.

1.1.1 Une approche générale

De nombreuses équipes de recherche ont tenté de définir un modélisateur géométrique déclaratif dans le sens décrit par Michel Lucas : *“La modélisation déclarative repose sur l'idée que nous ne pouvons appréhender le monde autrement que par*

sa description géométrique : nous pouvons le percevoir par ses propriétés, par ses caractéristiques, c'est-à-dire non pas seulement par l'apparence qu'il nous présente mais par les mécanismes et les contraintes qui font qu'il nous apparaît sous cette forme. Ainsi, nous nous plaçons à un plus haut niveau d'abstraction.

L'objectif de la modélisation déclarative de formes est de permettre d'engendrer des formes (ou des ensembles de formes) par la simple donnée d'un ensemble de propriétés ou de caractéristiques. L'ordinateur est chargé d'explorer l'univers des formes potentielles afin de sélectionner celles correspondant à la définition donnée. Le concepteur n'a plus qu'à choisir, à l'aide d'outils appropriés, la ou les formes qui lui conviennent. Ainsi, il se trouve libéré des calculs et peut se concentrer sur la phase de création.” [Lucas et al. 1995].

Avec un modeleur traditionnel, l'homme qui doit répondre à un problème commence par avoir une idée pouvant amener à une solution. Il la précise ensuite en déterminant l'ensemble des spécifications à respecter. À partir de là, il conçoit mentalement un objet assez précis. De cette image mentale, il construit petit à petit l'objet avec son logiciel de CAO. Les logiciels actuels permettent de plus en plus de contraindre l'objet par un ensemble de spécifications dimensionnelles afin de faciliter des corrections ou modifications ultérieures. Cette phase, relativement longue, se compose d'un ensemble d'étapes successives de construction, contrôle et correction jusqu'à obtenir un objet conforme aux spécifications.

Avec un modeleur déclaratif, le principe est de décrire les propriétés, les caractéristiques et les contraintes qui doivent être satisfaites par l'objet à construire. Un modèle d'utilisation d'un tel modeleur est proposé dans [Colin et al. 1997]. L'idée est qu'à partir de cette reformulation du cahier des charges d'un objet, le modeleur déclaratif produit automatiquement une ou plusieurs formes en adéquation avec la demande de l'utilisateur. Par principe, il n'est pas nécessaire d'appliquer des tests de validité sur ces objets puisqu'ils correspondent parfaitement au cahier des charges. Par contre, dans le cas général, il existe une infinité de solutions répondant à l'ensemble des spécifications. En effet, afin de ne pas restreindre trop rapidement le champs de solutions, l'utilisateur doit décrire la fonction à réaliser plutôt qu'une solution possible. La description devient alors trop vague pour un ordinateur. Les temps de calcul s'allongent. Le nombre de solutions est trop grand pour que l'utilisateur puisse ensuite faire un choix.

Malgré tout, des résultats encourageants ont été obtenus sur des exemples simples comme pour la modélisation déclarative de courbes gauches [Daniel 1995], la modélisation déclarative de volumes pour la création de scènes [Chauvat 1994] (Fig. 1.1), la répartition déclarative de boîtes [Desmontils et al. 1997] (Fig. 1.2).

Dans le monde de la CAO *industrielle*, les résultats de ce niveau n'ont toujours pas abouti. Afin de permettre des modifications ultérieures, les modeleurs utilisent l'enregistrement de l'historique de construction. Cet historique est généralement une description précise et explicite de la pièce. Par exemple, pour la pièce de

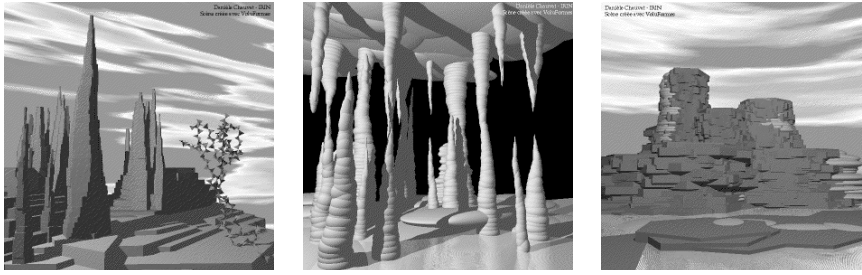


FIG. 1.1: Exemples de scènes obtenues par Chauvat et al.

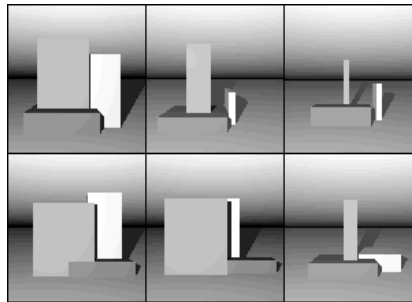


FIG. 1.2: Un exemple de scènes obtenues par Desmontils et al. répondant à la déclaration : “Il y a 3 cubes. Soit le cube A. A est grand. A n’est pas profond. Soit le cube B. La largeur de B est très importante. B n’est pas grand. Soit le cube C. C est derrière A. Le centre de C est à droite. C est plus grand que B.”

la figure 1.3 : “Soit une pièce en marbre, parallélépipédique de hauteur 20, de base rectangulaire de longueur 120 et de largeur 80. Sur la face supérieure, il y a un trou centré de profondeur 15 de section elliptique : diamètre du grand axe 100, du petit axe 60. Un chanfrein ($3 \times 45^\circ$) casse l’arête supérieure du trou. Toutes les autres arêtes sauf celles de la face inférieure du parallélépipède sont arrondies par un congé de raccordement de rayon 5”. Ainsi, d’un certain point de vue, l’utilisateur décrit au logiciel, les caractéristiques du modèle. On parle alors de modélisation par caractéristiques géométriques, ce qui constitue un premier niveau, basique, de modélisation déclarative.

1.1.2 Modélisation géométrique par caractéristiques

Les modeleurs paramétriques ont été, sans le vouloir, les premiers modeleurs déclaratifs utilisables dans un contexte industriel. Un exemple classique est un modeleur CSG conservant un historique de construction (Cf. sous-section 1.3.2). En effet, avec ces modeleurs, l’utilisateur déclare les étapes de construction de son objet géométrique et peut ainsi facilement modifier les paramètres de construction. Cette approche a été améliorée pour devenir la modélisation par caractéristiques¹, de plus en plus répandue dans le monde de la CAO. Avec ces mo-

¹dans le monde de la CAO, le terme anglais *feature* est très utilisé

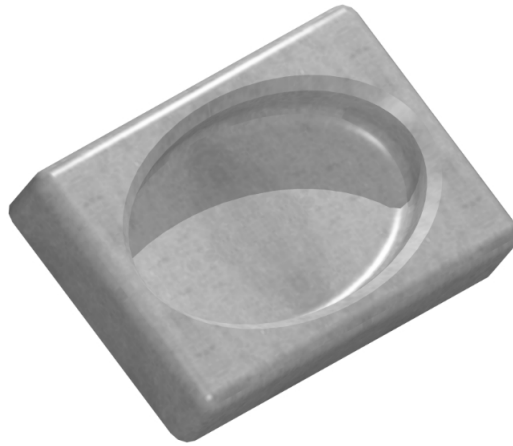


FIG. 1.3: Un exemple de pièce déclarée

de leurs, un objet est défini par l'ensemble de ses caractéristiques géométriques, dimensionnelles mais aussi mécaniques, thermiques, etc...

La modélisation géométrique variationnelle, qui pour l'instant n'existe que dans la plupart des *sketchers*², est la seule approche purement déclarative des logiciels de CAO industriels. En effet, dans ce module l'utilisateur définit une géométrie par un ensemble d'éléments positionnés les uns par rapport aux autres selon des contraintes dimensionnelles (longueur, distance, rayon, ...) et géométriques (coïncident, concentrique, parallèle, ...). Pour compléter le modèle, des relations d'ingénierie peuvent être ajoutées. Ce module est purement déclaratif. En effet, l'utilisateur ne fait que décrire l'objet géométrique par ses composants et par leurs relations. Les étapes de construction n'ont alors aucune importance. C'est le modelleur qui gère la construction de la solution. Au contraire, avec un modelleur CSG, l'utilisateur ne peut décrire une pièce que par des étapes de construction de la géométrie. Les spécifications dimensionnelles sont alors imposées par le processus de construction.

Quelle que soit l'approche utilisée, les spécifications dimensionnelles ont un sens bien défini. Ce langage est en effet un langage de base pour la conception mécanique. Il a donc été étudié, formalisé afin d'être univoque et international.

1.1.3 Les langages de la modélisation déclarative

L'ensemble des contraintes dimensionnelles forme une partie du langage nécessaire à la géométrie déclarative. En effet, à partir de ses concepts, l'utilisateur peut définir la forme, les dimensions et les caractéristiques de la surface d'une pièce qui lui assurent un fonctionnement optimal, ainsi que la variation tolérée autour de cet optimum pour laquelle la fonction est toujours respectée.

²module géométrique 2D d'un logiciel de CAO utilisé pour définir les contours nécessaires à la définition de primitives 3D

Or, ce langage n'a pas été défini pour la modélisation géométrique. Au contraire, bien avant l'apparition de la CAO dans les bureaux d'études, il a été défini afin de spécifier de façon précise les dimensions mais aussi les tolérances, les états de surfaces acceptables pour une pièce sur un dessin de définition. De nombreuses équipes de recherche ont donc travaillé sur la définition d'un langage nécessaire et suffisant de spécification. Ces équipes ont été relayées par les organismes de normalisation afin que ces langages soient univoques et internationaux. Les résultats les plus connus sont le modèle SATT, la normalisation ISO pour la cotation et l'échange de données CAO.

Le modèle SATT

Le modèle SATT³, initialement développé pour décrire le tolérancement des pièces mécaniques, définit la plupart des concepts nécessaires à la spécification des contraintes géométriques. Ce modèle s'appuie sur la structure mathématique du groupe des déplacements [Rivière 1993], [Clément *et al.* 1994]. Les mots clés de ce modèle sont :

les sept classes de surfaces, il a été démontré que le groupe de déplacements des solides, dans l'espace euclidien tridimensionnel, est subdivisé en douze sous-groupes (transformation identité, translation unidirectionnelle, ..., déplacement quelconque). Une analyse de ces sous-groupes de déplacements a permis de montrer qu'il n'en existe que sept qui laissent des surfaces invariants. Ainsi une surface appartient à un des sept sous-groupes :

- transformation identité,
- translation unidirectionnelle,
- rotation autour d'un axe,
- mouvement hélicoïdal,
- rotation autour d'un axe combiné avec une translation suivant le même axe,
- mouvement plan sur plan,
- rotation sphérique ;

le reclassement, l'association de surfaces forme une surface qui appartient à un des sept sous-groupes. Le reclassement permet d'étudier de manière exhaustive tous les cas d'association possibles afin de déterminer à quelle classe appartient le résultat ;

les éléments géométriques de référence minimum (EGRM), un EGRM est composé des trois éléments géométriques simples que sont le point, la droite et le plan. Il est associé à chaque classe de surfaces et il permet le positionnement univoque de la classe de surfaces à laquelle il est rattaché ;

³Surfaces Associées Technologiquement et Topologiquement

les treize contraintes géométriques, chaque surface peut être positionnée de façon univoque par un ensemble d'éléments géométriques simples. L'ensemble nécessaire et suffisant pour positionner ces éléments géométriques est un ensemble de 13 contraintes géométriques.

Ce modèle a été complété par P. Serré [Serré 2000] afin de traiter deux points qui jusqu'alors posaient problèmes : la gestion des ambiguïtés par la restriction des solutions grâce aux contraintes de chiralité et la définition univoque de contraintes géométriques entre surfaces. Les travaux de C. Cubeles-Valade [Cubeles-Valade 1998] ont permis de prendre en compte les surfaces réelles dans les problèmes de métrologie des pièces mécaniques.

La norme ISO

Afin de définir un langage univoque et international, les organismes de normalisation ont créé des comités techniques, chargés de la spécification et de la vérification dimensionnelle et géométrique des produits.

Ainsi, par exemple, l'organisation ISO⁴ a édité différentes normes autour du projet GPS⁵ afin de définir un langage standard. On peut citer quelques-uns des principaux documents généraux de cette norme : [ISO/TR 14638], [ISO 14660-1], [ISO 5458], [ISO 4287], [ISO 8785]. D'autres travaux portent sur la normalisation d'un langage graphique pour le dessin technique [ISO 128] et sur le langage graphique des spécifications dimensionnelles [ISO 129].

Les formats neutres

Avec la CAO, un nouveau problème est apparu : l'échange de données entre différents logiciels. Dans un premier temps, les organismes de standardisation ont défini des formats géométriques neutres (IGES⁶, VDA-FS⁷, SET⁸) afin de transférer des modèles géométriques entre différents systèmes.

Cette multiplicité des standards d'échange est contraire à l'idée même de standard. Le développement d'un standard unique, reconnu à l'échelle internationale et exploitant pleinement les acquis des différentes démarches est donc entrepris par l'ISO. Son Comité Technique 184 "Systèmes d'Automatisation industrielle et Intégration" au travers de son sous-comité 4 "Données industrielles", auquel coopèrent de nombreux experts venant de différents pays, élabore depuis plusieurs années une norme internationale d'échange de données complète d'un

⁴International Organisation for Standardization

⁵Spécification Géométrique des Produits

⁶Initial Graphics Exchange Specification

⁷Verband Der Automobilindustrie Flächen Schnittstelle

⁸Standard d'Echange et de Transfert

produit (spécifications géométriques, dimensionnelles et ingénierie) : la norme STEP⁹ [ISO 10303].

Conclusion

Quelques langages de spécification ont été rapidement introduits. Une description plus détaillée est réalisée dans [Serré 2000].

1.1.4 Conclusion

Dans un premier temps, la modélisation déclarative a été présentée comme un objectif très général, alors que l'introduction de la modélisation par caractéristiques a montré son potentiel industriel pour la modélisation géométrique. Cette présentation a permis d'introduire les différents modèles de spécifications géométriques.

Ces modèles sont globalement similaires et ont un vocabulaire suffisant pour la spécification géométrique et/ou dimensionnelle, mais ils n'ont été conçus (sauf STEP) que pour des problèmes purement géométriques. Or, la géométrie ne constitue qu'une partie des données associées aux modèles pour la conception. Ces données, composées de connaissances et d'informations sur le produit, se construisent ou se précisent en même temps qu'évolue le processus de conception. En début de conception, le produit peut être simplement défini à partir d'un diagramme fonctionnel alors qu'en fin de conception, une définition géométrique et technologique détaillée est disponible. La conception d'un produit fait ainsi appel à tout un ensemble d'outils complémentaires qui vont contribuer, soit à définir, soit à formaliser les choix de conception effectués.

Les interactions qui existent entre ces données se formalisent peu à peu au travers de nouveaux modèles, outils et normes (STEP). Elles permettent d'assurer une disponibilité et une accessibilité des informations associées au produit, à chaque étape de la conception. Le but étant à terme d'être en mesure de garantir, tout au long du processus, la cohérence de l'ensemble des informations relatives au produit. Ces modèles sont présentés dans la section suivante.

1.2 Les modèles d'information

1.2.1 Introduction

L'activité de conception peut être identifiée à l'action d'effectuer des choix techniques pour faire évoluer la définition d'un produit au juste besoin. Cette activité est au cœur des préoccupations de l'équipe *Conception Intégrée* (CI) du laboratoire 3S de Grenoble. Aidés par les sociologues du laboratoire CRISTO de

⁹Standard for the Exchange of Product model data

Grenoble, ils l'ont analysée afin de proposer des méthodes et des outils plus en adéquation avec le triplet "qualité, coût, délais".

L'idée principale qui ressort de cette analyse est le principe de conception intégrée. Intégrée pour prise en compte de tout le cycle de vie d'un produit, mais aussi pour la prise en compte de son environnement de développement et de l'ensemble des différents acteurs devant intervenir à un moment ou à un autre dans la vie du produit. En effet, la conception d'un produit est le fruit d'échanges et de mises en accord d'intérêts très variés. Elle résulte donc d'un compromis entre des logiques d'acteurs différentes.

Cette activité génère et utilise un ensemble de données et de connaissances important. En début de conception, la vision du produit est floue. Les acteurs disposent plutôt d'une description fonctionnelle. Tout au long du cycle de conception, la nature des données générées évolue et se diversifie énormément. La description du produit devient détaillée, rigoureuse et propre à chaque métier. Afin de faire participer simultanément des hommes de cultures professionnelles différentes (marketing, design, calcul, fabrication, maintenance, recyclage, etc), il est important de leur proposer des structures de données adaptées à l'interaction entre métiers, leur permettant d'intervenir pour un juste besoin.

Dans un process d'ingénierie simultanée, un système de conception assisté doit donc disposer d'une structure de données "produit" capable de stocker les éléments manipulés par chacun des métiers, de mémoriser les interactions entre ces éléments, de pouvoir les restituer aux différents acteurs en fonction de leurs besoins, et enfin de permettre la propagation des contraintes entre ces différents acteurs [Tichkiewitch *et al.* 1995].

Différents travaux de l'équipe CI ont débouché sur des propositions de structure pour : la spécification d'un modèle fonctionnel, la spécification d'un modèle produit multi-vues, l'utilisation et la réutilisation du calcul mécanique et enfin la multi-représentation de modèles géométriques.

1.2.2 La modélisation de flux

Un travail [Constant 1996] basé sur la notion de flux physique et sur le concept d'association de surfaces a permis de proposer une modélisation du produit en cours de conception. Ce modèle permet le passage d'une représentation purement fonctionnelle à une définition structurelle minimale d'une solution possible. Il utilise les SATT [Clément *et al.* 1994] et la théorie des groupes de déplacements, ainsi que des concepts issus des Bond-Graphs.

À partir d'un graphe minimal d'association de surfaces, décrivant les flux d'énergie d'origines diverses (mécanique, fluide, électrique...) entre le produit et son environnement, il permet, à l'aide d'opérateurs de décomposition, de générer des niveaux de définition du produit de plus en plus précis. Contrairement à

tous les modèles de représentation classique comme le schéma cinématique en mécanique, ou le circuit équivalent en électrotechnique, ce modèle est adapté à des systèmes multi-physiques. De plus, il propose une première gestion de la cohérence des spécifications entre les différents niveaux, mais elle ne porte que sur les paramètres de position relative entre surfaces fonctionnelles.

1.2.3 Système de Gestion des Connaissances de Calcul en Conception

Après l'analyse sur le terrain de l'utilisation et du déroulement des calculs de structure en mécanique, une méthodologie associée à une structure de données a été proposée par N. Troussier [Troussier 1999].

Alors que la méthodologie favorise une utilisation pertinente des calculs de structures, la base de données capitalise l'ensemble des informations associées aux différentes analyses du produit. Un bon accès à ces données permet en plus une réutilisation des modèles, des démarches de calcul et des connaissances issues du calcul.

1.2.4 Un modèle produit multi-vues

Les travaux de D. Constant [Constant 1996] ont permis de proposer un modèle supportant le passage d'une représentation fonctionnelle d'un produit à une représentation structurelle. Ceux de N. Troussier [Troussier 1999] ont permis d'obtenir une structuration intéressante des données afin de faciliter la coopération entre le concepteur et l'analyste mais aussi de générer et capitaliser un savoir-faire et des connaissances issus des calculs de structures. D'autres études ont tenté d'apporter une solution globale adaptée à l'interaction entre les différents acteurs de la conception. Le modèle produit proposé par S. Tichkiewitch en est un exemple.

Il est défini, pour être un modèle multi-vues, dans une application partagée. Avec celui-ci, un produit est défini par un ensemble de composants, liens, relations [Tichkiewitch 1995]. Un composant représente un ensemble matériel identifiable du produit. Chaque composant est défini par un ensemble de liens. Un lien est une caractéristique permettant un regard externe sur le composant. Ces liens sont reliés par des relations.

La notion de point de vue permet de séparer les descriptions d'un même composant suivant des intérêts différents. Un produit a des caractéristiques communes ou liées entre différents points de vue. Les relations vont alors pouvoir définir les interactions. Une notion de filtre va permettre à chaque acteur de travailler avec les données qui lui sont pertinentes.

Cette approche gère différents niveaux d'abstraction. En effet, un composant

peut être détaillé par un ensemble de composants. Pour les mêmes raisons, une relation entre différents liens peut être remplacée par un ensemble composant – liens – relations.

Cette structure multi-vues offre un état du projet de conception au fur et à mesure de son évolution. Il permet, aux différents acteurs, d'intervenir dans la définition du produit à toutes les phases de la conception. En effet, pour chaque point de vue l'intéressant, il peut choisir le niveau de définition adapté à son besoin. Les derniers travaux ont permis d'offrir à l'utilisateur une représentation d'une vue grâce à l'ajout d'interfaces graphiques spécifiques [Roucoules 1999].

1.2.5 Vers un modèle géométrique multi-représentation

Les modèles de données géométriques sont les modèles de données les plus utilisés en conception. Ils ont l'avantage d'être d'une lecture facile, rapide. Généralement, ces modèles réduisent le besoin d'une interprétation personnelle. Ils sont la plupart du temps le seul langage commun à toute l'équipe de conception. De plus, avec les systèmes de CAO actuels, les modèles géométriques offrent un aspect réaliste tel qu'il n'est plus nécessaire d'être spécialiste pour les comprendre.

Quoi qu'il en soit, il est toujours nécessaire de générer de nombreux modèles géométriques de natures différentes : des modèles exacts pour la définition finale du produit, pour la fabrication ; des modèles facettisés avec différents niveaux de détails pour la visualisation ; des modèles polyédriques plus ou moins idéalisés pour le calcul de structures ; des modèles discrétisés pour les calculs d'écoulements ; des modèles idéalisés pour les schémas cinématiques, ...

Ces différentes représentations géométriques peuvent être soit une représentation graphique d'une vue particulière, soit un composant d'une vue dans un modèle produit. Par exemple, le modèle géométrique idéalisé pour un calcul de structures n'est qu'une partie des données nécessaires au problème, alors qu'un modèle B-Rep est une représentation complète de la géométrie d'un composant.

Le problème actuel est de conserver des liens entre les différents modèles géométriques générés tout au long du cycle de conception. En effet, pour l'instant, le seul modèle géométrique qui est maintenu cohérent avec l'ensemble des spécifications est le modèle B-Rep. Les autres modèles sont générés pour un besoin particulier et ne conservent finalement aucun lien avec les spécifications. Pour remédier à cela, différents travaux ont été réalisés afin d'apporter des débuts de solutions :

- les travaux de F. Noël pour définir un mailleur auto-adaptatif pour des surfaces gauches [Noël 1994] ;
- les travaux de P. Véron pour la simplification de modèles polyédriques [Véron 1997] ;

- les travaux de S. Maza pour la génération de maillages multi-résolution [Maza 1998];
- les travaux de L. Fine pour l'adaptation et l'idéalisation de modèles dédiés à l'analyse de structures mécaniques [Fine 2001].

1.2.6 Conclusion

Dans un premier temps, les besoins en modèles de données pour la conception ont été introduits. Ensuite, les différents travaux réalisés sur ce thème au laboratoire 3S ont été présentés.

Or, les systèmes de CAO deviennent des vrais systèmes PLM. Ils gèrent la vie complète d'un produit, de la conception à la fabrication jusqu'au recyclage en passant par le marketing et la maintenance. Il n'est plus suffisant de définir un produit seulement par la définition géométrique de ses composants. Il est primordial de le définir suivant tous ses points de vue comme : la vue marketing, la vue résistance mécanique, la vue processus de fabrication, ..., la vue processus de recyclage. Il apparaît donc de plus en plus important de proposer des "modèles produits" permettant de gérer et de capitaliser les données associées à un produit, mais il faut aussi que cet ensemble de données reste constamment cohérent.

Après avoir introduit les "modèles produits", les principes de la modélisation géométrique vont être présentés. Ils permettent de définir de façon rigoureuse un solide d'un point de vue géométrique. Contrairement à la modélisation volumique utilisée dans les domaines tels que les jeux, l'image de synthèse, la rigueur géométrique est très importante pour les systèmes de CAO. En effet, les modèles réalisés ne servent pas simplement à une représentation graphique d'un produit, ils définissent les modèles de référence pour tous les autres modules d'un logiciel de CFAO.

1.3 Modeleur géométrique : description explicite de la géométrie

Après avoir vu quelques méthodes pour spécifier une géométrie, les différentes approches pour construire et sauvegarder cette géométrie d'un point de vue mathématique vont être présentées. Le but principal de ces méthodes est de permettre la réutilisation du modèle mathématique par d'autres applications (comme la simulation numérique et le calcul de trajectoires d'usinage, ...). En d'autres termes, il faut que ces méthodes disposent, entre autres, d'outils permettant de savoir si un point est à l'intérieur, à l'extérieur, ou sur la surface d'un solide. Ce type de fonctionnalité différencie un modeleur géométrique dédié à la CAO, d'un modeleur utilisé pour faire simplement de la représentation graphique 3D. Dans

ce cas, seul l'aspect visuel est important.

À ce stade, deux courants d'idées essayent de cohabiter. D'une part, un courant de tendance formaliste définissant la forme d'un objet par sa surface, d'autre part un courant de définition procédurale décrivant un objet par son historique de construction. Les différences entre ces deux courants d'idées se situent au niveau de la définition et de la construction d'un solide. Comme la solution choisie par les éditeurs de logiciels est, de plus en plus, une solution hybride, les différents aspects de chaque approche sont présentés dans un ordre facilitant la compréhension.

Mais avant tout, précisons la définition d'un solide.

1.3.1 Une définition formelle d'un solide

La CAO permet de construire une représentation virtuelle, dans un espace virtuel, d'un objet *réel* ou *idéalisé* selon les besoins d'une "vue" du produit. Afin de pouvoir utiliser cet objet dans d'autres applications, un ensemble de propriétés élémentaires doit être vérifié.

Un solide

J.F. Rameau dans [Rameau 1998] propose une première définition *physique* d'un objet solide :

Solide : Un *solide* est consistant, de masse finie (non nulle). De plus, il ne dégénère pas localement en morceaux filaires ou minces.

À partir de la théorie des ensembles (Cf. annexe A), il reformule la définition d'un point de vue mathématique :

Solide : une partie S de \mathbb{R}^3 est un solide si :

- S est bornée ;
- S est fermée ;
- S est une variété topologique de dimension 3 ;
- S est connexe.

Un solide est de volume fini, le sous-espace occupé par ce solide est donc fermé, borné. La matière est toujours contenue dans un volume, donc dans une variété topologique de dimension 3. En effet, une surface (variété topologique de dimension 2) n'a pas d'épaisseur, elle ne peut pas contenir de la matière (Figure 1.4-a). Le solide doit être connexe. En effet, il n'est pas concevable "physiquement" d'avoir une pièce définie par deux parties disjointes mais rigides l'une par rapport à l'autre...

Cette définition très stricte n'est finalement pas la définition généralement utilisée par les modelleurs industriels car des idéalizations des objets sont utilisées. La troisième propriété (S est une variété topologique) apparaît de son côté comme étant une contrainte plutôt qu'un avantage pour les utilisateurs finaux

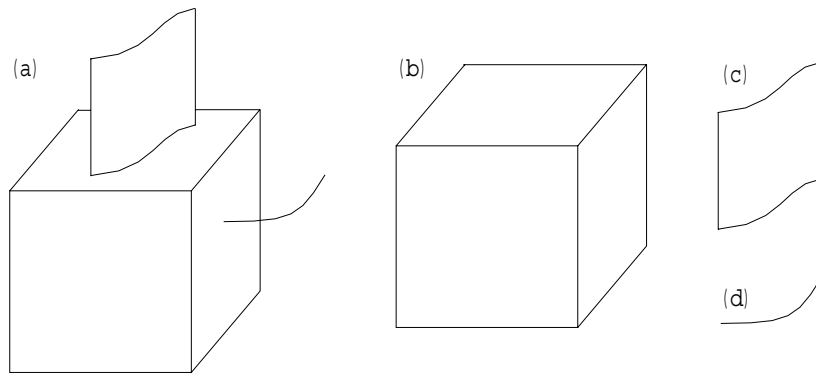


FIG. 1.4: La figure a présente l'exemple d'un objet localement dégénéré constitué des objets b, c et d. La partie b est une variété de dimension 3 alors que la variété de la partie c est de dimension 2 et celle la partie d de dimension 1.

car des besoins de représentation idéalisée existent pour l'analyse de structures, par exemple. D'autres définitions ont été proposées comme celle de Requicha [Requicha 1980].

1.3.2 Construction procédurale des solides

Le modèle procédural ou CSG¹⁰ est une méthode de construction d'un objet géométrique basée sur des opérations entre solides. L'objet volumique est défini de manière constructive et procédurale à partir de primitives géométriques et d'une série de règles ou d'opérations de "construction" basées sur une logique booléenne. Aux opérations sont associées un historique. Il va permettre de capter l'intention de conception de l'utilisateur pour autoriser les modifications ultérieures.

Les solides élémentaires sont soit des primitives simples, soit des primitives à contour. Dans la première catégorie, on retrouve les formes élémentaires : parallélépipède, cylindre, sphère, tore (Figure 1.5-a). Les primitives à contour sont les prismes, solides de révolution, tuyaux et pyramides. Elles sont construites à partir d'un contour de base et d'une opération comme par exemple, l'extrusion d'un profil (Figure 1.5-b).

Opérations booléennes

Les opérations booléennes sont des opérations logiques issues de la théorie des ensembles. À partir de deux ensembles, un troisième ensemble est construit en fonction de l'opérateur. Ces opérateurs sont :

- l'opérateur d'union " \cup " ;
- l'opérateur d'intersection " \cap " ;
- l'opérateur de soustraction " $-$ " ;

¹⁰Constructive Solid Geometry

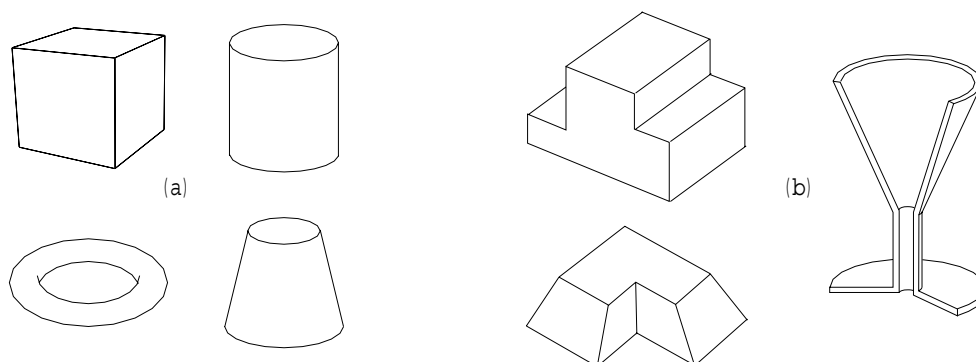


FIG. 1.5: Les différentes familles de solides élémentaires : (a) primitives simples ; (b) primitives à contour

La figure 1.6-c représente un exemple d'union entre les ensembles 1.6-a et 1.6-b. Le résultat de cette opération ne vérifie pas toujours la définition proposée pour un solide (Cf. sous-section 1.3.1). Par exemple, l'union de deux ensembles disjoints donne un ensemble non connexe. La figure 1.7 présente deux autres exemples d'opérations dont le résultat n'est pas conforme. L'opération d'intersection entre les figures 1.7-a et 1.7-b donne un ensemble (Figure 1.7-c) qui n'est pas une variété. La soustraction du solide A avec le solide B (Figure 1.7-d) donne l'ensemble non fermé de la figure 1.7-e.

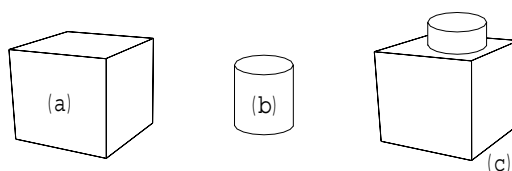


FIG. 1.6: Résultat d'une opération addition classique

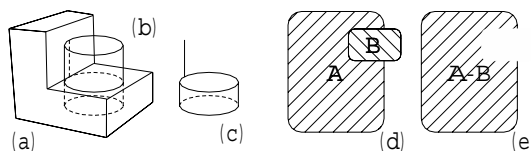


FIG. 1.7: Résultat d'opérations booléennes non régularisées

Les opérateurs booléens régularisés pour les solides sont une adaptation des opérateurs standards afin d'assurer un résultat conforme à la définition d'un solide [Requicha 1980]. La seule partie de la définition qui n'est pas prise en compte est la connexité.

Opérations contextuelles

Une limite de l'approche CSG est le manque d'outil de modification d'un objet adapté à certains métiers. En effet, dans le monde de la mécanique, les arêtes vives sont généralement supprimées des pièces usinées et remplacées par un chanfrein. Dans le monde de la forge, elles sont remplacées par un arrondi et les arêtes vives rentrantes par un congé de raccordement. En plus de ces opérations sur les arêtes, de nombreuses autres fonctions destinées à des corps de métiers particuliers existent, par exemple celles réalisant les dépouilles, les sur-épaisseurs, les coques à partir d'un volume massif, ...

Ces opérations peuvent, dans certains cas, être réalisées par un ensemble d'opérations booléennes. Par exemple, la figure 1.8-a présente un solide de révolution qui va permettre de réaliser un chanfrein sur une arête d'un cylindre pour obtenir le solide de la figure 1.8-b. Construire un congé de raccordement au niveau de la liaison de deux cylindres (Figure 1.8-c) devient une opération beaucoup plus délicate en n'utilisant que des opérations booléennes.

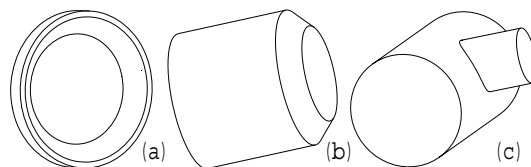


FIG. 1.8: Réalisation d'une opération contextuelle à partir de solide primitif et d'opérations booléennes.

L'arbre de construction

Afin de permettre des modifications de la géométrie, les différentes étapes de construction sont enregistrées. L'utilisateur peut alors modifier un solide primitif et demander la mise à jour de l'objet.

L'historique est écrit sous la forme d'un arbre où chaque feuille est un solide primitif et chaque nœud le résultat d'une opération booléenne (Figure 1.9).

L'arbre peut être modifié en déplaçant, copiant, supprimant, remplaçant des branches. Pour des solides complexes, les manipulations de l'arbre permettent de l'alléger. Néanmoins, la manipulation de l'arbre de construction n'est pas une opération aisée.

1.3.3 Les solides CSG

Les formes implicites

Afin de mettre en place une approche de construction procédurale, il faut définir les objets élémentaires sur lesquels la définition du solide (Cf. sous-section 1.3.1)

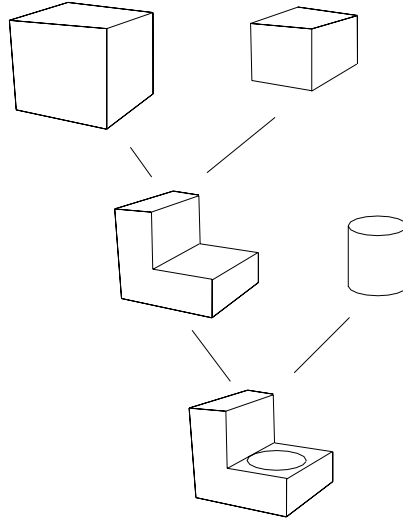


FIG. 1.9: Exemple d'un arbre de construction

va pouvoir être vérifiée. La première idée est de choisir une approche ensembliste : l'ensemble des points définissant le volume est connu de façon explicite. Pour cela, il suffit de construire une fonction $F_i(x, y, z)$, $\{x, y, z\} \in \mathbb{R}^3$ telle que :

$$F_i(x, y, z) \geq 0, \quad (1.1)$$

pour tous les points appartenant au solide.

Par exemple, l'espace occupé par un volume peut être défini par l'intersection d'un ensemble de demi-espaces, c'est-à-dire par un système d'inéquations implicites. Pour une boîte de largeur a , de profondeur b et de hauteur c , l'intérieur est défini (dans un repère particulier) par le système :

$$F_i(x, y, z) \begin{cases} x \\ a - x \\ y \\ b - y \\ z \\ c - z \end{cases} \cdot \quad (1.2)$$

De même, l'espace occupé par une boule de centre (a, b, c) et de rayon R est définie par l'équation :

$$F_i(x, y, z) = R^2 - (x - a)^2 - (y - b)^2 - (z - c)^2. \quad (1.3)$$

L'intérêt des volumes définis par un système d'inéquations implicites est de savoir directement si un point (x, y, z) appartient ou non à un solide. Mais l'ensemble restreint de solides élémentaires et la difficulté pour définir les opérations autres que booléennes (Cf. sous-section 1.3.2) limitent les performances de ces modeleurs.

1.3.4 Les solides B-Rep

Cette méthode de modélisation des solides est basée sur les propriétés topologiques de la surface d'un volume. Le principe de cette méthode ne repose donc plus sur les moyens de définir l'intérieur du solide mais sur la définition de la frontière de ce volume. Une raison qui semble être majeure à l'utilisation de cette approche est que la plupart des applications de CAO ont des algorithmes basés sur la peau du solide. La visualisation, le calcul de trajectoires d'usinage, l'édition de plans normalisés en sont quelques exemples.

Un solide peut être parfaitement défini à partir de sa surface. Cette surface doit, pour cela, vérifier un ensemble de propriétés topologiques afin d'être conforme à la définition d'un solide (Cf. sous-section 1.3.1). Une nouvelle définition peut alors être énoncée.

Solide : Une surface S définit un solide si et seulement si S est fermée, orientable, sans intersection avec elle-même. La surface d'un solide est donc *compacte*.

Pour les mêmes raisons que le modèle ensembliste, un solide peut ne pas être connecté.

Cette définition est indépendante de la forme physique du solide. En effet, d'un point de vue topologique un cube est identique à une sphère et une tasse est identique à un tore. Afin de pouvoir vérifier en permanence cette définition, la surface S de l'objet est découpée en un ensemble de *surfaces topologiques*¹¹ à bord S_i tel que :

$$\left\{ \begin{array}{l} \bigcup_i S_i = S \\ \bigcap_i \overset{\circ}{S}_i = \emptyset \end{array} \right. .$$

L'intérieur d'une *surface topologique* est généralement appelé *face*, sa frontière est définie par un ensemble de courbes topologiques fermées d'intérieur homéomorphe à un cercle ou une droite appelé *arête*. Le bord des arêtes est définie par deux points appelés *sommets*. Cette ensemble de faces F_i , arêtes E_i , sommets V_i définit un *pavage* du bord du volume.

Pavage : Un ensemble de faces ouvertes F_i , d'arêtes ouvertes E_i , et de sommets V_i est un *pavage* de S s'il constitue une partition de S .

Un ensemble de propriétés existe afin de vérifier qu'un ensemble de faces, arêtes et sommets définissent bien un pavage et donc la surface d'un volume topologique. Les principales sont [Lehmann *et al.* 1982] :

Fermé : Une surface est fermée si chaque arête d'un pavage est adjacente à deux faces.

¹¹variété topologique de dimension 2

Orientable : Une surface est orientable si l'orientation des normales à la surface est invariante pour tous les chemins possibles sur la surface.

L'invariant d'Euler-Poincaré : On appelle *invariant d'Euler-Poincaré* d'une surface compacte S le nombre entier χ tel que :

$$\chi = b_0(S) - b_1(S) + b_2(S),$$

avec $b_i(S)$ les nombres de Betti. $b_0(S)$ est le nombre de composantes connexes de la surface, $b_2(S)$ le nombre de composantes connexes orientables et $b_1(S)$ est égal à 0 (respectivement 1, 2, $2g$) pour les composantes homéomorphes à une sphère (respectivement une bouteille de Klein, un tore, un tore à g anses). L'invariant d'Euler-Poincaré d'une sphère (respectivement d'une bouteille de Klein, d'un tore, d'un tore à g anses) est 2 (respectivement 0, 0, $2 - 2g$).

Indice d'un pavage : Quelque soit un pavage d'une surface compacte S , admettant $F(S)$ faces, $E(S)$ arêtes, et $V(S)$ sommets :

$$F(S) - E(S) + V(S) = \chi.$$

L'invariant d'Euler-Poincaré définit l'indice du pavage.

Dans la pratique, afin de mettre en place une approche B-rep, la partie topologique est séparée de la partie géométrique : une face, une arête, un sommet sont portés respectivement par une surface, une courbe, ou un point. La figure 1.10 propose une structure de données d'un modelleur B-Rep.

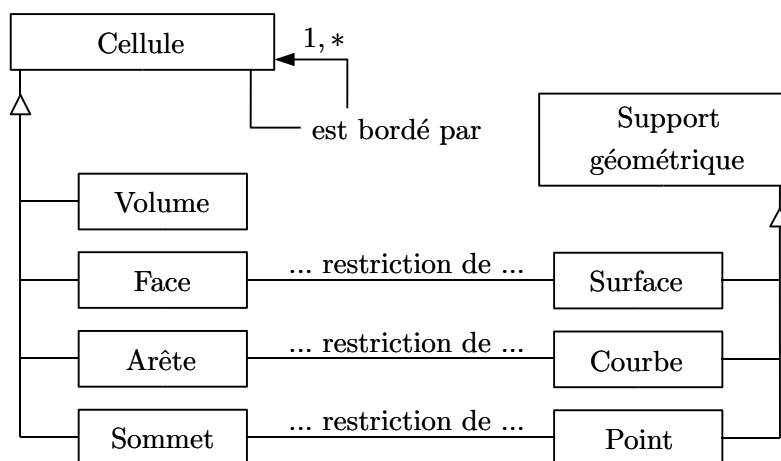


FIG. 1.10: Une structure de données pour un modelleur volumique B-Rep

Finalement, les algorithmes nécessaires pour la partie géométrique sont des algorithmes de calculs d'intersections surface-surface, courbe-surface, droite-surface [Léon 1991], [Lukács 1989]. L'intersection droite-surface permet, par exemple, de

dire si un point est à l'intérieur ou à l'extérieur d'un volume avec des algorithmes de lancé de rayons. Ces calculs sont réalisés à une précision numérique près. La conséquence est d'avoir certaines entités définies par plusieurs approximations géométriques différentes. Par exemple, la courbe définissant l'intersection de deux surfaces peut-être définie par trois courbes numériques distinctes : la première est définie dans l'espace de travail alors que les deux autres sont définies sur les surfaces. Jackson [Jackson 1995] propose d'utiliser une précision numérique locale appropriée aux différentes entités mises en jeu. L'intérêt de cette approche est de faciliter l'importation, l'utilisation et la modification de modèles réalisés sur différents systèmes de CAO utilisant des précisions différentes.

La gestion de la topologie d'un solide est un problème traité à partir des outils de la théorie des graphes. De nombreuses approches ont été proposées comme l'utilisation de cartes généralisées introduites par Lienhardt [Lienhardt 1992] ou le concept du complexe cellulaire. Afin de vérifier la cohérence topologique d'un objet, de nombreuses propositions ont été formulées dans la littérature dont [Sakkalis *et al.* 2000] où les fondements de la modélisation volumique B-Rep sont présentés suivis d'une démonstration rigoureuse des conditions suffisantes pour qu'un pavage définisse un solide. Quoi qu'il en soit, de nombreuses opérations sur le solide donnent un résultat non-conforme. Rossignac [Rossignac *et al.* 1999] propose un algorithme qui corrige le modèle non-conforme en un modèle infiniment proche mais conforme. Cette approche est initialement destinée aux modèles polyédriques mais il est imaginable de la généraliser aux modèles B-Rep classiques. Malgré tout, il est dans certains cas préférable de travailler plus ou moins temporairement avec un modèle non-conforme.

Les solides B-Rep dégénérés.

La modélisation volumique a pour finalité la création, la modification, l'utilisation d'une représentation virtuelle d'un objet réel. Les différents modèles présentés sont basés sur une définition mathématique suffisamment stricte d'un solide pour que toutes les propriétés du solide réel puissent être calculées à partir du modèle virtuel. Or, pour un grand nombre d'applications, l'utilisation d'un modèle (temporairement) dégénéré permet d'accroître considérablement les performances des algorithmes.

De nombreuses réflexions portent donc sur la spécification d'un modèle de solides dégénérés [Gueorguieva *et al.* 1994], [Cavalcanti *et al.* 1997], [Lee *al.* 2001]. Le point délicat est de définir des outils de modification de la géométrie dont le comportement est intuitif pour l'utilisateur.

1.3.5 Méthode de construction B-Rep

Un solide B-Rep est défini par sa surface. Elle sépare l'intérieur de l'extérieur. La première idée afin de modifier un solide est de modifier la frontière de ce solide. L'ensemble des outils repose logiquement sur ce principe. Il peut tout de même être divisé en deux groupes : les opérations *CSG* (bossage, trou) et les opérations contextuelles (chanfrein, congé, sur-épaisseur, dépouille).

La première catégorie d'outils peut-être vue comme la création d'une primitive géométrique combinée avec une opération booléenne, mais certaines étapes de construction d'un solide peuvent s'avérer être plus judicieuses à partir d'une méthodologie purement *CSG*. De plus, du point de vue "implémentation", il n'est pas rare que ces outils soient en interne découpés en "étapes *CSG*". Des opérateurs booléens ont donc été définis pour les modelleurs B-Rep : les *opérateurs booléens régularisés* [Requicha 1980]. Cazier propose, de son côté, les opérateurs booléens destinés au modèle proposé par Lienhardt [Cazier *et al.* 1996]. Comme l'utilisation de solides dégénérés semblent être, dans certains cas, un bon choix stratégique (Cf. sous-section 1.3.4) les opérateurs booléens ont été adaptés à ces modèles [Murayama 1996].

Les opérations contextuelles (Cf. sous-section 1.3.2) sont des opérations qui ne peuvent généralement pas être obtenues par une opération "CSG". Elles sont obtenues par transformation locale de la surface : une nouvelle surface est calculée et est ajoutée au modèle. Le calcul de cette surface est l'étape la plus délicate pour ces outils. Léon dans [Léon 1991] présente le principe de construction de carreaux de raccordement à profil circulaire. Varady réalise une étude plus détaillée sur les problèmes topologiques et géométriques de la réalisation des congés [Varady *et al.* 1989]. Il faut noter que la qualité des opérateurs contextuels d'un modelleur B-Rep est un critère significatif de la qualité d'un modelleur.

Que ce soit pour les opérations booléennes ou les opérations contextuelles, une fois la surface calculée, il faut la greffer au solide. C'est un problème topologique géré par des opérateurs topologiques [Marcheix *et al.* 1995].

1.3.6 Les modelleurs hybrides

Les modelleurs hybrides sont basés sur un compromis entre l'approche *CSG* et l'approche B-Rep. Un solide est défini par une approche B-Rep. Pour des questions de performance, l'arbre *CSG* est adapté. En effet, pour l'approche *CSG* seul l'historique de construction est utile puisqu'il est à la base des différents calculs d'intersection. Pour une approche hybride, peut être ajoutée à chaque nœud de l'arbre, la géométrie résultant de l'opération booléenne. La re-construction d'un objet consiste alors à ne re-calculer que les opérations sur le chemin menant de la primitive modifiée à la racine de l'arbre.

1.4 Conclusion

Les première et deuxième parties de ce chapitre ont introduit l'intérêt de la modélisation par caractéristiques. La présentation des "modèles produits" a confirmé l'importance de la gestion des informations géométriques et non géométriques dans un processus de conception. Le point capital dans la gestion des données est le maintien de la cohérence de l'ensemble des informations. Actuellement, ce maintien est géré manuellement par les différents acteurs. Les répercussions ne sont pas immédiates et pas toujours complètes, ce qui peut faire coexister des données contradictoires.

La troisième partie a présenté un état de l'art au niveau modélisation volumique. Les différents modèles décrits sont définis pour représenter une certaine forme de géométrie, mais cette représentation est rigide. Il n'est pas prévu que celle-ci puisse évoluer. En effet, le positionnement des différentes entités n'est pas considéré comme une information principale. Les entités sont simplement positionnées dans un repère particulier.

Cependant, les systèmes de CAO proposent tout de même des solutions de mises à jour. Les différents travaux menés afin de répondre à ce problème peuvent être répartis dans deux familles qui sont présentées dans le chapitre suivant.

Chapitre 2

Méthodes de mise à jour de la géométrie

La possibilité de modifier aisément un objet géométrique par la correction de la valeur d'un paramètre a permis aux systèmes de CAO de faire une avancée importante. L'introduction de ce chapitre présente les différents intérêts industriels. Ensuite deux familles de solutions seront expliquées.

2.1 Introduction

Deux qualités permettent aux industries mécaniques d'être concurrentielles : réactivité et compétitivité. L'introduction de ce manuscrit a présenté la conception intégrée comme une solution reconnue dans le monde industriel pour augmenter la compétitivité d'un produit. En effet, la participation au plus tôt des différents acteurs du cycle de vie d'un produit permet d'obtenir plus facilement une qualité optimale.

La réactivité d'une équipe peut se mesurer facilement par le temps nécessaire pour proposer une solution adaptée à un problème. Dans certaines industries, en particulier chez les sous-traitants – qui fournissent en grande quantité une petite gamme de produits – la principale tâche des concepteurs est de modifier un produit existant afin de vérifier les nouveaux paramètres imposés par le donneur d'ordres. L'équipe de conception doit alors adapter le modèle. Ce travail n'apporte que peu de valeur ajoutée au produit existant et doit être réalisé dans les plus brefs délais...

Avec l'apparition de la modélisation paramétrée, ce travail a été simplifié. L'utilisateur n'a alors besoin que de modifier la valeur d'un ensemble de paramètres. En pratique, cette étape n'est pas si simple. En effet, les paramètres à modifier sont rarement ceux du modèle. C'est ici qu'apparaît la principale

raison de ce travail de thèse : l'utilisateur doit pouvoir paramétrer son modèle géométrique par un ensemble de paramètres fonctionnels, ce qui n'est actuellement pas toujours possible. Les chapitres suivants exposeront les difficultés cachées.

Pour des raisons techniques, un modèle volumique n'est définissable que par une approche paramétrée. La section suivante présente et analyse cette approche. Les modules d'esquisse 2D proposent une approche dite "variationnelle". Ce sera d'ailleurs le thème de la troisième section.

2.2 Modélisation géométrique paramétrée

La modélisation géométrique paramétrée est essentiellement présente dans les modeleurs géométriques 3D, bien que quelques modeleurs 2D (*sketcheurs*) paramétriques ont été proposés. La raison est essentiellement technologique. La mise en place d'une telle approche en 3D est parfaitement complémentaire des approches de modélisation volumique (Cf. section 1.3), alors qu'une solution paramétrique dans un module 2D est trop restrictive pour l'utilisateur. La première partie de cette section présente plus en détails ce qu'est appelé "géométrie paramétrée". Une analyse en est faite ensuite.

2.2.1 Présentation

D'un point de vue général – c'est-à-dire sans se restreindre à la modélisation géométrique – un système est dit paramétrique s'il résout un problème de façon séquentielle où chaque sous-problème peut être vu comme un système d'équations de la forme $G(P) = X$, avec P les paramètres et X les inconnues. Autrement dit, un problème paramétrique est un problème qui peut être décomposé en un ensemble de sous-problèmes tel que chaque sous-problème place un seul élément par rapport aux autres éléments déjà en place [Hoffmann *et al.* 2001–1].

Pour un modeleur géométrique 3D, chaque étape de construction est un problème paramétrique. En effet, que ce soit pour les modèles de constructions procédurales (Cf. sous-section 1.3.2) ou les modèles de constructions par modifications de surfaces (Cf. sous-section 1.3.5), chaque étape peut être vue comme l'évaluation "d'une boîte noire", avec en entrée un ensemble de paramètres et d'objets géométriques et en sortie le résultat. Pour définir un modeleur paramétrique 3D, il suffit d'enregistrer l'historique de construction et de permettre à l'utilisateur la modification des différents paramètres. Suite à une modification, le système se contente de rejouer les différentes étapes de construction pour mettre à jour la géométrie.

2.2.2 Analyse

Cette approche reçoit un grand succès dans le monde de la CAO pour la modélisation volumique. En effet, tous les logiciels industriels l'ont adoptée dans leur modèleur volumique 3D.

La raison de la réussite commerciale de cette approche est très simple : c'est la *seule* approche existante en 3D, mais elle a de nombreux inconvénients qui limitent les possibilités de l'utilisateur.

En effet, le paramétrage est fonction des étapes de construction et des paramètres descripteurs de la géométrie générée. L'historique de construction est donc très important. Afin de changer les paramètres descripteurs de la géométrie, l'utilisateur est contraint de reconstruire localement (si possible) la pièce. Pour utiliser un système paramétrique d'une façon efficace, l'utilisateur doit anticiper les modifications qui pourront être réalisées [Chung *et al.* 1989]. Il est donc contraint de penser la construction de la géométrie en fonction du paramétrage souhaité. Ceci lui impose de connaître l'objet fini. Dans une approche de conception et même de re-conception, l'utilisateur découvre son mécanisme au fur et à mesure qu'il le dessine.

Un problème paramétrique doit être parfaitement contraint, c'est-à-dire que le problème est défini par un ensemble nécessaire et suffisant de contraintes. Or, l'utilisateur réalise un objet progressivement, tout en précisant localement la géométrie. Son objet n'est alors bien-contraint qu'en fin de conception. Avec une approche paramétrique, il est dans l'incapacité de travailler ainsi.

Les contraintes ne peuvent pas être cycliques, c'est-à-dire que chaque étape de résolution ne doit utiliser qu'une et une seule contrainte. Par exemple, un triangle $A B C$ (Figure 2.1) où les trois longueurs d_1 , d_2 et d_3 sont contraintes est un problème cyclique. En effet, les points A et B sont connus grâce à la longueur d_1 . Pour positionner C il faut utiliser d_2 et d_3 .

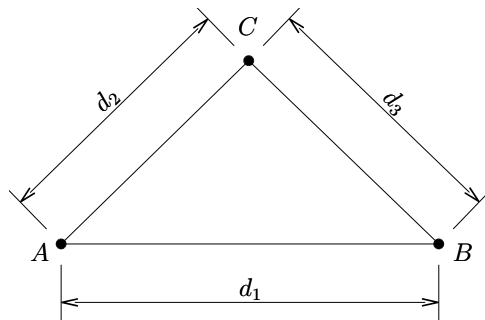


FIG. 2.1: Un triangle spécifié par un ensemble de contraintes cycliques

Une approche paramétrique ne capte pas l'intention de l'utilisateur. En effet, l'utilisateur est contraint par les différents outils de construction et leur paramétrage pré-défini. Or, ce point est important pour un système de CAO. En

cas de multiples solutions, cette information permet au système de choisir les solutions les plus proches de l'attente de l'utilisateur, mais cette intention ne peut pas être captée dans une approche paramétrique.

2.3 Modélisation géométrique variationnelle

La modélisation variationnelle est l'approche attendue par les utilisateurs. Elle est déjà présente dans les modules 2D, mais, dans certains cas, son utilisation n'est déjà pas évidente. Pour que cette approche soit intégrée dans les modules de conception de pièces, de nombreux problèmes théoriques doivent encore être résolus. Les difficultés en 2D laissent à prévoir que l'utilisation en 3D nécessitera de nombreux outils de contrôle, d'analyse et d'aide à la spécification.

Une présentation débute cette section, suivie d'une analyse. En conclusion seront introduits les différents problèmes à résoudre avant de développer une telle approche en 3D.

2.3.1 Présentation

Pour reprendre la définition précédente, la spécification d'un problème sera dite variationnelle si elle n'est pas paramétrique. Autrement dit, c'est un problème où certaines entités doivent être placées simultanément en relation les unes par rapport aux autres [Hoffmann *et al.* 2001–1].

La géométrie variationnelle est apparue dès que le choix des spécifications n'a plus été imposé par les systèmes de CAO. À partir de ce moment, le plan de résolution n'est plus déductible de l'historique de construction mais déduit par un module spécifique.

2.3.2 Analyse

Avec une approche variationnelle, l'utilisateur est libéré des contraintes techniques de l'approche paramétrique. Il peut se concentrer pleinement à sa tâche première de création. La contrainte majeure – l'historique de construction – n'a plus de signification. Il n'a donc plus besoin de se préoccuper des étapes de construction. Le schéma de contraintes n'est plus imposé. De plus, celles-ci peuvent être de nature différente (dimensionnelles, d'ingénierie) et couplées, permettant ainsi de définir d'une façon optimale les contraintes fonctionnelles, c'est-à-dire un ensemble de contraintes significatives pour le fonctionnement, l'utilisation de l'objet.

La géométrie variationnelle a aussi simplifié les modules d'esquisse. Comme l'utilisateur n'a plus besoin de réaliser parfaitement un dessin, il n'a plus besoin de disposer de toutes les méthodes de construction requises en l'absence de tels

modèles. Maintenant, il réalise une esquisse avec des outils élémentaires. Ensuite, ou pendant la construction, il précise la forme souhaitée par un ensemble de contraintes dimensionnelles, logiques, et d'ingénierie entre les différents éléments géométriques. Le problème n'a pas besoin d'être contraint par un ensemble suffisant de spécifications. L'utilisateur peut alors préciser petit à petit son objet et observer les conséquences de ses spécifications par des mises à jour successives. Le système gère la mise à jour de la géométrie pour garder cohérent l'ensemble des spécifications.

Du point de vue utilisation, cette approche semble supprimer tous les problèmes liés à la modélisation paramétrique, mais une nouvelle difficulté, et pas des moindres, apparaît : le choix de l'ensemble des contraintes. Avec la modélisation paramétrique, le système impose son choix de contraintes. Le problème est donc constamment bien défini. Avec une approche variationnelle, l'utilisateur choisit ses contraintes. Il peut les modifier à tout moment. Afin de bien transcrire son cahier des charges, il utilise des relations d'ingénierie. Finalement, en essayant d'obtenir un schéma de contraintes fonctionnelles, il définit un problème difficilement analysable : certaines zones sont sous ou sur-contraintes, d'autres sont définies par des schémas trop imbriqués. Une analyse "visuelle" d'un problème peut devenir rapidement complexe, voire impossible.

Cette approche a, d'un côté supprimé une grande partie des problèmes de la modélisation paramétrique et de l'autre, accentué les difficultés de la modélisation sous contraintes. Mais au final, elle apporte beaucoup. En effet, un objet est facilement modifiable par une personne autre que le concepteur puisqu'il est défini par ses spécifications fonctionnelles. Dans la même idée, une famille de mécanismes est modélisable par un unique modèle défini par de véritables paramètres fonctionnels.

2.3.3 Problèmes de mise en place

Un problème variationnel n'est pas résolvable d'une façon séquentielle, un module dédié doit alors gérer cette résolution. Ce module reçoit en entrée une instance initiale de la géométrie et toutes les spécifications (géométriques, dimensionnelles, logiques et d'ingénierie). À partir de ces données, il doit proposer à l'utilisateur la solution attendue. Or, un problème, même bien contraint, admet souvent plusieurs solutions. Le système doit être en mesure de les trouver afin d'offrir les différentes possibilités, mais comme l'étape de spécification est difficile, le système doit au préalable assister l'utilisateur. L'indication de l'état de contraintes local est un exemple d'aide précieuse. Sur les mêmes idées, en cas d'échec de la résolution, le système doit offrir un diagnostic pertinent sur les causes. Dès que des problèmes géométriques 3D sont traités, d'autres difficultés apparaissent. La spécification d'un objet volumique consiste à positionner les éléments surfaciques

les uns par rapport aux autres. L'objet reçoit une contrainte supplémentaire. Il doit conserver une réalité physique : par exemple, la surface de l'objet ne doit pas être en auto-intersection. En 3D, le système doit gérer les problèmes de cohérence volumique et donc, topologiques.

2.3.4 Conclusion

La modélisation variationnelle peut apporter beaucoup aux systèmes de CAO. Les modules 2D existants offrent déjà une aide précieuse aux utilisateurs. Les premiers modules 3D apparaissent mais ces modules ne gèrent pour l'instant que des problèmes simples de positionnement 3D. Une fois que ces modules seront aussi performants en 3D qu'ils le sont actuellement en 2D, les réflexions pourront porter sur la gestion de la topologie. Pour l'instant, il semblerait que seule l'équipe de Brüderlin ait traité ce problème [Döring *et al.* 1998].

2.4 Synthèse

La modélisation géométrique sous contraintes est maintenant bien intégrée dans les systèmes de CAO. Les modules 2D proposent des approches variationnelles alors que les modules 3D ne disposent que d'approches paramétriques. Dans l'ensemble, les systèmes actuels sont fortement limités aux contraintes géométriques. Ceci est dû aux limites des modules de résolution.

Le chapitre suivant présente un état de l'art de ces modules afin de comprendre et corriger leurs différentes limites.

Chapitre 3

Les solveurs géométriques

Alors que le chapitre précédent a présenté les deux familles de modélisation sous contraintes, celui-ci va se concentrer sur le module nécessaire à la modélisation variationnelle c'est-à-dire, le solveur géométrique. L'introduction présente rapidement les différentes applications utilisant un solveur géométrique. Ensuite, les différentes solutions proposées dans la littérature seront décrites. Enfin, ce chapitre aborde les premiers solveurs géométriques traitant des problèmes d'ingénierie.

3.1 Les différentes applications

De nos jours, la résolution de problèmes géométriques apparaît dans de nombreuses applications dans des domaines aussi divers que la biologie, la chimie, l'enseignement, ou la mécanique.

Par exemple en chimie, les algorithmes venant de la géométrie sont appliqués à la recherche de différentes configurations moléculaires. Dans ce cas, la difficulté est de trouver toutes les configurations spatiales possibles. Un exemple de travaux dans ce domaine est présenté dans le document d'Emiris [Emiris *et al.* 1996].

D'autres travaux ont été consacrés à l'obtention d'outils d'assistance pour l'enseignement de la géométrie. Les travaux de Bouhineau [Bouhineau 1997] et de Channac [Channac 1999] ont permis d'obtenir un système déclaratif de géométrie dynamique. Cet outil permet aux élèves de collège ou lycée de construire une figure géométrique répondant à un énoncé et de l'animer à l'aide de la souris tout en respectant continuellement les propriétés définies.

Au niveau des applications graphiques, les solveurs géométriques sont utilisés pour des problèmes de gestion d'espace, de génération d'interfaces [Freeman-Benson *et al.* 1990], [Sannella 1994], d'animation, de simulation, etc.

Une présentation plus détaillée de ces différentes applications a été réalisée par Kwaiter[Kwaiter *et al.* 1998].

En conception, les solveurs sont utilisés pour mettre à jour une esquisse définie par un ensemble de contraintes dimensionnelles. La figure 3.1 présente deux exemples simples mais représentatifs des problèmes à traiter.

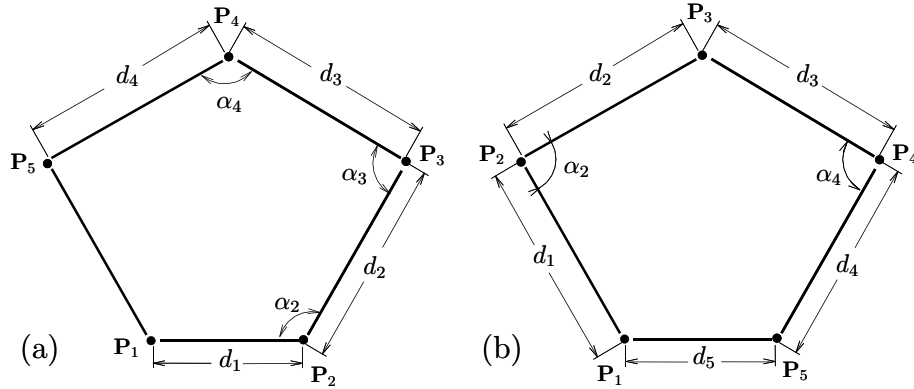


FIG. 3.1: Deux exemples de géométrie sous contraintes. L'exemple b a un schéma de contraintes cyclique.

Afin de répondre à ce besoin, différentes approches ont été proposées. La section suivante présente l'approche la plus élémentaire du point de vue formulation du problème. Elle consiste en la résolution du système obtenu par une mise en équation "triviale". La troisième partie de ce chapitre décrit la stratégie de résolution la plus développée. Son principe est d'analyser le problème afin de le décomposer en sous-problèmes plus facilement et rapidement résolubles. Enfin, la quatrième partie introduit quelques solutions de solveurs d'ingénierie.

3.2 Approches algébriques

Le principe de ces premières méthodes est de poser le problème sous la forme d'un ensemble d'équations et de résoudre, non plus un problème géométrique, mais le système obtenu. Une analyse des méthodes de mise en équations est présentée, suivie de différentes techniques de résolution.

3.2.1 Mise en équations d'un problème

Quelque soit le logiciel, les différents objets géométriques sont définis dans un repère cartésien. Par exemple, un point est défini par ses deux (ou trois) coordonnées. Il a donc paru naturel de choisir pour inconnues du problème, les paramètres descripteurs des différents éléments dans le repère global. La mise en équations devient alors triviale à partir du moment où chaque contrainte

est associée à une équation. Par exemple, Light propose comme équations [Light *et al.* 1982] :

$$\begin{aligned}
 & - (x_2 - x_1)^2 + (y_2 - y_1)^2 - d^2 = 0, \text{ pour la distance } d \text{ entre les points } P_1 \text{ et } P_2; \\
 & - \frac{((x_3 - x_2)(y_1 - y_2) - (y_3 - y_2)(x_1 - x_2))^2}{(x_3 - x_2)^2 + (y_3 - y_2)^2} - d^2 = 0, \text{ pour la distance } d \\
 & \text{entre le point } P_1 \text{ et la droite définie par les points } \{P_2, P_3\}; \\
 & - \frac{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)}{(x_1 - x_2)(x_3 - x_4) + (y_1 - y_2)(y_3 - y_4)} - \tan(\alpha) = 0, \text{ pour l'angle } \alpha \text{ entre} \\
 & \text{les droites définies par } \{P_1, P_2\} \text{ et } \{P_3, P_4\}.
 \end{aligned}$$

Afin d'illustrer cette étape, le problème de la figure 3.1-a est traité. C'est un exemple où quatre longueurs et trois angles sont spécifiés. Pour résoudre ce problème, il suffit de résoudre le système (3.1). Ce système est "bien contraint" à une rotation et à une translation près. Il suffit d'imposer la valeur de trois paramètres, par exemple $x_1 = 0, y_1 = 0, x_2 = 0$, pour fixer la géométrie dans le plan.

$$\left\{ \begin{array}{l}
 (x_2 - x_1)^2 + (y_2 - y_1)^2 = d_1^2 \\
 (x_3 - x_2)^2 + (y_3 - y_2)^2 = d_2^2 \\
 (x_4 - x_3)^2 + (y_4 - y_3)^2 = d_3^2 \\
 (x_5 - x_4)^2 + (y_5 - y_4)^2 = d_4^2 \\
 \frac{(x_2 - x_1)(y_3 - y_2) - (y_2 - y_1)(x_3 - x_2)}{(x_2 - x_1)(x_3 - x_2) + (y_2 - y_1)(y_3 - y_2)} - \tan(\alpha_2) = 0 \\
 \frac{(x_3 - x_2)(y_4 - y_3) - (y_3 - y_2)(x_4 - x_3)}{(x_3 - x_2)(x_4 - x_3) + (y_3 - y_2)(y_4 - y_3)} - \tan(\alpha_3) = 0 \\
 \frac{(x_4 - x_3)(y_5 - y_4) - (y_4 - y_3)(x_5 - x_4)}{(x_4 - x_3)(x_5 - x_4) + (y_4 - y_3)(y_5 - y_4)} - \tan(\alpha_4) = 0 \\
 \frac{(x_3 - x_2)(y_4 - y_3) - (y_3 - y_2)(x_4 - x_3)}{(x_3 - x_2)(x_4 - x_3) + (y_3 - y_2)(y_4 - y_3)} - \tan(\alpha_3) = 0
 \end{array} \right. \quad (3.1)$$

Il est bien connu que la qualité de la mise en équations d'un problème est primordiale pour le résoudre simplement. Dans le domaine des solveurs géométriques, ce principe semble avoir été oublié. Très peu d'auteurs ont travaillé sur ce point. Prenons l'exemple de Serré [Serré 2000] qui a proposé des mises en équations basées sur deux modélisations déclaratives de la géométrie. L'idée est de positionner les différents éléments les uns par rapport aux autres et non pas dans un repère absolu global. Ses systèmes originaux ne vont, pour l'instant, pas vers une simplification : de 5 000 à 4 000 000 équations imbriquées et fortement redondantes sont nécessaires pour positionner une centaine de points "bien contraints". La mise en équations ne semble donc pas être le but de ces travaux.

L'exemple du système (3.1) du problème de la figure 3.1-a illustre les conséquences d'une mauvaise mise en équations. Cette fois-ci, ce sont les va-

riables x_1, y_1, x_5 qui sont spécifiées. Le graphe biparti (Cf. sous-section 3.2.4) de cet exemple est le graphe de la figure 3.2 dans lequel l'ensemble des arcs en traits épais illustre la forte imbrication des équations. Il est donc nécessaire de résoudre ce système globalement.

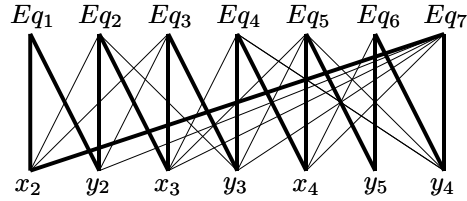


FIG. 3.2: Graphe biparti du système d'équations 3.1 avec x_1, y_1 , et x_5 constants

En revanche, la possibilité de décomposer un système en sous-systèmes est un critère intuitif d'une bonne mise en équations. L'idéal est que tous les sous-systèmes soient représentatifs d'une étape de construction de la géométrie "à la règle et au compas". Ces étapes peuvent être observées sur la figure 3.3 qui illustre la construction du problème de la figure 3.1-b. Les trois premières étapes représenteraient la construction des parties rigides, et la dernière, l'assemblage de ces trois parties.

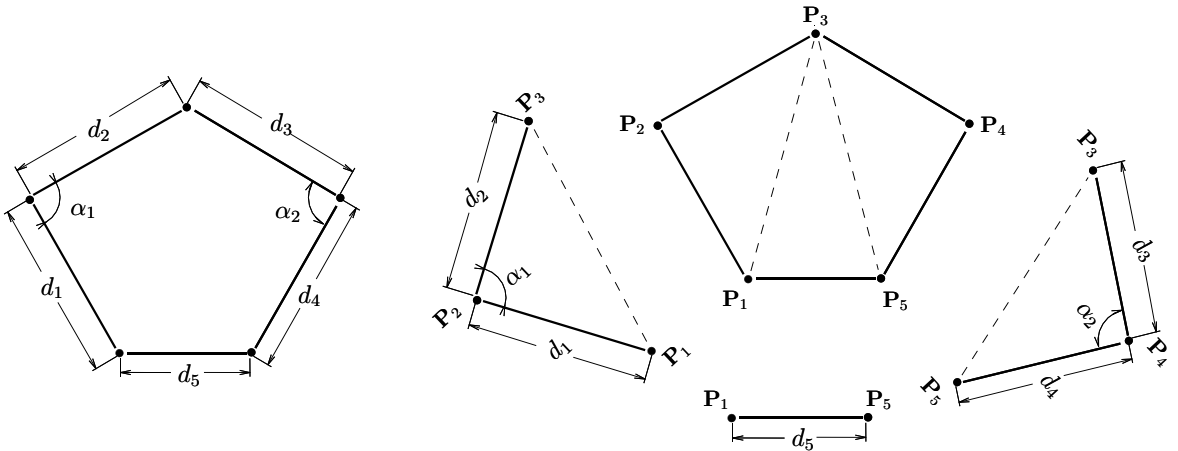


FIG. 3.3: Étapes de résolution de la figure 3.1-b

La résolution du problème est ensuite déléguée à un module de résolution de systèmes d'équations. C'est à ce niveau que les approches se divisent en deux grandes familles : les approches basées sur une résolution symbolique et les approches basées sur une résolution numérique. Le choix de la méthode de résolution utilisée est fonction du type d'application. Par exemple, pour la démonstration automatique de théorèmes de la géométrie élémentaire, une résolution formelle est nécessaire. Pour une application interactive, une solution numérique est préférée

en raison de la rapidité de sa résolution. Quelle que soit la méthode choisie, une décomposition préalable en sous-systèmes indépendants est préférable pour des raisons de performance. Ce dernier point sera présenté dans la dernière partie de cette section.

3.2.2 Méthodes formelles

De nombreuses méthodes symboliques existent, telles que les bases de Gröbner [Chou *et al.* 1987], l'algorithme de décomposition de Wu-Ritt ou d'autres techniques d'élimination. L'intérêt de telles méthodes est d'obtenir un résultat exact ou de permettre une discussion sur des conditions de non dégénérescence. En se basant sur ces méthodes, Gao [Gao *et al.* 1998] propose des algorithmes permettant d'analyser l'état de contraintes d'un problème et de définir s'il est possible ou non de le dessiner à la règle et au compas. Quant à Nguyen [Nguyen *et al.* 1996], il préfère utiliser le système de résolution formelle "MAPLE". Il cherche à obtenir toutes les solutions dans le cas d'un problème bien contraint ou une expression liant les différents paramètres dans le cas d'un problème sous-contraint.

Les gros points faibles de ces méthodes sont la lenteur, la consommation de mémoire et un temps de calcul qui évolue de façon exponentielle en fonction du nombre de paramètres. Elles ne sont donc destinées qu'à des petits problèmes. En effet, Lamure [Lamure *et al.* 1995] rappelle les résultats de Lazard qui a montré qu'il est impossible de calculer les bases de Gröbner d'un système polynomial irréductible de degré 2 en 10 inconnues et variables. Ces méthodes sont donc plus particulièrement adaptées à la démonstration automatique de théorèmes de la géométrie élémentaire qu'à la CAO, où, dans un problème industriel, le nombre d'inconnues se compte généralement en milliers.

3.2.3 Méthodes numériques

Les logiciels de CAO sont des applications interactives. Ils nécessitent donc des modules de résolution très rapides. De plus, un résultat exact n'est pas utile. Une certaine précision est suffisante. Pour ces raisons, la méthode de résolution des systèmes d'équations non-linéaires est généralement basée sur l'algorithme de Newton-Raphson ou une de ses variantes [Lin *et al.* 1981], [Nelson 1985], [Serrano *et al.* 1986].

Comme beaucoup de méthodes numériques, ces algorithmes nécessitent une approximation initiale. Cette contrainte n'est pas gênante pour les applications de CAO puisqu'elle est obtenue à partir de l'esquisse définie par l'utilisateur. Mais dès qu'une solution initiale est nécessaire, le résultat en est fonction. Pour certains algorithmes, la solution finale reste *proche* de la solution initiale. Par exemple, les méthodes homotopiques sont reconnues pour avoir un comportement

naturel, alors que la méthode de Newton-Raphson a un comportement chaotique et trop souvent ne converge pas. Pour illustrer ceci, la figure 3.4 reprend l'exemple de Lamure [Lamure *et al.* 1995]. Elle représente les bassins d'attraction pour la résolution dans le corps des complexes de l'équation $z^3 - 1 = 0$ par la méthode de Newton-Raphson (a) et l'homotopie (b). Pour la méthode de Newton-Raphson, les bassins sont fractals, ce qui explique son imprévisible convergence. Alors que dans cet exemple, les bassins d'attraction pour l'homotopie sont les cellules de Voronoï des points solutions. Lamure [Lamure *et al.* 1995] a choisi de privilégier ce comportement intuitif avec, en contre partie, une augmentation du temps de résolution. Ce temps reste du même ordre que celui de Newton-Raphson (c'est à dire en $O(n^3)$ dû en grande partie à l'inversion de la matrice Jacobienne).

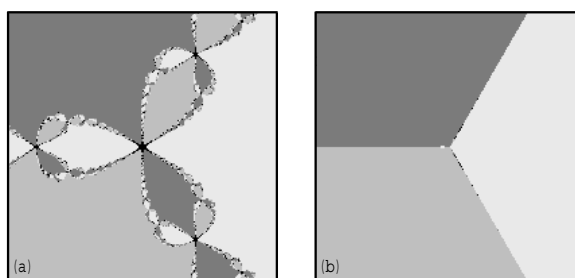


FIG. 3.4: Bassins d'attraction pour les méthodes de Newton-Raphson et homotopie

De nombreuses autres alternatives ont été proposées dans la littérature. Sutherland, par exemple, utilise une méthode de résolution numérique par relaxation [Sutherland 1963]. Dans l'optique de rendre la résolution moins sensible à la valeur initiale et de traiter les problèmes sur et sous-contraints d'une façon plus naturelle et efficace, Ge [Ge *et al.* 1999] a testé l'utilisation de méthodes d'optimisation comme BFGS. Enfin, Durand [Durand 1998], [Durand *et al.* 1999] utilise l'homotopie dans le corps des complexes afin de trouver plusieurs, voire toutes les solutions d'un système.

3.2.4 Méthodes de décomposition

L'utilisation d'une méthode purement algébrique reste limitée aux systèmes de petites dimensions. En effet, les approches symboliques explosent rapidement en temps de calcul, le comportement chaotique et divergent de Newton s'intensifie et le temps de résolution des méthodes numériques dépasse les limites pour rester interactif. Ce ne sont donc pas des méthodes viables pour les problèmes industriels.

Comme ces méthodes ne "peuvent" pas résoudre un problème globalement, certains auteurs ont opté pour une résolution progressive. La première approche proposée est basée sur une méthode de propagation. La seconde est une généralisation de la première. Elle est basée sur la théorie des graphes bipartis.

Méthodes par propagation

Le système d'équations est posé sous forme de graphe dans lequel les nœuds représentent les équations (ou relations), les variables et les constantes. Les arcs lient les variables et les constantes aux équations dans lesquelles elles apparaissent. L'étape importante est d'orienter le graphe afin de satisfaire toutes les équations.

Basés sur cette approche, de nombreux algorithmes destinés essentiellement à la spécification d'interfaces graphiques ont été présentés, comme la famille des algorithmes *Blue* [Freeman-Benson *et al.* 1990] [Sannella 1994]. Dans ces approches, les nœuds contiennent des “méthodes” et non pas des équations mais le principe ne change pas.

Ces algorithmes ne sont destinés qu'à des problèmes géométriques simples, c'est-à-dire sans cycle. En d'autres termes, chaque relation du graphe doit être résolue seule. La figure 3.1, par exemple, ne peut pas être traitée par cette approche. Malgré ces limitations, ces méthodes sont particulièrement performantes dans les domaines de l'animation, des interfaces graphiques et des applications interactives. En effet, dans ces domaines, les schémas de contraintes sont sans cycle, et la résolution doit être particulièrement rapide et peut être demandée de nombreuses fois [Kwaiter *et al.* 1998].

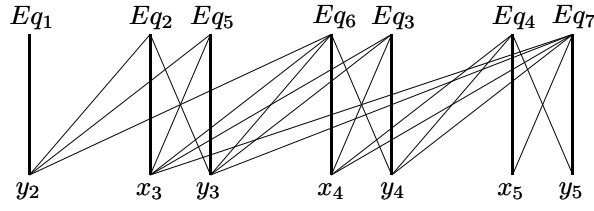
Le chapitre suivant présente l'utilisation de graphes bipartis qui pourraient être une bonne solution pour pallier les limitations de ces méthodes.

Les graphes bipartis

Un graphe biparti est un graphe dont l'ensemble des sommets peut être partitionné en deux classes, de sorte que deux sommets de la même classe ne soient jamais adjacents. Ce graphe est intéressant pour modéliser un système d'équations : le premier ensemble de nœuds regroupe les équations et le second les variables. Il est finalement identique à celui utilisé par les méthodes de propagation. Les algorithmes associés à cette famille de graphes [Murota 1987] permettent de décomposer le système d'équations en sous-systèmes indépendants, et cette étape de décomposition est celle qui permet de regrouper les équations appartenant à un même cycle. Le plan de résolution est obtenu par des algorithmes similaires aux algorithmes de propagation.

Cette famille de graphes est présentée en détails en annexe C.6 puisqu'elle définit un des principaux maillons de l'approche présentée dans ce manuscrit.

Pour illustrer l'intérêt des graphes bipartis, l'exemple “a” de la figure 3.1 est repris avec $x_1 = 0$, $y_1 = 0$, $x_2 = 0$. La figure 3.5 représente le graphe biparti du système d'équations (3.1). Cette analyse structurelle décompose le système en quatre sous-systèmes indépendants, à résoudre séquentiellement du système (3.2) au système (3.5) :


 FIG. 3.5: Graphe biparti du système d'équations 3.1 avec x_1 , y_1 , et x_2 constants

$$\{ (x_2 - x_1)^2 + (y_2 - y_1)^2 = d_1^2 \quad ; \quad (3.2)$$

$$\left\{ \begin{array}{l} (x_3 - x_2)^2 + (y_3 - y_2)^2 = d_2^2 \\ \frac{(x_2 - x_1)(y_3 - y_2) - (y_2 - y_1)(x_3 - x_2)}{(x_2 - x_1)(x_3 - x_2) + (y_2 - y_1)(y_3 - y_2)} - \tan(\alpha_2) = 0 \end{array} \right. \quad ; \quad (3.3)$$

$$\left\{ \begin{array}{l} (x_4 - x_3)^2 + (y_4 - y_3)^2 = d_3^2 \\ \frac{(x_3 - x_2)(y_4 - y_3) - (y_3 - y_2)(x_4 - x_3)}{(x_3 - x_2)(x_4 - x_3) + (y_3 - y_2)(y_4 - y_3)} - \tan(\alpha_3) = 0 \end{array} \right. \quad ; \quad (3.4)$$

$$\left\{ \begin{array}{l} (x_5 - x_4)^2 + (y_5 - y_4)^2 = d_4^2 \\ \frac{(x_4 - x_3)(y_5 - y_4) - (y_4 - y_3)(x_5 - x_4)}{(x_4 - x_3)(x_5 - x_4) + (y_4 - y_3)(y_5 - y_4)} - \tan(\alpha_4) = 0 \end{array} \right. \quad . \quad (3.5)$$

Cette approche est proposée par de nombreux auteurs [Serrano 1991], [Ait-Aoudia *et al.* 1993], [Sridhar *et al.* 1996], [Mathis 1997], [Lamure *et al.* 1998], mais son utilité est fortement dépendante de la qualité du système à décomposer (Cf. figure 3.2).

3.2.5 Conclusion

L'utilisation d'une méthode purement algébrique reste limitée aux problèmes de petites dimensions. En effet, les méthodes de résolution formelle ont une complexité qui évolue de façon exponentielle. Les méthodes itératives, qui sont beaucoup plus rapides, ont tout de même une complexité qui évolue rapidement ($O(n^3)$). La section 3.2.4 a montré que les outils de décomposition de systèmes en sous-systèmes semblent corriger le point faible de ces méthodes. L'analyse de cette approche a mis en évidence l'étape la plus importante : la mise en équations du problème. Or, elle n'a suscité que peu d'intérêt scientifique. Une approche qui semble beaucoup plus naturelle a été préférée : dans un premier temps, une analyse géométrique est réalisée afin de décomposer le problème en sous-problèmes. La résolution ne se fait que lors de la seconde phase au niveau des sous-problèmes. Cette approche sera exposée dans le chapitre suivant.

Quoi qu'il en soit, ces solutions sont utilisées par tous les solveurs industriels. En effet, elles apparaissent comme de très bonnes alternatives lorsque les

méthodes basées sur les techniques de décomposition géométrique échouent. Les approches géométriques auront généralement suffisamment réduit le problème pour que les performances de ces premières méthodes redeviennent acceptables et dans le cas contraire, la lenteur est préférable à l'échec, ...

3.3 Méthodes géométriques

Une grande partie des solveurs géométriques se base sur le fait que la plupart des problèmes industriels peuvent être construits à partir des outils classiques du dessin technique, à savoir règle, compas et rapporteur. Partant de cette remarque, le principe de ces solveurs est de décomposer les problèmes en problèmes plus simples pour lesquels des méthodes de construction sont connues. Deux grandes écoles se sont distinguées : la première est issue de l'intelligence artificielle et la seconde utilise principalement la théorie des graphes.

Parallèlement à ces deux mouvements, un troisième s'appuie sur des principes plutôt liés à l'étude des mécanismes. En effet, l'idée de cette approche est d'étudier les degrés de liberté de la géométrie, toujours dans le but d'identifier un plan de résolution.

3.3.1 Les systèmes à bases de connaissances

Le principe de base des systèmes experts est de définir, par un raisonnement logique, la construction d'une figure répondant à un ensemble de contraintes. Cette démarche se prête particulièrement bien au problème géométrique 2D. En effet, un faible nombre de règles permet de formaliser les étapes de construction réalisables avec les outils classiques du dessinateur. Par conséquent, cet ensemble de règles permet de résoudre la plupart des problèmes de la CAO.

Un problème à résoudre est à définir dans une base de faits. À partir d'un ensemble contenant des propriétés géométriques et les règles de construction, le système expert définira un plan de construction.

Ce principe est, en théorie, relativement simple. De plus, il ne semble, à première vue, pas exister de limites. En effet, si un cas n'est pas traité, il suffit d'ajouter à la base de règle, la "bonne" règle. Et comme les règles sont des formulations logiques : *"si telle condition est respectée alors faire..."*, elles peuvent être définies entre des objets géométriques et non géométriques. Ainsi, il semble concevable de définir un solveur générique sachant traiter un grand nombre de problèmes de différents types.

En pratique, ce n'est pas si simple et ce, toujours pour des raisons de performances. Plus il y a de règles, plus la recherche de la bonne règle à appliquer prend du temps. La difficulté de ces méthodes est donc de trouver un bon compromis

entre l'ensemble des règles qui définit le champ des problèmes résolubles et le temps de résolution.

Utilisation d'un langage de programmation déclarative

Une des premières solutions basée sur un système expert a été proposée par Brüderlin [Brüderlin 1986]. Elle utilise le langage de programmation *Prolog*. Ce langage dispose d'un moteur d'inférences avec une stratégie de chaînage arrière pour la démonstration d'un but. Ce module accepte les contraintes classiques (distance entre deux points, distance d'un point à une droite, angle entre deux droites, etc.), des contraintes liées à un repère (coordonnées d'un point, direction d'une droite) mais aussi des contraintes originales comme l'égalité de deux angles ou deux longueurs. Pour assurer une terminaison de la méthode et un plan de construction cohérent, il utilise l'algorithme de "Knuth et Bendix". Cet algorithme est basé sur des méthodes de réécriture. Le résultat obtenu est un ensemble canonique de règles.

Une description simple de la géométrie

Verroust propose une solution pour résoudre les problèmes définis par deux contraintes élémentaires [Verroust *et al.* 1992]. Ces contraintes sont la distance entre deux points et l'angle entre deux droites.

Cette approche utilise deux types d'ensembles. Le premier appelé "CA" regroupe les segments définis mutuellement par un angle. Un "CA" élémentaire contient un angle. L'utilisation de la relation de Chasles permet alors de connaître l'angle entre tout couple de segments d'un même ensemble. Le second appelé "CD", définit un ensemble de points rigides. Le "CD" élémentaire est un couple de points dont la longueur est connue. Pour un "CD" donné, un plan de construction connu permet de positionner tous les points de cet ensemble. Ainsi, il est possible de calculer la distance entre tout couple de points d'un même "CD" ou l'angle entre tout couple de segment d'un même "CD".

Le rôle du moteur d'inférences est alors d'identifier à partir de l'ensemble des "CD", des "CA" et des segments n'appartenant pas à un "CD", des schémas connus. Dès qu'une configuration de référence est identifiée, l'ensemble des éléments est alors regroupé dans un nouveau "CD". La résolution se termine lorsqu'il ne reste qu'un "CD".

Les configurations classiques sont les formes triangulaires. En effet, un triangle est constructible dès que, soit trois longueurs, soit deux longueurs et un angle, soit une longueur et deux angles, sont connus. D'autres configurations ont été définies comme le quadrilatère de la figure 3.6. Ce quadrilatère a deux "CD" opposés et trois angles connus. Ces configurations ne sont pas suffisantes pour traiter tous les problèmes, mais elles permettent déjà d'en résoudre une grande partie, dont

les problèmes résolubles à la “règle et au compas”. Verroust [Verroust *et al.* 1992] démontre qu’il n’existe pas de liste de configurations finie et suffisante. Pour cela, elle présente une famille récursive d’exemples non décomposables (Figure 3.7). Chaque élément de cette famille nécessite la définition d’une procédure de construction particulière.

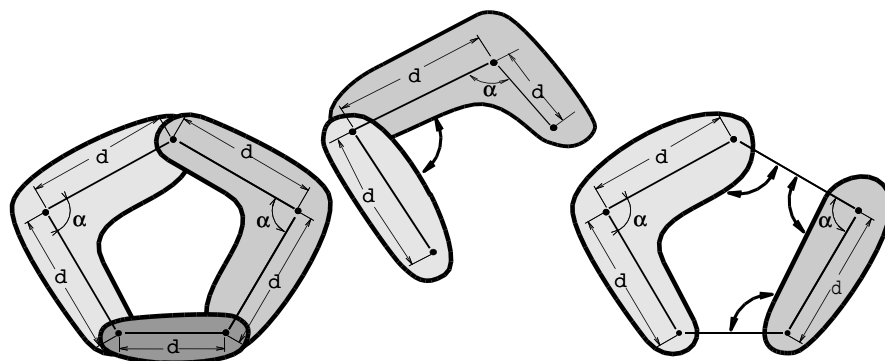


FIG. 3.6: Illustration de quelques configurations permettant la construction d’un “CD”

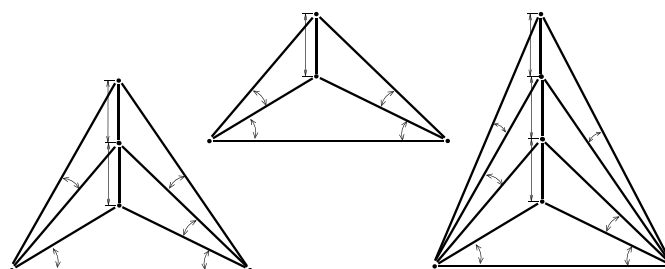


FIG. 3.7: Trois premiers membres d’une famille d’irréductibles

Une autre limite à cette approche est le faible choix des contraintes utilisables. Pour pallier cela, Juan-Arinyo et Mata ont présenté des extensions. Juan-Arinyo [Juan-Arinyo *et al.* 1997] propose l’utilisation d’une troisième spécification : la contrainte de distance d’un point à une droite. Mata expose, dans le rapport [Mata 1997], la gestion des contraintes d’incidence et de tangence.

Contrairement à l’approche de Brüderlin qui est basée sur un moteur d’inférences générique, la structure de données de cette approche la rend purement géométrique.

Approche de Mathis

Mathis propose une solution construite autour d’un univers géométrique formel. À cet univers est associé un ensemble d’axiomes traduisant les règles typiques entre les différents objets géométriques. Le mécanisme de raisonnement appelé

modus ponens est fondé sur un principe d'unification. La description de cette approche est détaillée dans [Mathis 1997].

Le principe de cette approche est de rechercher dans un premier temps les plus grandes sous-figures constructibles indépendamment en partant d'une figure minimale définissant un repère. À cette figure sont ajoutés progressivement les éléments du problème à l'aide de règles de construction. Dès cette première étape finie, l'algorithme essaie d'assembler les différentes sous-figures par une recherche de configurations connues. Si la résolution n'est pas terminée, l'algorithme réitère.

Bien que cette approche soit aussi purement géométrique, l'ensemble des axiomes rappelle l'approche de Brüderlin. La notion de *sous-figure* de Mathis semble être une généralisation des "CD" de Verroust, c'est-à-dire des paquets de points rigides.

Limitation

Contrairement aux méthodes algébriques (Cf. section 3.2), ces approches, en raison de leur raisonnement local, sont très robustes tant que le problème est un problème résoluble. En effet, les différentes étapes de construction sont généralement associées à une (ou plusieurs) solution(s) formelle(s). Toutes les solutions du problème global peuvent donc être calculées, et pour la même raison l'inexistence de solution peut être diagnostiquée de façon exacte.

En contrepartie, la complexité de ces solutions est très importante. En effet, Lee [Lee *et al.* 1996] rappelle qu'Aldefeld a prouvé que cette complexité est en $O(n^4)$ avec n le nombre d'éléments géométriques. Pour illustrer cette combinatoire, prenons comme exemple l'approche de Verroust. Cet algorithme doit chercher pour chaque configuration, s'il existe un ensemble de "CD", "CA", et "segment" lui correspondant. Pour le schéma "trois longueurs spécifiées", supposons que l'on ait n ensembles "CD" contenant chacun m points. Trouver cette figure revient à tester tous les triplets de "CD" soit C_n^3 combinaisons possibles. Pour chaque triplet, il faut regarder s'il existe trois points chacun présents dans deux "CD" distincts, soit près de $3m^2$ combinaisons...

3.3.2 Décomposition réalisée par un graphe

La meilleure solution actuelle pour réaliser un solveur géométrique 2D semble être l'utilisation d'un *graphe de contraintes*. Cette solution est utilisée par D-Cubed pour son solveur DCM 2D. Ce solveur très robuste a un quasi-monopole sur le marché des solveurs dédiés à la CAO.

Dans un graphe de contraintes, les entités géométriques telles que les points, ou les droites, sont représentées par les nœuds et les contraintes par les arcs. De par la nature d'un arc, la première limitation de cette approche est que les contraintes

utilisables ne sont que les contraintes liant deux éléments géométriques. La figure 3.8 représente le graphe de l'exemple de la figure 3.1-b.

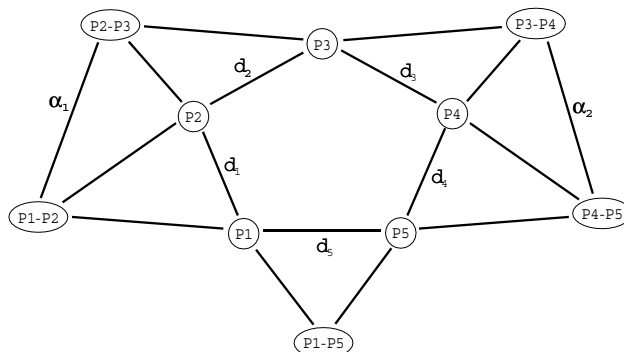


FIG. 3.8: Graphe de contraintes du problème de la figure 3.1-b

Ce graphe est utilisé essentiellement par deux équipes. La première, déjà citée, est l'équipe d'Owen de D-Cubed Ltd, la seconde est l'équipe d'Hoffmann de l'université de Purdue. La différence majeure entre ces deux équipes est l'algorithme d'analyse du graphe de contraintes. Owen propose une analyse descendante alors qu'Hoffmann procède par une approche ascendante. Cette analyse permet d'identifier un plan de résolution, c'est-à-dire un ordonnancement des étapes de construction de la géométrie.

Justification

La justification d'Owen [Owen 1991] sur l'utilisation d'un tel graphe est basée sur un raisonnement de dessinateur : quel que soit le triplet d'éléments géométriques dont chaque couple d'éléments est contraint, il représente une figure géométrique constructible avec les outils classiques. De plus, ces constructions sont résolubles par la résolution algébrique d'un système polynômial de degré 2 [Owen 1991]. Comme la majeure partie des problèmes bien contraints de la CAO sont constructibles à "la règle, au compas et au rapporteur", les graphes correspondants sont donc décomposables en un ensemble de sous-graphes triangulaires (3 nœuds et 3 arcs).

Hoffmann, quant à lui, utilise un critère basé sur l'analyse des degrés de liberté pour justifier son algorithme [Hoffmann *et al.* 1994]. En 2D, les éléments utilisés ont tous deux degrés de liberté. En effet, le point est parfaitement défini par deux coordonnées et la droite par une distance à l'origine et une direction. Les contraintes autorisées fixent un degré de liberté. Deux contraintes suffisent alors à positionner l'objet par rapport à un ensemble d'éléments déjà en place.

Algorithme d'Owen

Owen [Owen 1991] propose une analyse descendante. Le problème géométrique, à travers le graphe de contraintes, est considéré dans sa totalité pour être divisé en sous-graphes d'une façon récursive jusqu'à n'obtenir que des sous-graphes élémentaires. La figure 3.9 décrit les différentes étapes de cet algorithme. A chaque itération, l'algorithme divise le graphe aux *paires d'articulation*. Une paire d'articulation dans un graphe *bi-connecté* est un couple de nœuds tel que leur suppression et celle des arêtes incidentes, sépare le graphe en deux composantes connexes. Un graphe bi-connecté est un graphe où il existe au moins deux chemins reliant tout couple de sommets du graphe. Lors de la scission d'un graphe en deux, les sommets formant la paire d'articulation sont préservés dans les deux sous-graphes et une arête virtuelle est ajoutée entre les deux sommets. Un graphe bi-connecté représente d'un point de vue géométrique, une sous-figure. Le plan de résolution est défini par les étapes de décomposition dans l'ordre inverse.

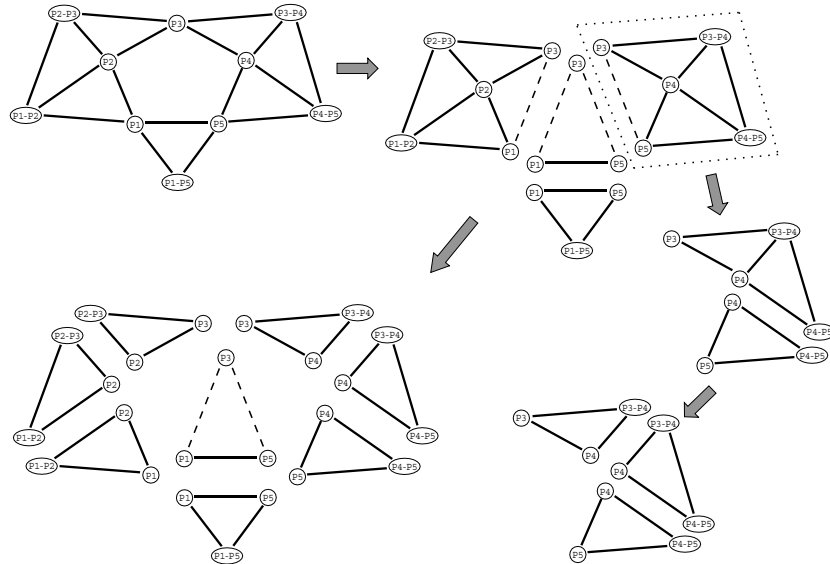


FIG. 3.9: Décomposition, suivant Owen, du graphe de contraintes du problème de la figure 3.1-b

L'inconvénient de cet algorithme est de bloquer dès qu'une figure n'est pas constructible à la règle et au compas. D'un point de vue graphique, un sous-graphe n'est pas décomposable s'il n'a pas de paire d'articulation. De plus, comme cette approche est descendante, elle ne peut pas distinguer les sous-figures constructibles d'une figure non-constructible. La figure 3.10 représente deux exemples non-décomposables par l'algorithme d'Owen. Le premier représente une plateforme de Stewart 2D, c'est-à-dire deux droites positionnées l'une par rapport à l'autre par trois distances point-point. Le second, qui semble être plus simple, n'est, lui aussi, pas constructible avec les outils classiques. Or, ces deux exemples

contiennent des sous-figures constructibles. En effet, pour le premier, les triplets de points peuvent être positionnés sur les droites et pour le second, la droite distante de d_1 du segment P1-P2 peut être construite.

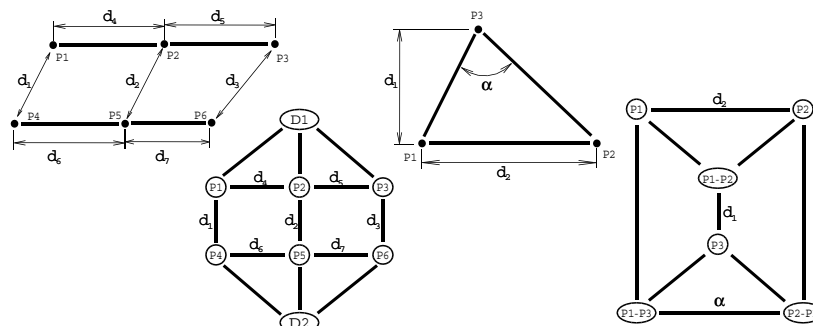


FIG. 3.10: Exemples non-décomposables par l'algorithme d'Owen

Algorithme d'Hoffmann

L'approche développée par l'équipe d'Hoffmann, est proposée en détails dans [Hoffmann *et al.* 1994], [Bouma *et al.* 1995], [Fudos 1995]. La justification théorique est exposée dans [Fudos *et al.* 1996–1]. Leur algorithme consiste, dans un premier temps, à identifier les sous-figures qu'ils appellent *cluster*. L'initialisation est un sous-graphe minimal composé de deux nœuds liés par un arc. Ce sous-graphe définit, d'un point de vue géométrique, un premier repère. Ensuite, l'idée est d'ajouter tous les nœuds liés à ce graphe par deux arcs. Ainsi, cet algorithme identifie les différentes sous-figures résolubles indépendamment (Figure 3.11).

Une deuxième étape consiste à assembler ces figures. Le principe, similaire à l'assemblage des CD de Verroust, est de retrouver des schémas de cluster connus. La configuration élémentaire correspond à l'assemblage de trois cluster (Fig. 3.12-a). Pour les mêmes raisons que pour l'approche de Verroust, cette première configuration n'est pas suffisante. Afin d'étendre les capacités de leur approche, Fudos propose de nouvelles configurations [Fudos *et al.* 1997], comme celle de la figure 3.12-b qui permet de terminer les exemples de la figure 3.10.

Conclusion

Pour le 2D, la solution d'Hoffmann semble toutefois être la plus performante en raison de sa stratégie ascendante. En effet, elle permet de résoudre toutes les sous-figures dont une procédure de construction est connue. Elle délèguera donc, à un autre solveur, la résolution d'un problème simplifié.

Par contre, pour les mêmes raisons que les approches basées sur des systèmes experts, ces deux algorithmes sont limités à quelques familles de problèmes

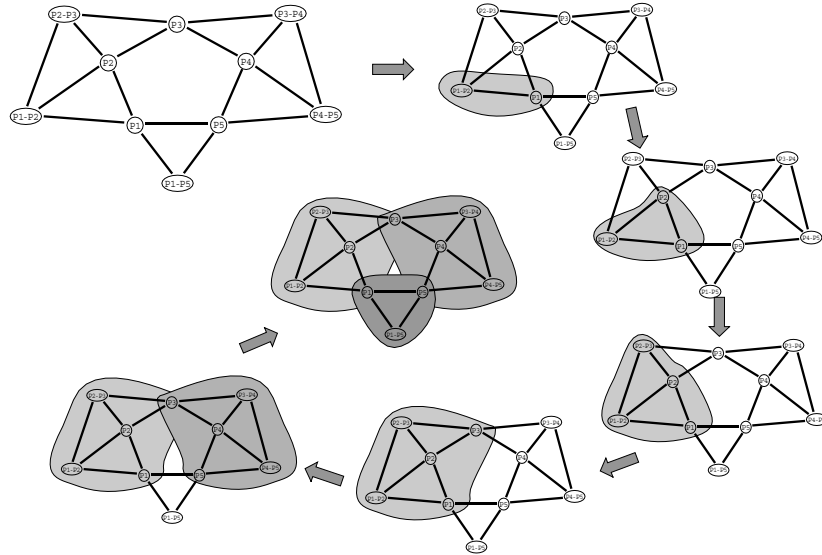


FIG. 3.11: Décomposition suivant Hoffmann du graphe de contraintes du problème de la figure 3.1-b

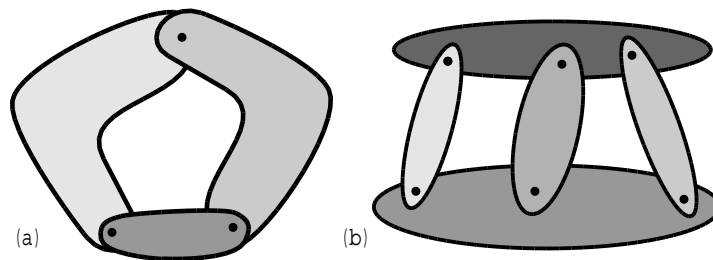


FIG. 3.12: Quelques schémas d'assemblage d'Hoffmann

géométriques 2D. En effet, la nature des algorithmes n'autorise que les éléments à deux degrés de liberté (point, droite, cercle à rayon fixe) et les contraintes de degré 1 liant deux éléments. Or, il est courant d'utiliser des cercles à rayons variables (3 ddl), des arcs de cercles (5 ddl) ou des contraintes de coïncidence point-point et distance droite-droite qui sont des contraintes de degré 2.

L'hétérogénéité des degrés de liberté des éléments 3D est encore plus importante. Il n'existe donc pas une configuration élémentaire en 3D qui permettrait de résoudre une grande partie de cas simples comme l'est le triangle en 2D. Pour ces raisons, les premières réflexions d'Owen [Owen 1996] et d'Hoffmann [Hoffmann 1994] sur la généralisation de l'approche au 3D ne présage pas de solution aussi naturelle. Mais Hoffmann présente tout de même une première solution ne traitant que des points et des plans [Hoffmann 1994]. La configuration élémentaire est alors un tétraèdre.

3.3.3 Décomposition réalisée par une analyse des degrés de liberté

Cette dernière famille de solutions base ses algorithmes sur l'analyse du degré de liberté des figures. Kramer [Kramer 1991] est le premier à proposer une telle approche. Le problème est décrit dans un graphe de contraintes. L'idée principale est alors d'identifier des sous-graphes définissant des parties rigides. Pour chaque sous-figure, un plan de construction est défini. Le sous-graphe correspondant est alors réduit à un seul nœud. Puis, l'algorithme réitère jusqu'à n'obtenir qu'un dernier nœud.

Différents algorithmes ont été proposés pour identifier les parties rigides. Kramer ou Latham [Latham *et al.* 1996] ont proposé d'utiliser un algorithme basé sur le couplage maximal d'un graphe biparti, dont les nœuds sont les entités géométriques d'une part et les contraintes géométriques d'autre part. La limite de cette approche est qu'elle ne décompose pas les problèmes définis par plusieurs parties rigides assemblées.

Lee [Lee *et al.* 1998] utilise un graphe de contraintes colorié, c'est-à-dire que les nœuds et les arcs sont de natures différentes. Le type de nœud est fonction des degrés de liberté de l'entité. Par exemple, une partie rigide est une entité à 3 ddl alors qu'un point n'en a que deux. De même, les contraintes sont de différentes natures. Elles réduisent donc différemment les degrés de liberté des deux entités liées : une coïncidence entre deux points ou une distance entre deux droites réduit le degré de 2 alors qu'une distance entre deux points ou une incidence d'un point sur une droite ne le réduit que de 1. Hoffmann (Cf. sous-section 3.3.2) et Lee utilisent, malgré les limites connues, un algorithme identifiant des configurations dans le graphe.

Afin d'étendre cette approche au 3D et de la rendre générique, Hoffmann

[Hoffmann *et al.* 1997–2] propose d'utiliser un graphe de contraintes pondéré. Ce graphe permet d'enrichir la description d'un problème géométrique. En effet, le poids d'un nœud ou d'un arc représente respectivement les degrés de liberté de l'élément associé et le degré de la contrainte associée. Par exemple en 3D, le point et le plan ont 3 ddl, la droite a 4 ddl et une figure rigide a 6 ddl. Le poids d'une distance point-point est de 1 alors que le poids d'une coïncidence point-point est de 3.

Dans ce graphe $G = (V, E, w)$, une partie rigide est vue comme un sous-graphe *dense*, c'est-à-dire un sous-graphe $A \subseteq G$ tel que :

$$\sum_{e \in A} w(e) - \sum_{v \in A} w(v) > K, \quad (3.6)$$

avec $K = -(D + 1)$. Pour les problèmes 2D, $D = 3$ puisqu'une figure rigide à 3 degrés de liberté (2 translations et une rotation). Pour le 3D, $D = 6$.

Une justification intuitive des graphes denses, par analogie aux systèmes d'équations, est que si le nombre total de degrés de liberté supprimé par des contraintes ($\sum_{v \in A} w(v)$ est supérieur ou égal aux degrés de liberté de la figure géométrique ($\sum_{e \in A} w(e)$) alors le système est rigide. Le problème consiste alors à trouver les plus petits sous-graphes denses. L'algorithme utilisé est présenté dans [Hoffmann *et al.* 1997–2]. Son principe consiste, dans un premier temps, à isoler un sous-graphe dense, puis à trouver dans celui-ci, le sous-graphe dense minimal. Ce sous-graphe minimal assure une résolution locale. Une comparaison des performances de cet algorithme par rapport aux deux familles précédentes est présentée dans [Hoffmann *et al.* 2001–2]. L'avantage principal est le caractère générique de cette approche : il n'y a pas de restriction dans les éléments ou les contraintes, le 2D et le 3D sont traités par les mêmes algorithmes et tous les cas de figure sont analysables.

3.3.4 Conclusion

Avant qu'une méthode générique n'ait été proposée, l'ensemble des solutions a été construit autour du même constat : la plupart des problèmes géométriques sous contraintes sont résolubles à la règle et au compas. Les solutions ont donc toutes les mêmes bases mais ont été réalisées à partir de techniques différentes. On retrouve en particulier des solutions issues de l'intelligence artificielle ou de la théorie des graphes.

Le problème de ces approches est que leur robustesse est fonction de la taille de la famille des configurations utilisées. Plus cette famille est grande, plus la complexité des algorithmes est importante, et plus les temps de calcul augmentent. Inversement, plus ces approches sont rapides, plus l'étendue des problèmes résolus est faible. Pour ces raisons, aucune approche viable n'a vu le jour pour le 3D actuellement.

3.4 Résolution de problèmes d'ingénierie

Les utilisateurs de logiciels de CAO emploient depuis maintenant quelques années les outils de la géométrie variationnelle dans le module d'esquisse 2D. Ils sont donc habitués à spécifier les sections de leurs pièces par un ensemble de contraintes géométriques. Or il est rarement possible d'établir une spécification d'ingénierie pour définir une pièce 3D à ce niveau de définition. Ils attendent donc de véritables modules variationnels 3D. Les résultats obtenus par une analyse des degrés de liberté sur les graphes de contraintes pondérés (Cf. sous-section 3.3.3) promettent une prochaine apparition de modules d'esquisse 3D. L'utilisateur pourra alors contraindre d'une façon réellement fonctionnelle ses objets.

La spécification dimensionnelle est un moyen d'assurer la cohérence des données entre le cahier des charges d'un produit et la solution proposée. Or, l'ensemble des contraintes définies dans ce recueil ne le sont initialement pas sous la forme de contraintes dimensionnelles. Il regroupe surtout des relations entre les différents paramètres du produit (physiques, économiques, marketing, etc.). Or, l'utilisateur souhaiterait définir directement ces contraintes sur la géométrie du produit par l'intermédiaire de relations d'ingénierie, c'est-à-dire des relations logiques ou algébriques entre différents paramètres.

Les solveurs variationnels ne sont, par nature, pas en mesure de répondre à ce besoin. Les chapitres précédents ont montré qu'ils avaient tous un raisonnement fortement géométrique. Or, par définition, une contrainte d'ingénierie lie des paramètres de natures diverses. Elle peut donc prendre toutes les formes possibles. Les approches actuelles basées sur un ensemble de règles ou sur une reconnaissance de configurations, ne sont donc pas en mesure de traiter ces contraintes. Il en résulte que les différentes solutions présentées dans la littérature sont basées sur une coopération entre solveurs, ceci afin de ne pas perdre l'apport des solveurs géométriques, tout en étendant leurs possibilités par l'utilisation – en parallèle – de solveurs numériques.

3.4.1 Un exemple de problème d'ingénierie

Les différents solveurs géométriques présentés définissent une contrainte géométrique comme une relation entre deux éléments géométriques. À partir de cette définition, une contrainte d'ingénierie peut être définie comme une relation liant au moins trois éléments géométriques. Pour généraliser, une contrainte est de type "ingénierie" si elle n'est pas géométrique. Un premier exemple, qui par sa nature pourrait pourtant être considéré comme une contrainte géométrique est l'aire d'une surface.

La figure 3.13 présente un problème d'ingénierie. Sur ce schéma 3.13-a, un ensemble de paramètres est mis en évidence. Ils ne définissent pas de contraintes

géométrique comme c'est généralement le cas. C'est un mécanisme. Au niveau du galet de rayon R_1 une force F est appliquée (Figure 3.13-b). La barre a un poids proportionnel à sa longueur :

$$P = \rho L_1,$$

avec ρ la masse volumique du matériau de la barre. ρ est connue. Le rôle de la force est de maintenir en équilibre le mécanisme. Celui-ci pouvant, sous son propre poids, tomber à la position représentée en pointillés (Figure 3.13-b). Pour des raisons liées au mécanisme, le second galet a un rayon proportionnel à la force appliquée :

$$R_2 = 0.05F.$$

La loi d'équilibre du système peut être définie comme suit :

$$F = \frac{0.5P}{0.73 \tan \alpha}.$$

Le rayon du premier galet, la distance L_2 et la longueur de la barre sont contraints.

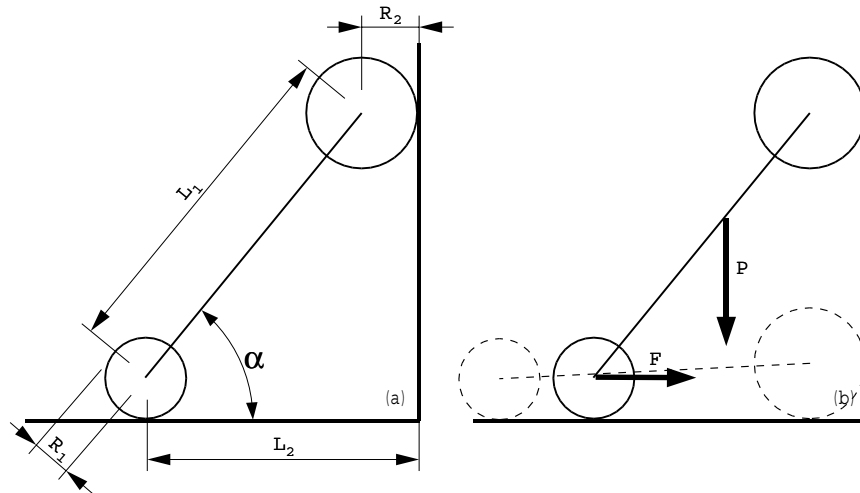


FIG. 3.13: Un exemple de problème d'ingénierie

Cet exemple très simple illustre parfaitement la difficulté à faire collaborer deux solveurs. En effet, la partie purement géométrique est sous contrainte de degré 1 : si l'angle ou le rayon du second galet étaient spécifiés, la géométrie serait parfaitement connue. De même, la partie ingénierie est sous contrainte de degré 1. En effet, il y a trois équations pour quatre inconnues F , P , R_1 , α . Si le rayon ou l'angle étaient connus, la partie ingénierie serait parfaitement résoluble. En conclusion, ce problème nécessite un traitement global.

3.4.2 Une résolution parallèle

Une première solution pour traiter les problèmes d'ingénierie, serait de partager simplement la résolution en deux. Pour cela, Chung [Chung *et al.* 1999] pro-

pose l'algorithme illustré par la figure 3.14. L'idée est très simple. Le problème géométrique est décrit dans un graphe de contraintes (Cf. sous-section 3.3.2). Les parties résolubles par cette approche, les *clusters*, sont identifiées. En revanche, les parties du problème qui ne peuvent pas être résolues (c'est-à-dire les parties sous-contraintes d'un point de vue géométrique et les parties iso-contraintes mais non-traitées par ces algorithmes) sont re-formulées afin d'être analysées avec un graphe biparti (Cf. sous-section 3.2.4), puis traitées par une méthode de résolution algébrique (Cf. section 3.2).

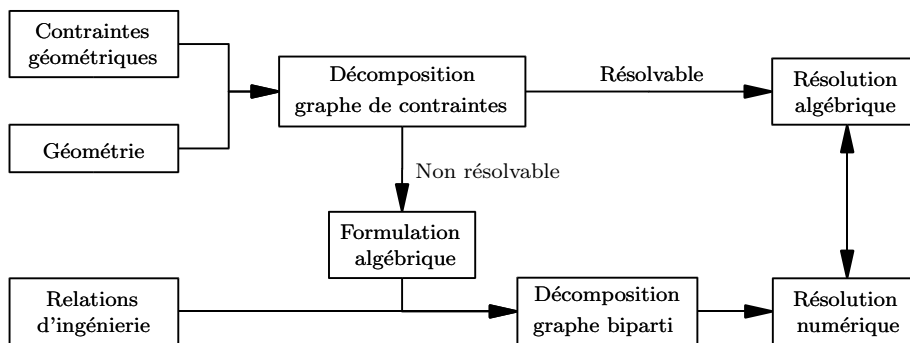


FIG. 3.14: Architecture proposée par Chung

Cet algorithme semble utiliser comme hypothèse simplificatrice que, les problèmes d'ingénierie comme les sous-problèmes non-résolvables apparaissent en fin de résolution. Or, rien n'empêche d'avoir comme première partie à résoudre, un sous-problème d'ingénierie dont la résolution autoriserait à finir par des algorithmes géométriques. Mais, comme dans ce cas le problème n'est initialement pas décomposable par les algorithmes graphiques, il est formulé par un ensemble d'équations et résolu entièrement par des méthodes algébriques, ce qui entraîne les conséquences que l'on connaît.

3.4.3 Un algorithme itératif

Hoffmann et Joan-Arinyo [Hoffmann *et al.* 1997–1] [Juan-Arinyo *et al.* 1999] proposent une solution itérative à la résolution de problèmes d'ingénierie :

- Étape 1 – construire le graphe de contraintes et le graphe biparti ;
- Étape 2 – tant que des sous-figures peuvent être résolues dans le graphe de contraintes, les résoudre ;
- Étape 3 – mettre à jour le graphe biparti en supprimant les variables calculées à l'étape 2 ;
- Étape 4 – décomposer le graphe biparti en sous-systèmes indépendants ;
- Étape 5 – résoudre le premier niveau de sous-systèmes (les sous-systèmes pouvant être résolus seuls) et l'effacer du graphe ;

Étape 6 – mettre à jour le graphe de contraintes en ajoutant les paramètres géométriques calculés en tant que contraintes dimensionnelles ;

Étape 7 – si tout est résolu, fin ; si une inconsistance est identifiée, fin ; si non retourner à l'étape 2.

L'idée de cet algorithme est très simple : tant que de la géométrie peut être résolue, elle l'est. Ensuite, le système d'équations d'ingénierie est analysé afin de trouver l'ensemble des équations à résoudre pour débloquer la partie géométrique. Le processus se réitère jusqu'à la fin de la résolution.

Pour cette approche, la partie ingénierie est bien distincte de la partie géométrie. Les auteurs ne proposent d'écrire dans le graphe biparti, que les équations d'ingénierie. Ceci indique qu'ils supposent que la partie géométrique se résout indépendamment de la partie ingénierie. Il n'est donc pas possible avec cette approche de résoudre le problème présenté à la section 3.4.1 où ingénierie et géométrie sont imbriquées.

3.4.4 Conclusion

Les deux approches présentées donnent l'impression que les contraintes d'ingénierie sont utilisées de façon exceptionnelle. Les solutions proposées ne sont que des premières solutions qui permettent, dans certains cas, d'obtenir une solution. Or, en général, elles ne sont pas suffisamment robustes pour offrir à l'utilisateur des pré et post-traitements. En effet, les analyses sont réalisées par deux modules différents. Chaque module utilise des données différentes. Seuls, ils ne peuvent pas conclure sur l'état de contraintes global puisqu'ils n'analysent qu'une partie du problème. La résolution de problèmes d'ingénierie à l'aide de modules de nature différente ne semble donc pas être une solution pertinente. La question devient alors : "comment écrire la partie géométrique pour qu'elle soit bien traitée par un solveur d'ingénierie ?" ou "comment écrire les contraintes d'ingénierie pour qu'elles soient comprises par le solveur géométrique ?".

3.5 Synthèse

Les différents exposés de ce chapitre ont montré les difficultés à mettre en place un module de résolution. En effet, les solutions purement algébriques ne sont pas satisfaisantes. La raison majeure est le taux d'imbrication du système d'équations traduisant le problème. La principale conséquence est l'inefficacité des algorithmes de décomposition. Dès lors, les diagnostics pré et post-traitement ne peuvent pas être pertinents et la résolution ne peut pas être optimisée.

Les solutions géométriques semblent offrir un meilleur compromis. Elles s'appuient sur une analyse géométrique des problèmes. Ainsi, ils sont décomposés en sous-problèmes plus facilement et rapidement résolubles. Cet examen per-

met aussi d'offrir les différents diagnostics. Or, jusqu'à l'utilisation des graphes pondérés, ces solutions n'étaient pas génériques. Seul, un ensemble de problèmes était décomposable, mais elles permettaient quand même de répondre aux besoins de la CAO, ce qui leur a conféré un bon accueil dans le monde industriel.

Les approches géométriques ne sont pas sans inconvénients. En effet, initialement conçues pour résoudre des problèmes purement géométriques, elles se retrouvent bloquées lorsqu'il s'agit de prendre en compte des contraintes d'ingénierie. Les premières solutions de solveurs d'ingénierie s'appuient sur une coopération entre solveurs géométriques et solveurs numériques. Ces solutions qui ont le mérite d'exister, n'apportent finalement pas grand chose : le champ des problèmes traités est trop faible et aucun diagnostic fiable ne peut être délivré.

Les parties suivantes vont proposer une solution qui aspire à définir un solveur d'ingénierie répondant aux attentes des concepteurs, à savoir :

- un couplage natif de l'ingénierie et de la géométrie ;
- une gestion des solutions multiples ;
- des diagnostics pertinents sur l'état de contraintes du problème ;
- des diagnostics pertinents en cas d'échec de la résolution.

Deuxième partie

Présentation de l'approche
proposée pour la gestion de la
cohérence des spécifications d'un
modèle mécanique et description
du modèle d'information

Chapitre 4

Description générale de l'approche

La première partie de cette thèse a présenté l'état de l'art des outils de maintien et d'analyse de la cohérence des spécifications d'un modèle géométrique de composants mécaniques. Ce manuscrit propose une nouvelle approche répondant à cette problématique. Afin d'avoir un premier aperçu, une description globale est proposée dans ce chapitre. Cette solution sera ensuite détaillée dans les deux parties suivantes.

4.1 Introduction

Au cours du processus de conception, la description d'un produit évolue. Elle est initialement formulée dans un cahier des charges sous la forme d'un ensemble de besoins. Ensuite, elle est décomposée en plusieurs entités fonctionnelles, puis reformulée suivant différents points de vue. Une étape clé de ce processus est la désignation précise du modèle géométrique des différents éléments. Celle-ci est obtenue par un ensemble de spécifications de forme (c'est-à-dire géométriques) dont les dimensions et la position peuvent être contraintes par des relations. Ces liens entre les différents éléments géométriques sont généralement élémentaires. Nous parlerons de contraintes géométriques. Dans les autres cas, nous parlerons de contraintes d'ingénierie. Celles-ci permettent de retranscrire plus fidèlement les liens entre les différents paramètres physiques, économiques, marketing,... et géométriques du produit.

L'utilisation de contraintes facilite les modifications du modèle géométrique. En effet, elles seront réalisées automatiquement par un module spécialisé, mais la réduction d'une intervention humaine dans le processus de modification a un coût. En effet, plus le paramétrage du modèle permet des modifications directes, plus la définition de ce paramétrage est complexe. Des outils sont donc nécessaires

pour seconder le concepteur. La première partie de ce chapitre propose un cahier des charges du module d'une application de CAO permettant de répondre à ces besoins. La seconde partie présente la solution proposée dans ce manuscrit.

4.2 Cahier des charges

Cette section justifie les caractéristiques nécessaires d'un module variationnel. Ensuite, un ensemble de caractéristiques différenciatrices sera proposé.

4.2.1 Caractéristiques nécessaires

Gestion des configurations sous, iso et sur-contraintes

Un modèle dynamique doit maintenir la cohérence des spécifications. Or, l'ensemble de celles-ci est enrichi au fur et à mesure que la procédure de conception progresse. L'état de contraintes du problème évolue en même temps. Celui-ci reste généralement sous-contraint. En effet, tant que la conception n'est pas terminée, le produit n'est pas parfaitement défini. Or, même si le problème n'est pas complètement spécifié, le modèle géométrique doit être cohérent avec l'ensemble des spécifications.

Pour une meilleure compréhension de la spécification d'un produit le concepteur peut préférer rendre hyperstatique le problème. La redondance de spécifications permet aussi de restreindre le nombre de solutions. Ces sur-contraintes volontaires doivent être distinguées des sur-contraintes inconsistantes. Celles-ci réduisent à zéro le nombre de solutions à un problème. D'où la difficulté pour un (ou plusieurs) concepteur d'appréhender l'influence des modifications

Analyse des spécifications

L'analyse des spécifications est un module nécessaire même si le problème à traiter n'est que de dimension 2 sans contrainte d'ingénierie. En effet, l'œil averti du concepteur peut avoir des difficultés à déceler les erreurs de spécification. En contrepartie, ce module devient particulièrement important en 3D ou pour les problèmes d'ingénierie tant la complexité de l'analyse s'est accrue. Même du point de vue des "contraintes dimensionnelles", la lecture des spécifications n'est pas des plus aisées. De plus, en 3D, les degrés de liberté des éléments couramment utilisés sont différents, tout comme les contraintes utilisables, sans compter la gestion nécessaire des problèmes topologiques. Pour toutes ces raisons, un module d'analyse est nécessaire afin d'offrir à l'utilisateur un maximum d'informations pertinentes.

Afin d'aider l'utilisateur dans la spécification dimensionnelle de son problème, l'état de contraintes local doit être indiqué. Les conséquences des spécifications

suivantes sont alors plus facilement appréhendées. Sur les mêmes bases, une sur-contrainte incohérente peut alors être expliquée.

Gestion de la topologie

En 3D, le concepteur spécifie le modèle géométrique de composants mécaniques. En d'autres termes, il spécifie des objets ayant une topologie particulière. Suite à la modification d'une contrainte, la mise à jour du modèle géométrique doit respecter les conditions de cohérence topologique de l'objet afin d'offrir un résultat cohérent.

Spécifications géométriques

Un modèle dynamique de spécifications remplace le modeleur d'une application de CAO. Par conséquent, il doit accepter l'ensemble des éléments géométriques classiquement utilisés en CAO. En 2D, cet ensemble est composé des points, droites, segments, coniques, arcs de coniques et courbes planes quelconques. En 3D, l'ensemble est complété par des courbes gauches quelconques, plans, surfaces quadriques et surfaces quelconques. Généralement, les courbes et surfaces quelconques sont modélisées par des courbes et surfaces NURBS¹.

Spécifications dimensionnelles

Au niveau des contraintes dimensionnelles, le modèle dynamique doit accepter l'ensemble des contraintes géométriques classiquement utilisé, mais ces contraintes ne sont définies qu'entre éléments simples c'est-à-dire entre points, droites, plans et cercles, ce qui limite les possibilités de l'utilisateur.

Spécifications d'ingénierie

La gestion des contraintes d'ingénierie correspond à un besoin des concepteurs. Il est donc primordial de prendre en compte cette spécification en début de définition d'un nouveau modèle dynamique au risque de ne pouvoir l'intégrer par la suite.

Robustesse de la résolution

Le modèle dynamique permet de maintenir cohérent un ensemble de spécifications. Certains modules permettent d'offrir à l'utilisateur différents diagnostics pendant la spécification ou en cas d'échec de résolution. À partir du moment où le système offre différents diagnostics, ceux-ci doivent être le plus fiable possible. En cas de doute, il faut que le modèle dynamique puisse le préciser.

¹NURBS : B-spline rationnelle non uniforme

Persistence, stabilité de la solution

Pour des raisons similaires à la nécessité d'offrir une approche robuste, la solution doit être persistante dans le temps.

La stabilité d'une solution est un problème plus délicat à résoudre dont la cause est la multiplicité des solutions d'un modèle. En effet, en raison de cette multiplicité, rien n'assure la convergence d'une résolution vers la solution attendue. Celle-ci est dite stable si, suite à une petite perturbation de l'état initial la solution n'est pas modifiée, ou si, suite à une petite perturbation de l'ensemble des contraintes la solution obtenue est "proche" de la solution initiale. Le critère subjectif "proche" est une des causes de la difficulté d'implémentation de cette caractéristique pour le second cas.

2D, 3D

À partir du moment où un concepteur dispose d'un outil de spécification d'un objet géométrique 3D, il voudra pouvoir spécifier indifféremment des problèmes 3D et des problèmes 2D. Le module de résolution doit pouvoir résoudre le problème quelque que soit la dimension du problème posé.

Mise à jour en "temps réel"

La réalisation d'un modèle géométrique est progressive. Le concepteur alterne entre la description de la géométrie et la spécification des contraintes. Le module variationnel est alors constamment sollicité pour mettre à jour la géométrie. La résolution doit alors être interactive afin de ne pas bloquer le concepteur dans son travail.

4.2.2 Caractéristiques différentielles

Les caractéristiques différentielles ne sont pas nécessaires à la définition d'un bon module variationnel. Elles ne représentent que des atouts supplémentaires permettant de rendre le travail de l'utilisateur plus agréable. Un produit n'est jamais parfait, il est toujours possible de l'améliorer. Cette liste est donc infinie. Les quelques caractéristiques présentées ici, sont des points qui ne sont pas réellement nécessaires mais dont l'intérêt est suffisamment important pour les proposer.

Multiplicité des solutions

Un problème variationnel admet généralement plusieurs solutions. Le solveur en propose une à l'utilisateur. Elle est choisie sur des critères propres au solveur,

mais il n'y a aucune raison pour que ce module propose toujours la solution attendue. L'utilisateur peut aussi souhaiter parcourir les différentes solutions. De plus, certaines solutions peuvent être refusées par le modeleur à cause d'incohérences topologiques. Pour toutes ces raisons, un solveur doit pouvoir proposer, si ce n'est l'ensemble des solutions, au moins plusieurs solutions.

Spécifications des courbes et des surfaces

Une courbe, comme une surface quelconque, est un objet difficile à définir. En effet, le nombre de paramètres décrivant ces objets peut être très élevé. En pratique, des approximations de degré fini et un principe de découpage sont utilisés afin de définir des formes particulières. Les utilisateurs peuvent alors souhaiter contraindre ces formes pour qu'elles répondent au cahier des charges, mais le résultat obtenu doit aussi vérifier des critères subjectifs pour que l'utilisateur soit satisfait.

Paquets rigides

Un sous-problème peut être sous-contraint mais correspondre parfaitement aux attentes de l'utilisateur. Afin de rendre les modifications locales impossibles, il peut toujours ajouter un ensemble suffisant de contraintes pour le rendre bien-contraint. Cette approche n'empêche pas une éventuelle modification de cette zone en une autre solution du problème. L'utilisateur souhaite donc un moyen plus simple et plus sûr d'imposer la "rigidité" d'un sous-problème.

Comportements par défaut dits naturels

L'utilisateur lorsqu'il ajoute ou modifie une contrainte, s'attend à une certaine modification de l'objet. Il se représente le résultat mentalement. Celui retourné par le module doit donc être conforme à son attente, pour ne pas le surprendre et surtout, ne pas le satisfaire. Un solveur qui propose des solutions naturelles a donc plus de chances de plaire aux utilisateurs qu'un solveur très puissant qui les déstabilise à chaque résolution.

4.3 Description globale de l'approche

Considérant la résolution d'un problème géométrique contenant des contraintes d'ingénierie, l'analyse bibliographique a montré l'incapacité des approches géométriques à les traiter (Cf. section 3.3). Une approche algébrique a donc été préférée pour tendre vers une plus grande généralité des domaines abordés.

L'état de l'art sur les solveurs géométriques basés sur une approche algébrique (Cf. section 3.2) a d'un côté, montré que ces solutions n'étaient pas très satis-

faisantes, mais a surtout fait ressortir le point qui semble être la cause de leurs mauvaises performances. En effet, aucune réflexion n'a été réalisée sur la mise en équations alors qu'un point important pour la résolution d'un problème, quelque soit le domaine, est la qualité de sa formulation qui n'est pas sans effet sur la capacité à élaborer des diagnostics sur l'état de contrainte, la cause d'échec de résolution, ...

La présentation de l'architecture générale du module va permettre d'introduire les différentes parties traitées dans ce manuscrit en tirant parti des remarques précédentes.

4.3.1 Architecture générale

La solution proposée peut être vue comme un "modèle dynamique de spécifications d'ingénierie". Afin de justifier cette description, cette section commence par des rappels sur la modélisation paramétrique et variationnelle. Ensuite, ce modèle est présenté comme l'évolution naturelle d'une application de CAO qui souhaite répondre à cette problématique.

Les approches paramétriques

Jusqu'à présent un logiciel de CAO est construit autour d'un modelleur géométrique. Ce module a toujours occupé une place centrale. Initialement, il ne permettait que de construire la modélisation géométrique d'un composant ou d'un mécanisme. Il est devenu ensuite paramétrique. Son rôle a donc évolué jusqu'à la gestion actuelle de la cohérence de la description du produit et de son modèle géométrique. Cependant, les spécifications dans un modelleur paramétrique sont limitées aux procédures et aux paramètres de construction qui sont ordonnés de façon irrévocable dans l'historique de construction (Cf. section 2.2).

Les approches paramétriques ne sont donc que des solutions qui ne répondent que partiellement à la problématique de ce manuscrit.

Les approches variationnelles

Les approches variationnelles n'apparaissent dans une application de CAO qu'au niveau du sketcher 2D. Ce module n'est qu'un satellite du modelleur. Ce comportement variationnel est obtenu d'un module généralement extérieur appelé solveur. L'utilisateur peut alors spécifier dans ces modules, un ensemble d'éléments géométriques contraints par des spécifications d'ingénierie (généralement limitées aux spécifications dimensionnelles).

Le solveur doit maintenir la cohérence d'une partie des spécifications. En effet, seul le positionnement des différents éléments géométriques est assuré. La

cohérence de la topologie est alors vérifiée par l'utilisateur qui accepte ou non la solution proposée.

Une approche variationnelle 3D.

Les solveurs variationnels 2D existent et sont utilisés dans les logiciels de CAO depuis maintenant une dizaine d'années. La problématique actuelle est de définir un module variationnel 3D. Le rôle de ce module n'est plus uniquement de résoudre un problème géométrique. En 3D, il est beaucoup plus complexe. En effet, en 2D, il suffisait de repositionner un ensemble d'objets géométriques sans autre considération. L'objet 3D, lui, a initialement des propriétés qui doivent être conservées tout au long de la conception. Cette tâche est trop complexe pour être déléguée à l'utilisateur. Elle revient donc à ce module.

La propriété la plus importante d'un objet 3D est sans doute la conservation de sa cohérence volumique, tâche qui incombe classiquement au modelleur géométrique. Contrairement à un modelleur paramétrique, ici, elle consiste à reconstruire un objet avec des propriétés topologiques semblables à l'original. Un modelleur paramétrique assemble des objets tout en conservant leurs propriétés topologiques. Ce module n'est donc pas uniquement un solveur mais aussi un modelleur.

Un modèle dynamique

Le rôle premier d'un module variationnel est de maintenir la cohérence des spécifications. Ces données doivent donc être gérées par ce module. Un modèle d'information est donc nécessaire. Celui-ci est différent de l'historique de construction, très rigide, d'une approche paramétrique. Il rassemble des spécifications de nature très différente dont l'ordre de création n'a plus d'importance. La gestion de la cohérence déléguée à un solveur et un modelleur lui offre ce caractère dynamique.

Conclusion

Dans une approche paramétrique, le modelleur géométrique est le noyau central de l'application. Dans une approche variationnelle, le cœur d'un logiciel de CAO devient un modèle dynamique de spécifications d'ingénierie. Celui-ci est composé d'un modèle d'information qui regroupe les données persistantes et de deux modules de maintien de la cohérence : un solveur d'ingénierie pour l'aspect "dimensionnel" et un modelleur géométrique variationnel pour la partie topologie.

4.3.2 Concepts du modèle d'information

Le modèle d'information contient la description des spécifications d'un produit. Il peut donc être vu comme une structure relationnelle de données, à l'instar des systèmes de gestion de données techniques (SGDT). Mais cette vue n'est pas suffisante. En effet, les informations à ce niveau n'ont pas un format et une structure qui permettent à ce module de réaliser les différentes tâches qui lui incombent. Elles ne représentent donc que la partie publique du modèle d'information, c'est-à-dire la partie enrichie et modifiée par l'utilisateur.

Deux autres niveaux sont alors nécessaires pour maintenir la cohérence des informations. Le niveau intermédiaire est un modèle d'information géométrique. Il a deux rôles. Le premier est de proposer une formulation intermédiaire du problème permettant une "bonne" mise en équations. Le second permet au modéleur de maintenir une cohérence topologique.

Le niveau le plus bas est une reformulation algébrique du problème global. Il est donc exclusivement destiné au solveur d'ingénierie. Les informations à ce niveau sont composées de données issues du modèle géométrique et de données non géométriques provenant du modèle public.

Le modèle d'information géométrique

Une géométrie, qu'elle soit volumique, surfacique ou linéique, est composée d'un ensemble d'objets élémentaires. Tous ces objets peuvent être définis par un ensemble de paramètres dans un espace de travail. De plus, avec une approche de type B-rep, ces objets doivent être reliés entre eux par des relations topologiques pour définir complètement une géométrie.

Classiquement, un élément géométrique est défini par sa position et sa forme dans un espace cartésien. La position de chaque élément est donc absolue dans un repère particulier. Ce repère n'a aucune utilité pour une définition intrinsèque d'un objet, alors que l'information importante pour un objet géométrique est la position relative de ces différents éléments. Par ce repère, des choix prématurés sont donc faits qui n'apportent rien à la définition d'une pièce.

L'idée de base de ce modèle est de décrire la position des différents éléments de façon naturelle : un élément n'est connu que par ses dimensions et/ou sa position par rapport à d'autres éléments. Cette description est donc très proche des descriptions habituelles d'un objet par un concepteur. Par exemple, un plan de définition ne décrit que la forme et les dimensions. Dans un plan, aucun repère n'est défini. La description est alors intrinsèque à la pièce.

Pour cela, la géométrie est séparée en deux composants (Figure 4.2). Le premier établit la position des éléments géométriques les uns par rapport aux autres. Ce positionnement est donc relatif. Aucune notion de repère n'est présente. Le formalisme utilisé (Cf. chapitre 5) fait penser à une structure mécanique. Pour cette

raison, cette partie de la description sera la *structure* d'un objet géométrique. Le second est un corps déformable. Il n'a alors aucune "forme", aucune dimension tant qu'il n'est pas positionné sur la structure. Par analogie avec une structure à toile tendue, il représente la *toile* de l'objet.

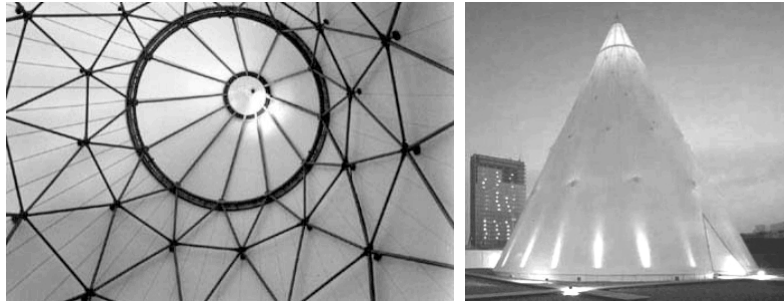


FIG. 4.1: Cet exemple de tipi illustre le concept de toile tendue sur une structure rigide. A gauche, une vue sur la structure du haut du tipi.

Cette approche étant basée sur une modélisation volumique de type B-rep, la toile est composée d'éléments surfaciques et linéiques reliés entre eux par des relations topologiques. Ils sont portés par des supports géométriques. Ces supports sont positionnés les uns par rapport aux autres grâce à la structure. Le couple toile-structure définit alors complètement un objet géométrique.

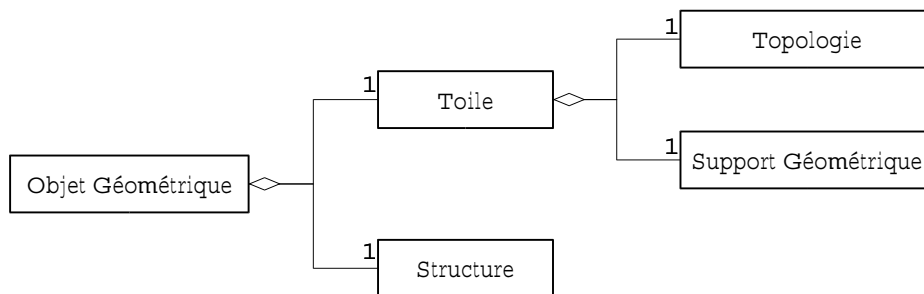


FIG. 4.2: Représentation UML d'un objet géométrique

Ce découpage permet de séparer la partie "géométrie et topologie" de la partie dimensionnelle, tout comme une approche B-rep sépare la partie topologie de la partie géométrie. Les raisons de ces séparations sont identiques. Dans un modéleur, les traitements géométriques et topologiques sont relativement indépendants. De même, la position des différents éléments n'est pas directement liée aux deux premières. Cette séparation a ainsi l'avantage de bien distinguer les différents domaines traités.

Une structure relationnelle horizontale et verticale

L'analyse de la structure montre une forte imbrication des données. En effet, des liens horizontaux sont présents à chaque niveau. Par exemple, le ni-

veau supérieur contient les spécifications d'un produit, c'est-à-dire des entités en partie géométriques définies par un ensemble de contraintes. Chaque entité est donc contrainte pas un ensemble de spécifications. Chaque contrainte relie un ensemble d'entités. Ce niveau est nécessaire pour modifier et/ou corriger les spécifications. Le niveau inférieur, qui contient l'ensemble des équations, est aussi fortement relationnel. Une équation définit les liens entre des paramètres. Ces liens sont différents des relations initiales, mais ils sont nécessaires pour résoudre le problème. La présentation du modèle géométrique fera aussi ressortir une imbrication horizontale à ce niveau.

La persistance de ces trois niveaux est nécessaire pour plusieurs points. Le premier qui vient à l'esprit est une optimisation du temps de calcul. En effet, l'ajout d'une nouvelle spécification n'entraîne alors qu'une modification locale de cet ensemble de données. Pour cela, des liens descendants doivent exister pour que les modifications puissent se répercuter sur les équations. Le second point concerne les outils d'analyse. Ces examens sont réalisés au niveau des équations. Une fois l'analyse effectuée, le diagnostic doit être retourné à l'utilisateur, mais celui-ci ne comprend que le langage des spécifications. Des liens verticaux sont donc nécessaires pour lier les paramètres des équations aux spécifications initiales et inversement.

4.3.3 Rôle du solveur

Un problème géométrique est composé d'éléments dont la valeur des paramètres est soit spécifiée, soit inconnue. La tâche première du solveur est de proposer des valeurs pour les paramètres non-spécifiés telles que l'ensemble soit cohérent. La valeur de ces paramètres permet, entre autres, de positionner les différents éléments les uns par rapport aux autres.

La figure 4.3 illustre cette tâche. Comme la modélisation géométrique est basée sur une approche B-rep, le solveur doit positionner les supports géométriques des composants du modèle. La figure 4.3-b représente la position initiale des différents supports géométriques du modèle (Figure 4.3-a). La figure 4.3-c est une solution possible. La tâche du solveur s'arrête ici. En effet, il n'a aucune représentation de modèle topologique.

L'utilisateur attend plus d'un solveur. Comme cela a été exposé dans le cahier des charges, le module doit pouvoir analyser l'état de contraintes du problème. Cette tâche est importante pendant l'étape de spécification. En effet, l'état de contraintes permet à l'utilisateur de mieux anticiper les conséquences de ses spécifications. Elle est aussi primordiale en cas d'échec de résolution. Elle va permettre d'en expliquer les causes. Ces tâches sont complémentaires de l'étape de résolution, c'est pourquoi elles sont confiées au solveur.

4.3.4 Rôle du modelleur

Le concepteur définit des modèles géométriques ayant une topologie particulière (Figure 4.3-a). Généralement le modèle représente un volume. Il est contraint par un ensemble de spécifications. Le solveur le met à jour pour qu'il soit cohérent avec sa description (Figure 4.3-c). Le placement des supports géométriques ne suffit pas à définir un modèle cohérent d'un point de vue topologique.

Le modelleur, connaissant le modèle initial (Figure 4.3-a) et la nouvelle disposition des éléments géométriques (Figure 4.3-c), doit proposer un modèle topologiquement cohérent. Les figures 4.3-(d et e) sont deux modèles parmi d'autres. Or, sur ces deux résultats, seul le premier (Figure 4.3-d) est correct d'un point de vue topologique.

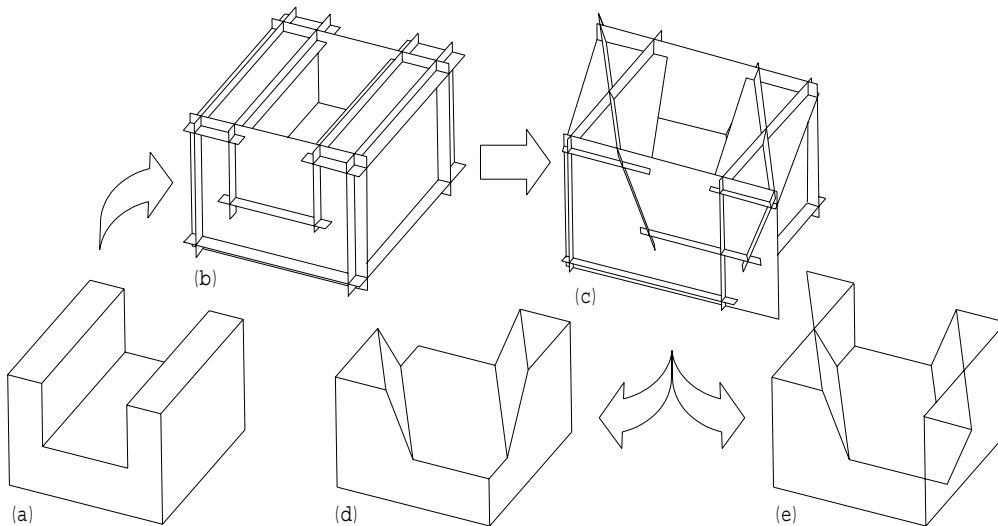


FIG. 4.3: Les différentes étapes d'une mise à jour d'un modèle variationnel.

4.3.5 Conclusion

Une solution globale à la problématique de cette thèse a été proposée. Nous avons vu que le premier niveau du modèle dynamique est la partie publique des données. La prochaine section va présenter les différentes spécifications acceptées par le modèle.

4.4 Les spécifications

4.4.1 Introduction

Le principe de la géométrie variationnelle est de définir un objet par un ensemble de spécifications, contrairement à la géométrie paramétrée qui le définit par un

ensemble de procédures de construction. Pour ce faire, l'utilisateur dispose de trois familles de spécifications :

- les éléments géométriques ;
- les contraintes géométriques ;
- les contraintes d'ingénierie.

L'objet doit alors être décrit à partir de ces concepts. Contrairement à la géométrie paramétrée, la grammaire de ce langage est très simple. La seule règle s'adresse à quelques spécifications qui doivent s'appuyer sur des objets géométriques existants. Il est nécessaire que ces éléments existent pour qu'elles puissent être formulées. Les différents mots sont donc écrits dans une simple base de données relationnelle. Cette section ne décrit donc que les différentes familles.

4.4.2 Les éléments géométriques

Un objet est décrit dans un module paramétrique par un plan de construction (Cf. sous-section 1.3.2). Les spécifications sont alors essentiellement des procédures : extrusion, révolution, protrusion.

Dans une application variationnelle, l'utilisateur ne décrit plus les étapes de construction de son objet, mais l'objet lui-même. Les mots utilisés sont alors très simples : face plane, face gauche, segment, arc de courbe plane, arc de courbe gauche, point. Il peut aussi utiliser des mots plus précis : congé de raccordement à rayon constant ou variable, chanfrein, surface quadratique, surface cylindrique ou conique ou sphérique, arc de cercle ou de conique.

Ce nouveau vocabulaire ne veut pas dire que les outils de construction de la géométrie vont changer. Rien n'empêche à un objet d'être construit avec une fonction similaire à celles énumérées ci-dessus. Cette méthode de construction a peut-être l'avantage de proposer un paramétrage par défaut, mais l'intérêt est que ce paramétrage puisse être modifié sans avoir à reconstruire la géométrie. Les concepts constituant cette approche variationnelle sont présentés ci-après.

4.4.3 Les contraintes géométriques

Les éléments géométriques permettent de décrire la forme d'un objet. Or, les dimensions de cet objet ne sont pas fixées. La position, les dimensions des différents éléments le constituant sont bien sûr modifiables *manuellement*, mais la cohérence de l'objet global n'est en rien assurée. C'est alors le rôle de l'utilisateur de la vérifier entièrement mais son jugement n'est pas toujours sans erreur.

L'utilisation de contraintes géométriques va permettre à l'utilisateur d'imposer certaines propriétés et dimensions. Pour cela, il dispose d'un certain vocabulaire. Plusieurs langages ont été proposés (Cf. sous-section 1.1.3). Celui utilisé ici est inspiré des treize contraintes du modèle SATT [Clément *et al.* 1994]. Il est,

entre autre, enrichi de contraintes s'appliquant à des courbes. Ces contraintes peuvent être divisées en deux groupes : les contraintes dimensionnelles et les contraintes de coïncidence. Un troisième groupe de contraintes existe : la spécification des caractéristiques propres à certains éléments complexes (cercles, paraboles, etc.).

Les contraintes dimensionnelles

La contrainte de distance. Il est toujours possible de définir la distance entre deux éléments. Cependant, en fonction du couple d'éléments, cette définition doit être clarifiée. En effet, la distance entre deux points existe toujours alors que la distance entre droites et/ou plans sous-entend que ces éléments ne sont pas incidents. Si le second élément est une courbe, ce critère doit être précisé. En effet, dire que la distance d'un point à une courbe est d signifie que la distance minimale du point aux différents points de la courbe est d . De plus, le point de la courbe satisfaisant la condition n'est pas nécessairement unique. Or, en CAO, seule une portion de courbe est connue. L'expression de la distance doit donc être aménagée en conséquence. L'utilisateur peut aussi préférer localiser sa contrainte en une portion ou un point particulier de la courbe.

Une distance ne peut pas être nulle au risque de faire dégénérer le problème. En effet, prenons un triangle qui est normalement parfaitement défini par la longueur de ses trois côtés. Si une des longueurs est nulle, le problème change de dimension pour devenir un problème de dimension 1. Il devient alors sur-contraint puisqu'une seule longueur suffit à construire la géométrie. Pour éviter ces dégénérescences, l'utilisateur doit préférer des contraintes de coïncidence aux contraintes de distances nulles.

La contrainte angulaire Une contrainte angulaire s'applique entre deux éléments de type linéique ou surfacique. Dans les cas où la valeur de cette contrainte est respectivement égale à $0 [\pi]$ ou $\pi/2 [\pi]$, nous parlerons respectivement de contrainte de parallélisme et de contrainte de perpendicularité.

Contrairement au modèle SATT, les contraintes angulaires ne sont pas associées à une contrainte de distance. L'utilisateur peut alors librement la compléter ou non d'une telle contrainte. De plus, les contraintes de perpendicularité et de parallélisme ne sont pas distinguées des contraintes angulaires.

La contrainte de la distance d'un point avec une courbe est une contrainte globale, c'est-à-dire qu'elle s'applique à tous les points de la courbe, par contre la contrainte angulaire est délicate à spécifier puisqu'elle s'applique à un point particulier de la courbe. Le problème est alors de définir le point d'application de la contrainte. En effet, il peut être défini comme le point de paramètre u^* , ou le point d'abscisse curviligne s^* ou par de nombreux autres critères.

Par contre, cette contrainte se définit très bien pour les extrémités de l'arc de courbe. En effet, ces points particuliers sont parfaitement connus quelque soit les modifications apportées à la courbe.

Les contraintes de coïncidence Les contraintes de coïncidence imposent que tous les points de l'élément de variété géométrique de plus petite dimension se superposent aux points du second.

Les contraintes internes

Cette famille de contraintes s'applique aux paramètres intrinsèques de certains éléments. La plus classique est le rayon d'un cercle, alors que la longueur d'un segment est généralement transformée en distance entre les deux points extrémités. Il y a aussi, la distance focale d'une parabole, ainsi que les autres paramètres des coniques,... Ces contraintes, contrairement aux autres contraintes géométriques ne s'appliquent que sur un élément.

4.4.4 Les contraintes d'ingénierie

Les contraintes d'ingénierie s'appliquent sur un ou plusieurs éléments. Elles peuvent être une simple généralisation des contraintes géométriques. Par exemple, une contrainte pourrait spécifier qu'une distance appartient à un intervalle de valeurs discrètes ou continues, au lieu de préciser une valeur stricte. Les contraintes d'ingénierie peuvent être beaucoup plus générales. Par exemple, elles peuvent lier différents paramètres de natures diverses. Le critère principal pour qu'une contrainte puisse être définie est que la relation entre les différents paramètres soit formulée sous la forme d'un ensemble d'équations et d'inéquations ou qu'il existe une *boîte noire* dont les paramètres d'entrée et sortie soient les mêmes que ceux de la contrainte.

4.4.5 Conclusion

La partie publique est la partie compréhensible par l'utilisateur. C'est au travers de ces spécifications qu'il enrichit et modifie son modèle. Par contre, le format de ces données ne permet de réaliser aucune analyse. C'est pour cette raison que ces données sont reformulées soit sous une forme *géométrique* pour les éléments et les contraintes géométriques, soit directement sous une forme *algébrique* pour les autres. Le chapitre suivant présente le modèle géométrique.

4.5 Synthèse

Une solution globale à la problématique de cette thèse a été proposée. Cette solution repose sur une complète redéfinition des principaux modules d'une application de CAO. En effet, le modelleur laisse la place à un modèle dynamique de spécifications. Il passe alors, au même titre que le solveur, au service de ce modèle pour maintenir la cohérence des données.

La construction de ce modèle dans son ensemble sort de l'étendue du travail décrit. Les travaux se sont donc concentrés autour de la définition d'un solveur d'ingénierie répondant au maximum de points du cahier des charges. Afin de partir sur des bases saines, les réflexions ont tout de même débuté sur l'architecture globale du modèle. Certains points n'ont, quant à eux, pas encore été traités. En particulier, le modelleur, dont le rôle est tout aussi important que celui du solveur et la problématique aussi riche, n'a reçu aucune attention.

Les parties suivantes présentent en détails les différentes propositions.

Chapitre 5

Modèle géométrique

Le modèle géométrique est le second niveau d'information du modèle de spécification. Son rôle est de formuler le problème géométrique afin qu'il puisse retranscrire le plus fidèlement possible les spécifications sous forme de relations algébriques.

5.1 Introduction

La section 4.3.2 a introduit le principe du modèle d'information géométrique. Nous avons vu qu'il est fondé sur une description originale de la géométrie. En effet, l'idée de base de ce modèle géométrique est de décrire un problème d'une façon intrinsèque. L'intérêt de cette modélisation est qu'elle favorise une mise en équations locale. Le problème global n'est alors plus décrit dans un repère absolu, mais de façon naturelle afin d'exprimer la position relative des éléments géométriques. Ce chapitre présente ce modèle géométrique, mais son originalité ne sera dévoilée que dans les différents chapitres suivants, en particulier au chapitre présentant le passage des données du modèle géométrique au modèle algébrique.

5.2 La structure

Le modèle géométrique présenté est basé sur une approche B-rep. Un objet peut être donc défini par un ensemble de points, d'éléments surfaciques et/ou linéiques. Tous ces composants peuvent être définis par la position d'un ensemble de points caractéristiques dans l'espace de travail et pour certains, par un ensemble de données supplémentaires (vecteur nodal et coordonnées homogènes pour les NURBS). Il existe différentes approches pour définir la position de ces points dans l'espace. La plus classique définit un repère et positionne ces points dans le repère. La description obtenue est alors globale. Cette approche est très

utilisée dans toutes les applications graphiques. Or, un objet géométrique est composé d'éléments positionnés les uns par rapport aux autres. Le principal inconvénient pour la problématique de cette thèse est qu'avec ce type de description, les relations entre éléments ne sont pas définies directement.

L'idée est alors de privilégier un positionnement relatif des points. Or, deux points sont parfaitement positionnés l'un par rapport à l'autre dès que la distance entre ces points est connue. Pour trois points, la connaissance de trois longueurs, de deux longueurs et d'un angle ou d'une longueur et de deux angles est suffisante. De nombreux exemples de positionnement relatif existent dans la vie de tous les jours : les feuilles d'un arbre sont positionnées les unes par rapport aux autres grâce aux branches, qui le sont de la même manière sur le tronc. L'ossature du tipi (Cf. figure 4.1) est un exemple de positionnement des points caractéristiques de la toile. Un dernier exemple est le robot de manutention de la figure 5.1. Ce robot est constitué de poutres de longueurs variables et de liaisons qui définissent l'angle entre deux poutres adjacentes. En fonction des liaisons et de la longueur des poutres, la nacelle peut être positionnée en chaque point de l'espace.

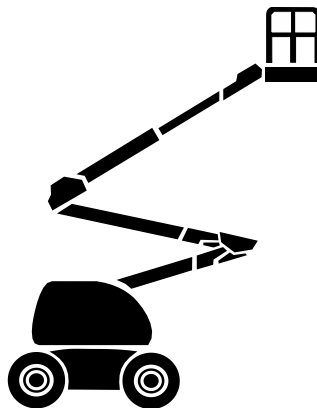


FIG. 5.1: Un robot de manutention illustre parfaitement le principe du positionnement relatif

5.2.1 Définitions

La structure de données est composée de six éléments principaux définissant la structure d'un objet :

la variable représente la mesure des éléments mesurables de la structure. Elle connaît sa valeur initiale, sa valeur courante. Elle peut avoir une valeur imposée ;

le point est un point appartenant à la structure sur laquelle peut être accroché un (des) point(s) caractéristique(s) de l'objet à définir. Seul, le **point** n'a aucune position relative. Dès que plusieurs **points** sont créés, ils ne sont

connus que par leurs positions relatives. Afin d'afficher la géométrie, il est nécessaire de connaître la position de chaque **spoint** dans un repère. C'est donc seulement pour l'affichage du résultat final qu'un **spoint** est défini par ses coordonnées dans un repère. Lorsqu'il n'y a pas d'ambiguïté, afin ne pas alourdir l'exposé, un point ne sera pas distingué de son **spoint** support ;

le **vecteur** est un élément de l'espace vectoriel qui possède à la fois un sens et une direction. Le sens et la direction d'un **vecteur** dans un plan vectoriel orienté ne sont connus, dans ce formalisme, que par rapport à d'autres **vecteurs**. Un **vecteur** est unitaire. Il est généralement noté \mathbf{e}_i ;

l'**angle** (orienté) est une quantité qui définit le positionnement relatif de deux vecteurs dans un plan vectoriel orienté. Un angle est donc composé de deux vecteurs ordonnés (le premier pour l'origine et le second pour l'extrémité) et d'un scalaire signé : une **variable**. Il définit donc de façon univoque la position d'un **vecteur** par rapport à un autre dans le plan vectoriel commun. L'angle entre les vecteurs \mathbf{e}_i et \mathbf{e}_j est noté $(\widehat{\mathbf{e}_i, \mathbf{e}_j})$;

le **pointlink** est un vecteur de l'espace affine. Il est composé de deux **spoints** ordonnés : le premier pour l'origine et le second pour l'extrémité ;

le **bipoint** appartient à la famille des pointlinks. Il est porté par un **vecteur**. Sa mesure est définie par une **variable**, dont le signe représente son sens par rapport à son **vecteur** support. Un **bipoint** ne doit pas être de mesure nulle au risque de faire dégénérer le problème. En effet, la direction du **vecteur** support d'un bipoint de mesure nulle est difficilement identifiable. Le **bipoint** liant les points **A** et **B** est noté \mathbf{AB} ;

le **copoint** fait partie de la famille des pointlinks. C'est un **pointlink** de mesure nulle. Il n'est donc pas porté par un **vecteur**. Il permet de spécifier la coïncidence entre deux points au lieu d'utiliser un bipoint de longueur nulle. Le **copoint** liant les points **A** et **B** est noté $\widehat{\mathbf{AB}}$;

le **graphe affine** regroupe l'ensemble des **spoints** et des **pointlinks**. C'est un graphe orienté dont les arcs sont les **pointlinks**. Il définit un espace affine ;

le **graphe vectoriel** regroupe l'ensemble des **vecteurs** et des **angles**. C'est un graphe orienté dont les arcs sont les **angles**. Il définit un espace vectoriel.

5.2.2 Structure 2D

Le modèle géométrique positionne l'ensemble des points caractéristiques d'un objet géométrique avec la même approche que le robot de manutention de la figure 5.1. Chaque point est accroché à l'extrémité d'un **bipoint** ou d'un **copoint**. Les **bipoints** représentent les poutres d'une structure. Les **copoints** permettent de définir la coïncidence entre deux points. Les **copoints** sont utilisés pour ne

pas faire dégénérer la structure avec un **bipoint** de longueur nulle. Les points jouent aussi une partie du rôle des liaisons : à chaque point sont accrochés des **bipoints**. Le deuxième rôle de la liaison, qui consiste à définir l'angle entre deux poutres, est obtenu par deux autres entités : le **vecteur** et l'**angle**. Par définition, un **bipoint** est porté par un **vecteur**. Pour définir l'angle entre deux **bipoints**, il suffit alors de définir l'angle entre deux **vecteurs**. À partir de ces quatre éléments, une structure de données (Figure 5.2) a pu être définie afin de stocker les informations nécessaires pour représenter la structure d'un objet 2D.

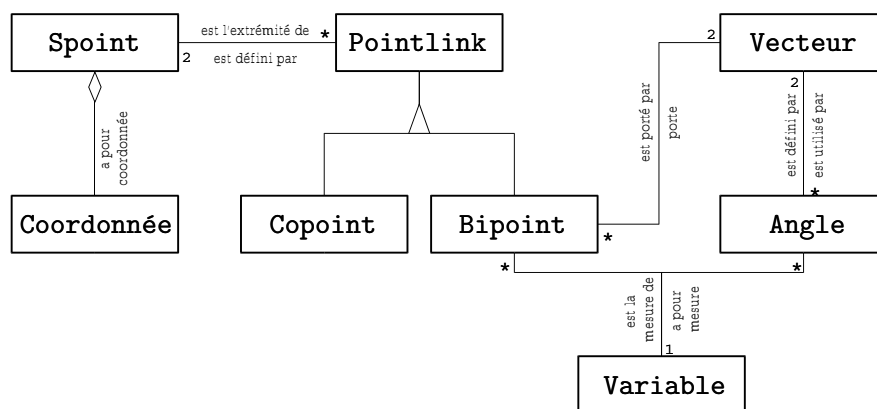


FIG. 5.2: diagramme de classes de la structure 2D

5.2.3 Explications

Dans une approche déclarative, un point seul est connu dès qu'il a été créé. Il n'y a aucune raison d'essayer de le situer puisqu'il est seul (Figure 5.3-a). Pour deux points, l'information nécessaire est la distance entre ces deux points. Cette distance est définie dans ce formalisme par le module du **bipoint** rattaché à ces deux points (Figure 5.3-b). La position d'un troisième point en fonction des deux premiers est connue par les caractéristiques des **bipoints**, **vecteurs** et **angles** nécessaires à décrire les trois points (Figure 5.3-c). Pour trois points alignés, les **vecteurs** supports des **bipoints** sont colinéaires. Suivant les cas, il sera préféré d'avoir deux **bipoints** portés par un seul **vecteur** (Figure 5.3-d) ou bien des **bipoints** portés par des **vecteurs** colinéaires (Figure 5.3-e).

Le point fort de cette approche est de pouvoir définir une géométrie en fonction de l'intention de l'utilisateur et des spécifications géométriques. En effet, la structure d'une géométrie n'est pas unique, elle est construite en même temps que la géométrie. Elle est déduite des actions de l'utilisateur. L'intention de l'utilisateur est captée en fonction des opérations qu'il réalise. Prenons l'exemple d'une géométrie polygonale simple (Figure 5.4-a). La figure 5.4-b représente un squelette possible de cette géométrie. Cette structure provient d'une définition du contour

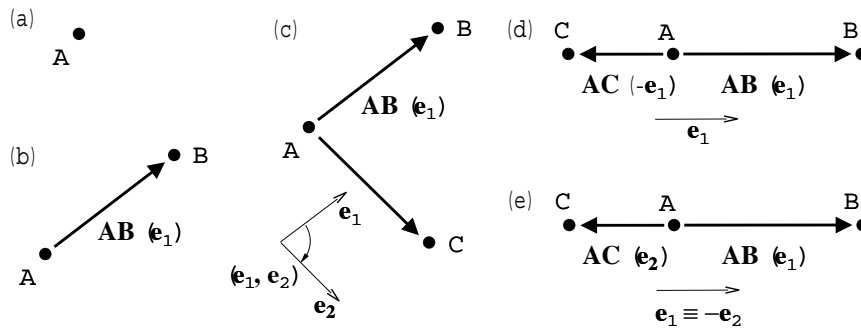


FIG. 5.3: Quelques exemples simples de positionnement de points

à l'aide d'une polyligne. Une polyligne est une succession de courbes reliées avec une continuité C^0 . Chaque point est positionné en fonction des points précédents. La figure 5.4-c présente un autre squelette possible. Dans cet exemple, l'intention de l'utilisateur serait de positionner quatre points par rapport à un cinquième.

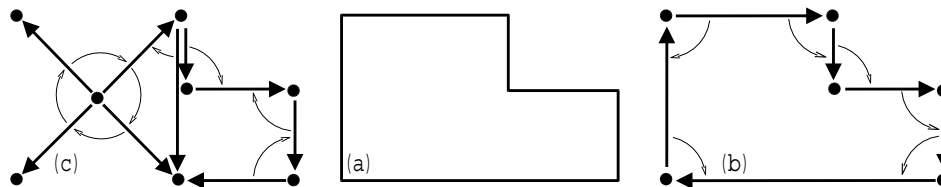


FIG. 5.4: Différentes structures pour un même objet

5.2.4 Généralisation au 3D

Un des intérêts du modèle géométrique 2D est de distinguer l'espace vectoriel de l'espace affine. En effet, l'ensemble des **vecteurs** et des **angles** définit un espace vectoriel. Le but de cette séparation est de faciliter la mise en équations du problème. Ceci sera exposé à la section 8.3 où sera expliqué aussi la contribution obtenue en identifiant les sous-espaces communs aux différents éléments. Il faut noter que ceci est déjà le cas dans la structure 2D. En effet, le **vecteur**, le **spoint** et le **bipoint** sont respectivement des éléments vectoriels de dimension un, ou affine de dimension zéro et un.

Les objets manipulés en CAO sont généralement des objets construits à partir de projections 2D. Ce qui signifie qu'ils sont composés de nombreux plans de référence matérialisés éventuellement par des faces planes. En d'autres termes, un problème 3D sera généralement composé de sous-problèmes 2D. Toujours dans l'esprit de décomposer le problème en sous-problèmes, il paraît donc nécessaire de reconnaître les différents sous-espaces.

Dans cet esprit, après avoir défini les entités de dimension zéro et un, la structure intègre les éléments de dimension 2 : le **plan vectoriel** et le **plan**

affine. Avant de les définir, il convient de présenter deux nouveaux éléments vectoriels :

la normale est un vecteur normé. Elle possède donc à la fois un sens et une direction. Comme pour le **vecteur**, ses caractéristiques ne sont connues, que par rapport aux autres **normales**. Une **normale** est généralement notée \mathbf{n}_i ;

l'angle sphérique est, par abus de langage, l'angle entre deux **normales**. Nous verrons qu'en réalité un angle sphérique est l'angle formé par deux plans (Cf. sous-section 8.5.1). Contrairement à un **angle**, un **angle sphérique** n'est pas orienté et n'appartient pas à un **plan vectoriel**.

À partir de ces deux nouveaux éléments, les **plans vectoriels** et **affines** peuvent être définis :

le plan vectoriel regroupe l'ensemble des vecteurs et des angles coplanaires. Il est porté par une **normale** définissant son vecteur normal. Le sens de ce vecteur définit l'orientation du plan et donc celui de ces angles. Il est aussi appelé **vplan**.

le plan affine ou **aplan** est porté par un **plan vectoriel**. Il contient l'ensemble des **spoints** et des **bipoints** qui lui sont coplanaires.

le graphe sphérique regroupe l'ensemble des **normales** et des **angles sphériques**. C'est un graphe orienté dont les arcs sont les **angles**. Il définit aussi un espace vectoriel.

La figure 5.5 représente la nouvelle structure de données. Il faut préciser qu'il n'est pas possible de définir un angle (**angle** ou **angle sphérique**) entre un **vecteur** et une **normale**. Ce choix de structure est dû aux relations que l'on sait écrire entre ces angles.

5.3 La toile

Le modèle géométrique utilise une modélisation B-rep pour décrire la toile d'un objet. La théorie utilisée est celle des complexes cellulaires (Cf. annexe D). L'intérêt de ce concept est qu'il est le plus général et le plus fondamental. Son avantage principal est de gérer les modèles dégénérés.

5.3.1 Structure de données

Les données représentant un complexe cellulaire sont structurées en deux parties (Figure 5.6). La première partie, traditionnellement appelée *topologie* représente et gère le graphe d'adhérence. La seconde partie, appelée *support géométrique* représente tous les supports dont les cellules sont des restrictions. Ces deux parties

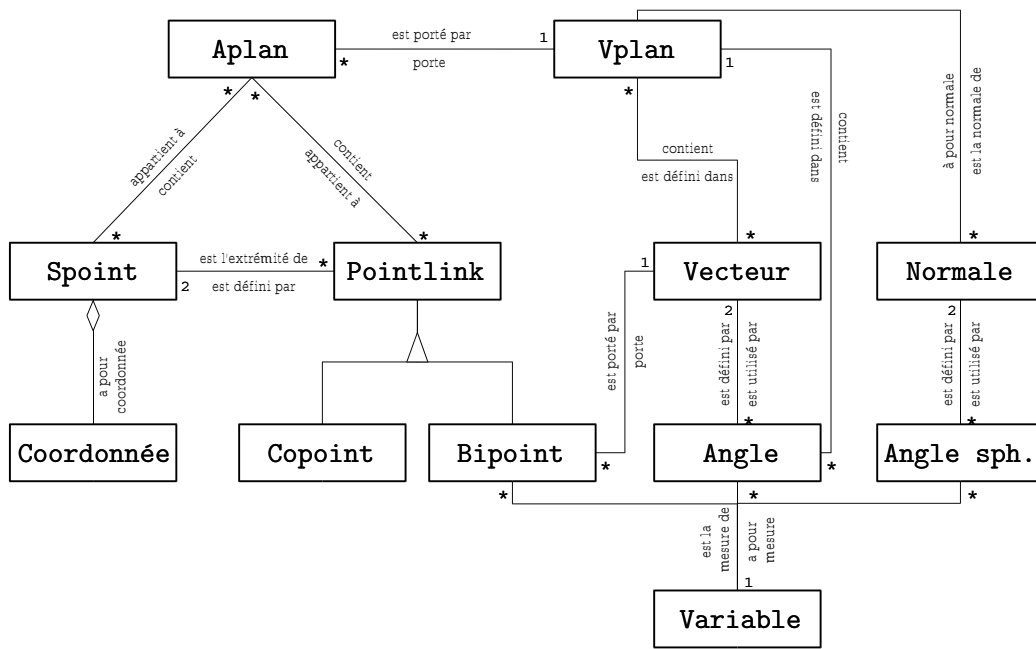


FIG. 5.5: diagramme de classes de la structure 3D

communiquent par le lien “... est une restriction de ...” d’une cellule vers son support. Les supports sont, quant à eux, définis en partie par un ensemble de spoints de la structure.

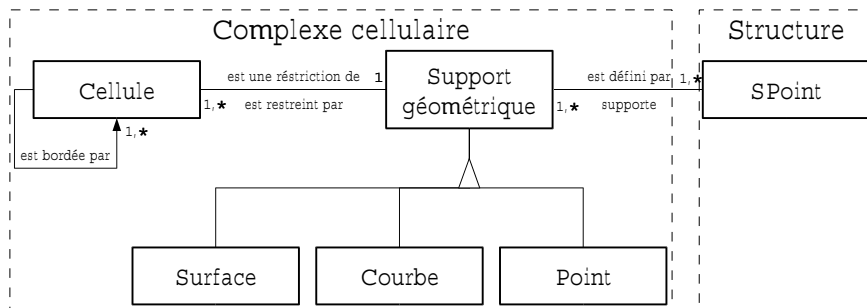


FIG. 5.6: Diagramme de classes partiel d’un modèle géométrique

Cette séparation présente l’intérêt de définir plusieurs restrictions distinctes sur un même support. En outre, elle permet l’organisation des opérateurs en deux classes. Les opérateurs topologiques qui ne manipulent que le graphe d’adhérence et sont le plus souvent appliqués à des cellules de même dimension. Les opérateurs géométriques sont responsables de calculs numériques sur les supports.

5.3.2 Le support

Chaque objet élémentaire peut être identifié dans l’espace par la position d’un ensemble de points et d’autres caractéristiques supplémentaires. En effet, les

éléments canoniques sont parfaitement connus à partir d'un ensemble de points et de paramètres. Par exemple, le cercle est défini par son centre et la valeur de son rayon. Les autres éléments géométriques utilisés en CAO sont généralement modélisés par des courbes ou des carreaux paramétriques de la famille des NURBS.

La structure qui vient d'être présentée permet de positionner des **spoints** les uns par rapport aux autres, par une approche déclarative. Connaissant la position d'un ensemble de **spoints**, il suffit alors de leur rattacher les différents éléments géométriques définissant l'objet (Figure 5.6).

Les objets géométriques les plus élémentaires utilisés en CAO sont le point et le segment. Un point est simplement associé à un **spoint** de la structure. Le segment qui est une portion finie de droite est déterminé par ses deux extrémités qui sont alors rattachées à deux **spoints**. Le cercle, les coniques, les courbes et surfaces gauches regroupent les autres entités utilisées en CFAO. Elles seront décrites en détail au chapitre 6.

5.4 Conclusion

Ce chapitre a permis de détailler la structure de données du modèle géométrique. Cette structure basée sur une modélisation volumique de type B-rep a surtout l'avantage de décrire la géométrie par une approche déclarative originale. Cette approche va permettre une *bonne* mise en équations du problème (Cf. section 8.3) mais avant tout, la structure élémentaire et la toile de chaque spécification géométrique doivent être présentées.

Chapitre 6

Les éléments géométriques

Ce chapitre présente la structure et la toile des entités géométriques élémentaires (point, droite, segment, plan), des courbes planes et des coniques.

6.1 Introduction

Le chapitre précédent a présenté le modèle géométrique global comme étant composé de deux entités : la structure et la toile. Un objet géométrique est construit petit à petit par adjonction d'éléments. L'ajout d'un élément modifie le modèle d'une façon bien précise. En effet, chaque élément possède une structure et une toile minimale. L'utilisation de quelques règles de construction permet de compléter le modèle. Ce chapitre décrit la structure et la toile de chaque entité élémentaire.

6.2 Les objets élémentaires

6.2.1 Le point

Le point est un élément de l'espace affine de dimension zéro. Il est rarement utilisé seul en CAO. Il accompagne surtout les différents éléments afin de préciser leurs points caractéristiques. Par exemple, les points d'un segment peuvent être ses deux extrémités. Quelques fois, le milieu d'un segment est aussi indiqué. Le centre d'un cercle ou le foyer d'une parabole sont aussi des points importants pour l'utilisateur. Un point, qu'il soit seul ou qu'il soit caractéristique d'un élément, est composé d'un point de la toile rattaché à un `spoint` de la structure.

6.2.2 La droite, le segment de droite

La droite est un élément de l'espace affine de dimension un. C'est un élément très utile en CAO pour préciser les différents axes importants d'une pièce ou d'un élément d'une pièce : axe de symétrie, axe de rotation. Il peut aussi définir des traits de construction.

Le segment qui est un arc de droite doit être l'élément le plus utilisé dans un modèle géométrique. En effet, la plupart des composants mécaniques sont des pièces principalement polyédriques avec quelques zones cylindriques et sphériques. Ces zones sont dues à un habillage des surfaces non fonctionnelles par des surfaces de raccordement.

La toile d'une droite \mathcal{D} est une droite qui sera attachée soit à un **spoint** \mathbf{A} et la direction sera définie par un **vecteur** \mathbf{e}_1 (Figure 6.1-a), soit à deux **spoints** (\mathbf{A} , \mathbf{B}) existants (Figure 6.1-b). Le segment sera quant à lui, attaché à deux **spoints** (\mathbf{A} , \mathbf{B}) existants (Figure 6.1-c).

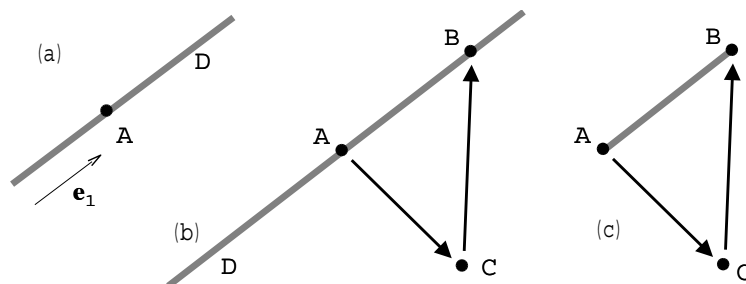


FIG. 6.1: Structure minimale et toile d'une droite et d'un segment

6.2.3 Le plan

Le plan est un élément de l'espace affine de dimension deux. Comme la plupart des composants mécaniques sont des pièces principalement polyédriques, le plan est l'élément support de la plupart de leurs faces.

La toile d'un plan \mathcal{P} est modélisée par une surface de Bézier de degré un. Les pôles de la surface sont attachés à des **spoints** coplanaires. Ces **spoints** appartiennent donc à un **plan affine** \mathcal{P}^A porté par un **plan vectoriel** \mathcal{P}^V de normale n . La position de ces **spoints** dans le plan \mathcal{P}^V est connue grâce aux **bipoints** définissant le réseau caractéristique. Cet ensemble est la structure d'un plan (Figure 6.2).

6.3 Les arcs de courbes

Les courbes ou plus particulièrement les arcs de courbes, sont très utilisés en CAO. En raison de leur nombre de degrés de liberté théoriquement infini, ces

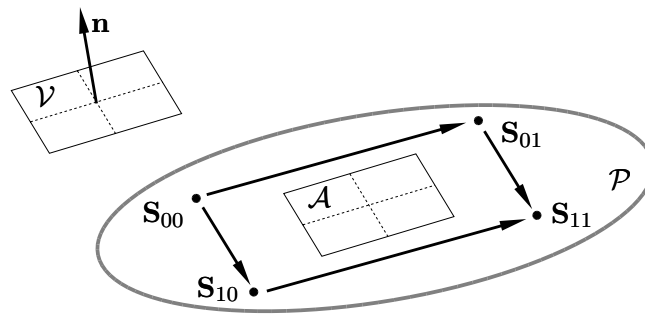


FIG. 6.2: Structure minimale et toile d'un plan

entités seront surtout utilisées par des designers à des fins esthétiques, mais elles peuvent aussi représenter des entités fonctionnelles.

Les courbes sont modélisées par des courbes paramétriques de degrés finis. Les modèles couramment employés sont les modèles de la famille des NURBS tels que les courbes de Bézier.

Ces modèles sont utilisés en raison de leurs propriétés géométriques simples exploitables par l'utilisateur. En effet, le comportement de ces courbes est contrôlé à partir de grandeurs géométriques faciles d'accès comme le *polygone caractéristique* dont la forme est en relation avec la courbe. Comme l'illustre la figure 6.3, la prédiction de la déformation d'une courbe suite à une modification de ce réseau est alors intuitive pour l'utilisateur.

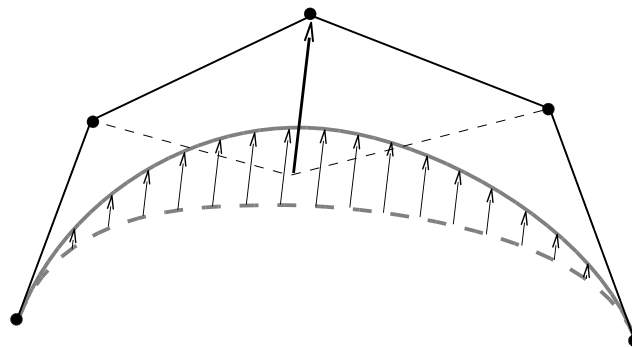


FIG. 6.3: Influence de la modification du polygone caractéristique sur la courbe

Cette section montre, à partir de la présentation des courbes de Bézier, pourquoi les courbes NURBS sont adéquates pour le modèle géométrique. La structure d'une courbe est alors proposée. Ensuite, les différentes fonctions de base et leurs propriétés sont rappelées. Enfin, cette section se termine sur la structure d'un point courant d'une courbe et de sa tangente.

6.3.1 Structure et toile d'une courbe

Une courbe NURBS est classiquement associée à son réseau caractéristique. L'utilisation de ce polygone est due à P. Bézier. En effet, la première formulation de sa famille de courbes¹ est basée sur ce réseau [Bézier 1987].

Dans un référentiel orthonormé $(O, \mathbf{e}_1, \dots, \mathbf{e}_m)$ de dimension m , il définit une *courbe de base* C . Les points $\mathbf{P}(u)$ de cette courbe sont exprimés par :

$$\mathbf{P}(u) = \sum_{i=1}^m \mathbf{e}_i f_{i,m}(u), \quad (6.1)$$

avec $f_{i,m}$ des fonctions de base. Cette courbe est ensuite plongée dans un espace de travail de dimension deux ou trois par une transformation linéaire telle que les vecteurs $\mathbf{e}_1, \dots, \mathbf{e}_m$ deviennent respectivement $\mathbf{a}_1, \dots, \mathbf{a}_m$ et une translation caractérisée par un vecteur \mathbf{a}_0 . La courbe C est alors définie par l'ensemble des points $\mathbf{P}(u)$ répondant à l'équation (6.2). Les vecteurs \mathbf{a}_i sont préférés être placés bout à bout. Ils constituent ainsi un polygone, dit "caractéristique", définissant la courbe C (Figure 6.4).

$$\mathbf{P}(u) = \mathbf{a}_0 + \sum_{i=1}^m \mathbf{a}_i f_{i,m}(u), \quad u \in [0, 1]. \quad (6.2)$$

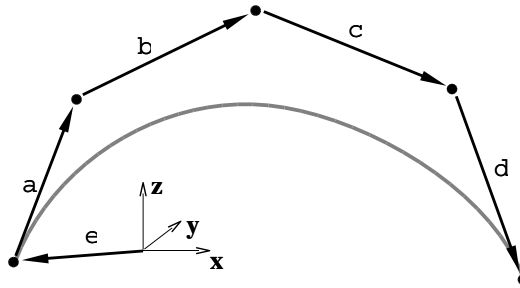


FIG. 6.4: Le polygone caractéristique d'une courbe de Bézier selon P. Bézier

À cette formulation initiale, est associée une autre formulation basée sur les polynômes de Bernstein en raison de leurs nombreuses propriétés [Bézier 1987] [Léon 1991]. Elle utilise non plus les barres \mathbf{a}_i du réseau mais les sommets \mathbf{S}_i de ce polygone (Équation (6.3), Figure 6.5).

$$\mathbf{P}(u) = \sum_{i=0}^m \mathbf{S}_i B_{i,m}(u), \quad u \in [0, 1]. \quad (6.3)$$

¹Le développement de ce premier modèle de courbe a été réalisé de façon indépendante par P. Bézier [Bézier 1987] alors ingénieur chez Renault et P. de Casteljau [de Casteljau 1986] chez Citroën.

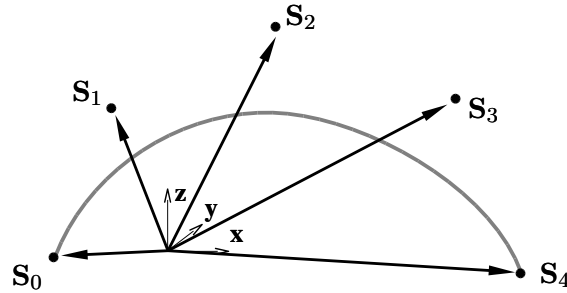


FIG. 6.5: Formulation conventionnelle d'une courbe de Bézier

Avec cette nouvelle formulation, le réseau n'a donc plus la même signification mathématique, mais il reste très utile pour analyser et modifier une courbe. En effet, de nombreuses propriétés des courbes NURBS² sont déductibles de ce polygone [Léon 1991] :

- une courbe $\mathbf{P}(u)$ commence au sommet \mathbf{S}_0 et se termine au sommet \mathbf{S}_m ;
- la direction de la tangente à la courbe $\mathbf{P}(u)$ en $u = 0$ (respectivement $u = 1$) est définie par le côté \mathbf{a}_1 (respectivement \mathbf{a}_m) ;
- la concavité de la courbe $\mathbf{P}(u)$ en $u = 0$ (respectivement $u = 1$) est définie qualitativement à partir de la direction des côtés \mathbf{a}_1 et \mathbf{a}_2 (respectivement \mathbf{a}_m et \mathbf{a}_{m-1}).

En raison de la similitude du polygone caractéristique avec sa courbe et de l'ensemble des propriétés géométriques de la courbe déductible de son polygone, la structure choisie pour une courbe est son polygone caractéristique (Figure 6.6).

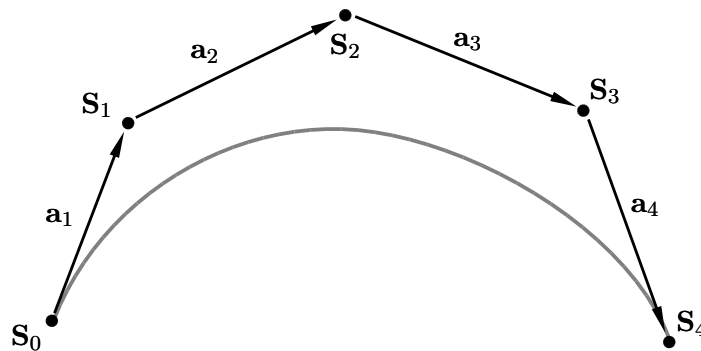


FIG. 6.6: Structure et toile d'une courbe NURBS

6.3.2 Les fonctions $f_{i,m}$

Les fonctions $f_{i,m}$ ont été introduites par P. Bézier [Bézier 1987] pour la définition de son modèle de courbe paramétrique. Elles sont utilisées dans de nombreuses

²les coordonnées homogènes des différents pôles d'une courbes NURBS sont, par convention pour la CAO, positives.

relations associées à la structure d'une courbe de Bézier. Pour cette raison, cette section présente la forme générale de cette famille de fonctions ainsi que ses différentes propriétés et ses relations avec les polynômes de Bernstein.

Équations

La famille de fonction $f_{i,m}$ a été obtenue en écrivant les propriétés attribuées à la courbe de base de dimension 3 [Bézier 1987]. Ces propriétés sont :

- le point $\mathbf{C}(0)$ est l'origine \mathbf{O} du référentiel. Ses coordonnées sont $(0, 0, 0)$;
- les coordonnées du point $\mathbf{C}(1)$ sont $(1, 1, 1)$;
- la tangente en $\mathbf{C}(0)$ (respectivement $\mathbf{C}(1)$) est \mathbf{e}_1 (respectivement \mathbf{e}_3) ;
- la courbe est définie pour $u \in [0, 1]$;
- la courbe est osculatrice au plan $(O, \mathbf{e}_1, \mathbf{e}_2)$ au point $(0, 0, 0)$ et au plan $(O, \mathbf{e}_2, \mathbf{e}_3)$ au point $(1, 1, 1)$.

La généralisation de ces conditions à une courbe de base de dimension m a permis d'obtenir les $m(m-1)$ conditions définissant les $(m+1)$ fonctions $f_{i,m}(u)$:

$$f_{0,m}(u) = 1, \quad u \in [0, 1] ; \quad (6.4)$$

$$f_{i,m}(0) = 0, \quad i \in \{1, \dots, m\} ; \quad (6.5)$$

$$f_{i,m}(1) = 1, \quad i \in \{1, \dots, m\} ; \quad (6.6)$$

$$\frac{\partial^j f_{i,m}}{\partial u^j}(0) = 0, \quad i \in \{j+1, m\}, \quad j \in \{1, \dots, m-1\} ; \quad (6.7)$$

$$\frac{\partial^j f_{i,m}}{\partial u^j}(1) = 0, \quad i \in \{1, m-j\}, \quad j \in \{1, \dots, m-1\}. \quad (6.8)$$

Ces fonctions sont :

$$f_{i,m}(u) = \frac{(-u)^i}{(i-1)!} \cdot \frac{\partial^{i-1} \left(\frac{(1-u)^m - 1}{u} \right)}{\partial u^{i-1}}. \quad (6.9)$$

Elles peuvent aussi s'exprimer par :

$$f_{i,m}(u) = \sum_{j=0}^m (-1)^{i+j} \cdot C_m^j \cdot C_{j-1}^{i-1} \cdot u^j ; \quad (6.10)$$

$$f_{i,m}(u) = \sum_{j=i}^m (-1)^{i+j} \frac{m!}{j \cdot (i-1)! (j-i)! (m-j)!} u^j. \quad (6.11)$$

Propriétés des fonctions $f_{i,m}$

Parmi les nombreuses propriétés des fonctions $f_{i,m}$, on notera celles qui ont des applications pratiques :

- les fonctions $f_{0,m}$ sont strictement égales à 1 ;

- les fonctions $f_{i,m}$ sont symétriques par permutation entre u et $(1-u)$, soit :

$$f_{i,m}(1-u) = 1 - f_{m-i+1,m}(u) ; \quad (6.12)$$

- les fonctions $f_{i,m}$, $i \neq 0$, sont nulles pour $u = 0$ et égales à 1 pour $u = 1$;
- les dérivées premières sont toutes nulles aux extrémités sauf les dérivées de $f_{1,m}$ en $u = 0$ (Cf. éq. (6.7)) et de $f_{m,m}$ en $u = 1$ (Cf. éq. (6.8)), soit :

$$\frac{\partial f_{1,m}}{\partial u}(0) = \frac{\partial f_{m,m}}{\partial u}(1) = m ; \quad (6.13)$$

- pour les mêmes raisons, seules les dérivées secondes des deux premières fonctions (respectivement des deux dernières) ne sont pas nulles en $u = 0$ (respectivement en $u = 1$) soit :

$$\frac{\partial^2 f_{1,m}}{\partial u^2}(0) = -\frac{\partial^2 f_{2,m}}{\partial u^2}(0) = -m(m-1) ; \quad (6.14)$$

$$\frac{\partial^2 f_{m,m}}{\partial u^2}(1) = -\frac{\partial^2 f_{m-1,m}}{\partial u^2}(1) = m(m-1) . \quad (6.15)$$

Relation entre les fonctions $f_{i,m}$ et $B_{i,m}$

Les courbes de Bézier sont exprimables dans deux bases de fonctions polynomiales (Cf. éq. (6.2) et (6.3)). Il existe des relations liant ces deux familles. L'emploi de certaines permet de simplifier l'expression de certaines propriétés géométriques de la courbe :

$$f_{i,m}(u) = \sum_{j=i}^m B_{j,m}(u) ; \quad (6.16)$$

$$B_{i,m}(u) = f_{i,m}(u) - f_{i+1,m}(u) ; \quad (6.17)$$

$$\frac{\partial f_{i,m}}{\partial u}(u) = m B_{i-1,m-1}(u) ; \quad (6.18)$$

$$\frac{\partial^2 f_{i,m}}{\partial u^2}(u) = m(m-1)(B_{i-1,m-2}(u) - B_{i,m-2}(u)) . \quad (6.19)$$

Généralisation aux B-Splines non uniformes et aux NURBS

Il n'existe pas de fonctions équivalentes aux fonctions $f_{i,m}(u)$ pour les autres modèles de courbes. La généralisation de cette approche à ces modèles passera donc par la définition algébrique de toutes les fonctions nécessaires. Cette partie introduit cette généralisation. Pour le reste du document, les courbes utilisées sont généralement des NURBS. Dans le cas où la généralisation au NURBS n'est pas directe pour certains points étudiés, les courbes seront alors des courbes de Bézier ou B-Spline non uniforme.

Une courbe NURBS de degré m est définie dans l'espace homogène par le polygone caractéristique \mathbf{S}_i^h , $i \in \{0, \dots, n\}$, la séquence de coordonnées homogènes

h_i , $i \in \{0, \dots, n\}$ et la séquence nodale u_i , $i \in \{0, \dots, (m+n+1)\}$. Le point $\mathbf{P}^h(u)$ d'une courbe NURBS dans l'espace homogène est défini par la relation :

$$\mathbf{P}^h(u) = \sum_{i=0}^n \mathbf{S}_i^h N_{i,m}(u), \quad u \in [u_0, u_{n+m+1}]. \quad (6.20)$$

La forme rationnelle d'une courbe s'obtient en exprimant l'équation 6.20 à l'aide de l'application projective homogène [Léon 1991], soit :

$$\mathbf{P}(u) = \sum_{i=0}^n \mathbf{S}_i \frac{h_i N_{i,m}(u)}{\sum_{j=0}^n h_j N_{j,m}(u)}, \quad u \in [u_0, u_{n+m+1}]. \quad (6.21)$$

Pour revenir à une formule similaire à celle des courbes de Bézier (Équation (6.3)), les fonctions $N_{i,m}^*(u)$ sont définies comme les fonctions de base des courbes NURBS avec :

$$N_{i,m}^*(u) = \frac{h_i N_{i,m}(u)}{\sum_{j=0}^n h_j N_{j,m}(u)}. \quad (6.22)$$

Un point d'une courbe NURBS est alors défini par la relation :

$$\mathbf{P}(u) = \sum_{i=0}^n \mathbf{S}_i N_{i,m}^*(u), \quad u \in [u_0, u_{n+m+1}]. \quad (6.23)$$

Par analogie avec les courbes de Bézier, on définit les fonctions de base $F_{i,m}^*(u)$ et $F_{i,m}(u)$ telles qu'un point d'une courbe NURBS vérifie alors la relation (6.24) et un point d'une courbe B-spline non uniforme, la relation (6.25), soit :

$$\mathbf{P}(u) = \sum_{i=0}^n \mathbf{a}_i F_{i,m}^*(u), \quad u \in [u_0, u_{n+m+1}] ; \quad (6.24)$$

$$\mathbf{P}(u) = \sum_{i=0}^n \mathbf{a}_i F_{i,m}(u), \quad u \in [u_0, u_{n+m+1}]. \quad (6.25)$$

Les fonctions $F_{i,m}^*(u)$ et $N_{i,m}^*(u)$ vérifient alors des relations similaires aux relations (6.16) et (6.17), soit :

$$F_{i,m}^*(u) = \sum_{j=i}^n N_{j,m}^*(u) ; \quad (6.26)$$

$$N_{i,m}^*(u) = F_{i,m}^*(u) - F_{i+1,m}^*(u). \quad (6.27)$$

Les dérivées d'une courbe rationnelle n'ont pas de propriété qui permettrait d'obtenir des relations équivalentes aux relations (6.18) et (6.19). En pratique, ces dérivées n'ont pas d'intérêt pour notre modèle puisque les courbes NURBS ne

sont pas utilisées en tant que tel en CAO. Pour une B-spline non uniforme, ces relations deviennent :

$$\frac{\partial F_{i,m}}{\partial u}(u) = m \frac{N_{i,m-1}(u)}{u_{i+m} - u_i}, \quad u \in [u_1, u_{n+m}] ; \quad (6.28)$$

$$\frac{\partial^2 F_{i,m}}{\partial u^2}(u) = \frac{m(m-1)}{u_{i+m} - u_i} \left(\frac{N_{i-2,m-2}(u)}{u_{i-1+m} - u_i} - \frac{N_{i-1,m-2}(u)}{u_{i+m} - u_{i+1}} \right), \quad (6.29)$$

$$u \in [u_2, u_{n+m-1}].$$

6.3.3 Point courant

La structure d'une courbe est définie par le polygone caractéristique de celle-ci. Un point de la courbe doit donc, lui aussi, être déduit de cette structure. En utilisant la formule originale de P. Bézier, un point $\mathbf{P}(u)$ d'une courbe NURBS est défini par (6.24). Afin de déterminer la structure du point courant d'une courbe, il a fallu définir une nouvelle entité, le *multi-point*, qui puisse lier un point de la courbe à son polygone caractéristique.

le **multi-point** est un **pointlink** qui lie le premier point d'une courbe au point courant (Figure 6.7). Il connaît les listes ordonnées des bipoints du polygone caractéristique et des fonctions de base $N_{i,m}^*(u)$, la coordonnée du point courant sur la courbe et son vecteur support. On le note $\widetilde{\mathbf{S}_0\mathbf{P}(u)}$.

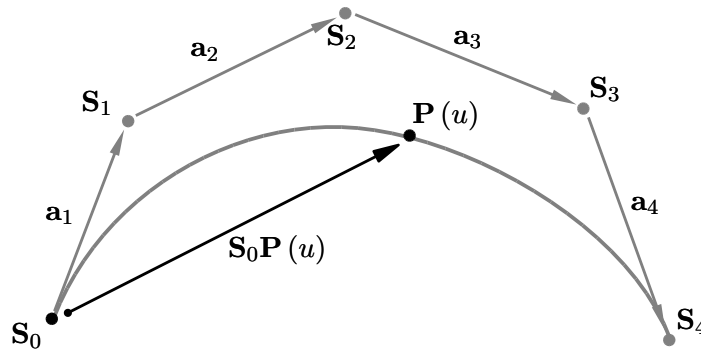


FIG. 6.7: Structure d'un point courant d'une courbe

6.3.4 La tangente courante

Par définition, le vecteur tangent \mathbf{t} à $\mathbf{P}(u)$ répond à l'équation :

$$\mathbf{t} = \frac{\partial \mathbf{P}}{\partial s} = \frac{\partial \mathbf{P}}{\partial u} \left(\frac{\partial u}{\partial s} \right), \quad \forall u \in [0, 1] \quad (6.30)$$

où s est l'abscisse curviligne d'un point de la courbe et $\partial u / \partial s$ un scalaire. Donc, seul le vecteur $\partial \mathbf{P} / \partial u$ définit la direction du vecteur tangent \mathbf{t} .

La tangente aux extrémités

La direction du vecteur tangent à la courbe en $u = u_0$ et en $u = u_{n+m+1}$ est définie par le premier côté du polygone caractéristique, \mathbf{a}_1 et par le dernier côté, \mathbf{a}_n , lorsqu'ils ne sont pas nuls. En effet, pour une courbe NURBS de degré m , le comportement de la courbe à sa première extrémité par rapport à son polygone caractéristique est régi par les équations [Léon 1991] :

$$\mathbf{P}(u_0) = \mathbf{S}_0 ; \quad (6.31)$$

$$\frac{\partial \mathbf{P}}{\partial u}(u_1) = \frac{h_1 m}{h_0 (u_0 - u_1)} \mathbf{a}_1. \quad (6.32)$$

Ce qui définit la direction du vecteur tangent à la courbe en u_0 . Par symétrie, le vecteur tangent en u_n est donc porté par le vecteur de \mathbf{a}_n .

La tangente courante

Pour la tangente courante, seules les courbes non rationnelles sont traitées. Le vecteur tangent est porté par le vecteur $d\mathbf{P}/du$. Or, en utilisant la relation (6.18) pour les courbes de Bézier et la relation (6.28) pour les courbes B-Splines non uniformes, le vecteur tangent est porté par un vecteur vérifiant respectivement les relations (6.33) et (6.34) :

$$\frac{\partial \mathbf{P}}{\partial u}(u) = \sum_{i=0}^m \mathbf{S}_i \frac{\partial B_{i,m}}{\partial u}(u) = m \sum_{i=1}^m \mathbf{a}_i B_{i-1,m-1}(u) ; \quad (6.33)$$

$$\begin{aligned} \frac{\partial \mathbf{P}}{\partial u}(u) &= \sum_{i=0}^n \mathbf{a}_i \frac{\partial F_{i,m}}{\partial u}(u) ; \\ &= m \sum_{i=0}^n \mathbf{a}_i \frac{N_{i,m-1}(u)}{u_{i+m} - u_i}, \quad u \in [u_1, u_{n+m}]. \end{aligned} \quad (6.34)$$

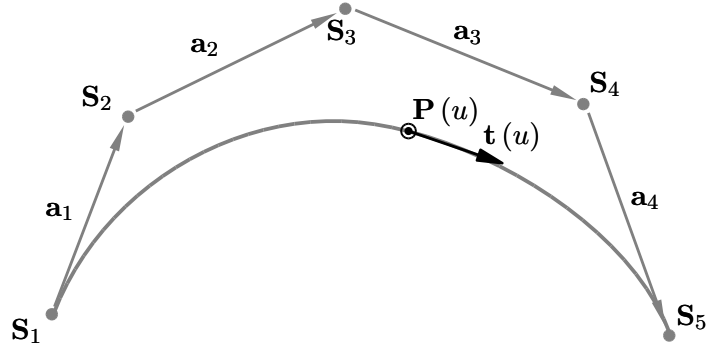
La structure du vecteur tangent à un point $\mathbf{P}(u)$ d'une courbe peut alors être défini par un multipoint d'origine $\mathbf{P}(u)$ vérifiant ces deux relations (Figure 6.8).

6.3.5 Le rayon de courbure

Par définition, le rayon de courbure d'une courbe répond à l'équation :

$$R = \frac{\left\| \frac{\partial \mathbf{P}}{\partial u}(u) \right\|^3}{\left\| \frac{\partial \mathbf{P}}{\partial u}(u) \wedge \frac{\partial^2 \mathbf{P}}{\partial u^2}(u) \right\|}. \quad (6.35)$$

Or, nous avons vu les expressions de $\frac{\partial \mathbf{P}}{\partial u}(u)$ pour les courbes de Bézier (Cf. éq. (6.33)) et pour les courbes B-Splines non uniformes (Cf. éq. (6.34)). À partir


 FIG. 6.8: Structure du vecteur tangent à un point $\mathbf{P}(u)$ d'une courbe

des relations (6.19) et (6.29), les expressions des dérivées secondes de ces courbes deviennent :

$$\frac{\partial^2 \mathbf{P}}{\partial u^2}(u) = m(m-1) \sum_{i=1}^m (B_{i-1,m-2}(u) - B_{i,m-2}(u)) \mathbf{a}_i, \quad (6.36)$$

pour une courbe de Bézier et :

$$\frac{\partial^2 \mathbf{P}}{\partial u^2}(u) = \sum_{i=0}^n \frac{m(m-1)}{u_{i+m} - u_i} \left(\frac{N_{i-2,m-2}(u)}{u_{i-1+m} - u_i} - \frac{N_{i-1,m-2}(u)}{u_{i+m} - u_{i+1}} \right) \mathbf{a}_i, \quad (6.37)$$

pour une courbe B-Spline non uniforme sous certaines conditions. Ces expressions se simplifient aux extrémités pour devenir en $u = 0$:

$$\frac{\partial^2 \mathbf{P}}{\partial u^2}(u) = m(m-1) (\mathbf{a}_2 - \mathbf{a}_1), \quad (6.38)$$

pour une courbe de Bézier et :

$$\frac{\partial^2 \mathbf{P}}{\partial u^2}(u) = \frac{m(m-1)}{(u_{2+m} - u_2)(u_{1+m} - u_2)} (\mathbf{a}_2 - \mathbf{a}_1), \quad (6.39)$$

pour une courbe B-Spline non uniforme.

À partir des relations (6.32) et (6.38), le rayon de courbure en $\mathbf{P}(0)$ d'une courbe de Bézier de degré m devient :

$$R = \frac{m \|\mathbf{a}_1\|^3}{(m-1) \|\mathbf{a}_1 \wedge \mathbf{a}_2\|}, \quad (6.40)$$

alors que pour une courbe B-Spline non uniforme de degré m , ce rayon de courbure est obtenu à partir des relations (6.32), (6.39) et devient en $\mathbf{P}(0)$:

$$R = \frac{\frac{m}{(u_0 - u_1)^2} \|\mathbf{a}_1\|^3}{\frac{(m-1)}{(u_{2+m} - u_2)(u_{1+m} - u_2)} \|\mathbf{a}_1 \wedge \mathbf{a}_2\|}. \quad (6.41)$$

6.4 Les arcs de coniques

Comme leur nom le suggère, les coniques sont des courbes obtenues par l'intersection d'un cône de révolution à l'aide d'un plan. Actuellement ces courbes sont encore très utilisées en conception mécanique pour leurs propriétés géométriques très appréciées dans certains domaines comme l'optique ou l'aérodynamique.

En raison de leurs degrés de liberté (différents de 2 en 2D), les coniques sont difficilement spécifiables par les approches classiques (Cf. section 3.3). Quelques travaux ont tenté d'apporter des solutions [Fudos *et al.* 1996–2], mais les schémas de spécifications proposés sont très restreints puisque les extrémités et les tangentes aux extrémités doivent être contraintes.

Afin d'offrir à l'utilisateur une complète liberté sur ses choix de spécification, les coniques sont définies dans le modèle géométrique en utilisant toutes leurs caractéristiques. Pour justifier leur structure, elles seront présentées dans un premier temps par une définition mathématique et un modèle géométrique. Ensuite, chaque famille sera étudiée indépendamment.

6.4.1 Les coniques

Définition mathématique

Une conique est le lieu géométrique des points \mathbf{M} vérifiant la relation (6.42) avec \mathbf{F} le foyer, \mathbf{P} la projection orthogonale de \mathbf{M} sur une droite \mathcal{D} appelée directrice, et e l'excentricité de la conique (Figure 6.9). La droite passant par \mathbf{F} et orthogonale à la directrice est appelée *axe focal* de la conique.

$$\frac{\|\mathbf{MF}\|}{\|\mathbf{MP}\|} = e. \quad (6.42)$$

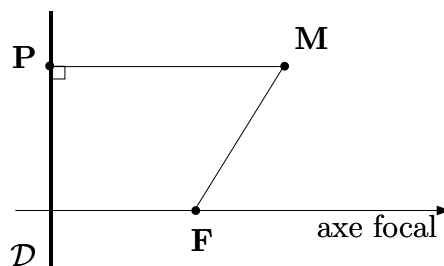


FIG. 6.9: Définition d'une conique par ces caractéristiques intrinsèques

Modèle géométrique

Les courbes de Bézier rationnelles quadratiques sont des arcs de coniques [Lee 1987]. Afin que les coniques soient homogènes aux autres courbes utilisées en CAO, ce modèle sera utilisé pour les décrire.

Ces courbes sont définies par un polygone caractéristique $\{\mathbf{S}_0, \mathbf{S}_1, \mathbf{S}_2\}$ et une séquence de coordonnées homogènes $\{h_0, h_1, h_2\}$.

Dans le but de relier la courbe de Bézier rationnelle aux caractéristiques intrinsèques d'une conique, Lee [Lee 1987] définit les scalaires κ, ν et les vecteurs \mathbf{u} et \mathbf{v} tels que :

$$\mathbf{u} = \mathbf{S}_1 \mathbf{S}_0 ; \quad (6.43)$$

$$\mathbf{v} = \mathbf{S}_1 \mathbf{S}_2 ; \quad (6.44)$$

$$\kappa = \frac{h_0 h_2}{4h_1^2} ; \quad (6.45)$$

$$\nu = \frac{2\kappa}{4\kappa - 1} ; \quad (6.46)$$

$$(6.47)$$

et $\lambda_1 < \lambda_2$ solutions de :

$$\|\mathbf{u} \wedge \mathbf{v}\|^2 \lambda^2 - 2(\kappa \|\mathbf{u} - \mathbf{v}\|^2 + \mathbf{u} \cdot \mathbf{v}) \lambda + 4\kappa - 1 = 0. \quad (6.48)$$

Le type de la conique est déterminé entièrement par la séquence de coordonnées homogènes. En effet, si :

- $4\kappa > 1$: la courbe est une ellipse ;
- $4\kappa = 1$: la courbe est une parabole ;
- $4\kappa < 1$: la courbe est une hyperbole.

Afin de simplifier les traitements, tout en ne réduisant pas les possibilités de ces courbes, les coordonnées homogènes h_0 et h_2 sont imposées comme étant égales à 1. Ainsi, une conique est une ellipse si $h_1 < 1$, une parabole si $h_1 = 1$ et une hyperbole si $h_1 > 1$.

6.4.2 Arc de parabole

Définition mathématique

Une parabole est une conique dont l'excentricité e est égale à 1. Elle est donc le lieu géométrique des points \mathbf{M} équidistants du foyer \mathbf{F} et de la directrice \mathcal{D} (Équation 6.49). Soit \mathbf{S} le sommet de la parabole, alors $\|\mathbf{SF}\|$ définit la distance focale f . Il est intéressant de remarquer que la tangente \mathbf{t}_m à la courbe en \mathbf{M} est portée par la bissectrice de $(\widehat{\mathbf{MF}, \mathbf{MP}})$ (Figure 6.10).

$$\|\mathbf{MF}\| = \|\mathbf{MP}\|. \quad (6.49)$$

Modèle géométrique

Une courbe de Bézier (non rationnelle) quadratique est un arc de parabole. Le passage d'un arc de parabole à la courbe de Bézier équivalente est donc direct. En

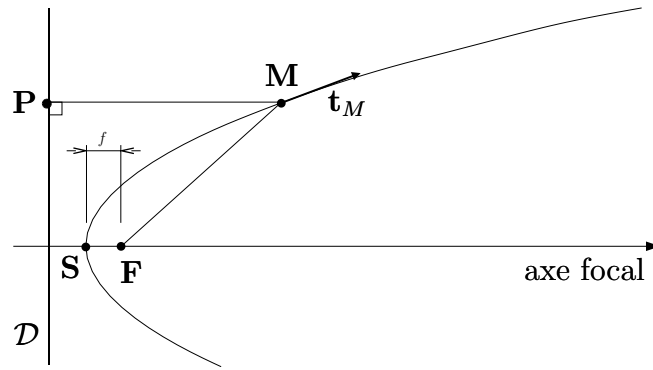


FIG. 6.10: Une parabole

effet, il suffit de définir le second pôle de la courbe qui appartient aux tangentes des points des extrémités de l'arc (Figure 6.11).

Connaissant une courbe de Bézier quadratique, les éléments caractéristiques se déduisent du polygone caractéristique par la relation (6.50) pour le foyer \mathbf{F} et la relation (6.51) pour le sommet \mathbf{S} :

$$\mathbf{S}_1\mathbf{F} = \frac{\gamma\mathbf{u} + \alpha\mathbf{v}}{\zeta} ; \quad (6.50)$$

$$\mathbf{S}_1\mathbf{S} = \left(\frac{\gamma + \beta}{\zeta}\right)^2 \mathbf{u} + \left(\frac{\alpha + \beta}{\zeta}\right)^2 \mathbf{v} ; \quad (6.51)$$

avec $\alpha = \|\mathbf{u}\|^2$, $\beta = \mathbf{u} \cdot \mathbf{v}$, $\gamma = \|\mathbf{v}\|^2$ et $\zeta = \alpha + \gamma + 2\beta = \|\mathbf{u} + \mathbf{v}\|^2$.

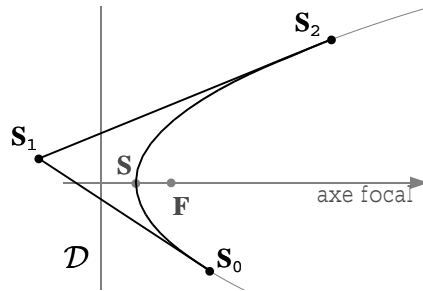


FIG. 6.11: Arc de parabole porté par une courbe de Bézier quadratique.

Structure

La structure d'un arc de parabole (Figure 6.12) regroupe les propriétés mathématiques des points des extrémités et du sommet ainsi que le polygone caractéristique de la courbe de Bézier. Ainsi, les **bipoints** \mathbf{OS} et \mathbf{SF} permettent de positionner le sommet \mathbf{S} et le foyer \mathbf{F} sur l'axe focal par rapport à la directrice \mathcal{D} . Pour cela, ces deux **bipoints** sont portés par le même **vecteur support** \mathbf{e}_1 . La direction de celui-ci par rapport à celle de la directrice \mathcal{D} définie par le **vecteur**

\mathbf{e}_2 étant imposée par un angle droit :

$$\widehat{(\mathbf{e}_1, \mathbf{e}_2)} = \frac{\pi}{2}. \quad (6.52)$$

De plus, afin de vérifier la définition mathématique d'une parabole (Éq. 6.49), les deux **bipoints** ont la même **variable**.

Pour les mêmes raisons, chaque extrémité $\mathbf{M}_i, i=\{1,2\}$ de l'arc de parabole est lié au foyer par le **bipoint** $\mathbf{M}_i\mathbf{F}$ et à la directrice \mathcal{D} par le **bipoint** $\mathbf{M}_i\mathbf{P}_i$ porté par le vecteur support \mathbf{e}_1 , le **spoint** \mathbf{P}_i définissant la projection orthogonale de \mathbf{M}_i sur la directrice \mathcal{D} . Le **bipoint** $\mathbf{P}_i\mathbf{O}$ porté par le vecteur support \mathbf{e}_2 assure la coïncidence de \mathbf{P}_i sur la directrice \mathcal{D} . Il suffit alors d'assurer la définition mathématique d'une parabole (Éq. 6.49), pour cela les bipoints $\mathbf{M}_i\mathbf{P}_i$ et $\mathbf{M}_i\mathbf{F}$ ont une mesure commune.

L'arc de parabole étant modélisé par une courbe de Bézier, il ne reste plus qu'à lier cette première partie de structure à celle d'une courbe de Bézier de degré deux. Pour cela, le polygone caractéristique d'une courbe de degré deux défini par le **bipoint** $\mathbf{M}_1\mathbf{S}_1$ porté par le vecteur \mathbf{e}_5 et le **bipoint** $\mathbf{M}_2\mathbf{S}_1$ porté par le vecteur \mathbf{e}_6 est ajouté. Or, les vecteurs supports de ces **bipoints** définissent la tangente de l'arc aux extrémités \mathbf{M}_1 et \mathbf{M}_2 . Comme la tangente \mathbf{t}_m à la courbe en un point \mathbf{M} est portée par la bissectrice de $\widehat{(\mathbf{MF}, \mathbf{MP})}$ (Figure 6.10), il suffit de lier la direction des vecteurs \mathbf{e}_5 et \mathbf{e}_6 aux autres vecteurs de la structure par les relations ci-dessous pour assurer la cohérence globale :

$$\widehat{(\mathbf{e}_1, \mathbf{e}_3)} = \widehat{(\mathbf{e}_3, \mathbf{e}_5)} ; \quad (6.53)$$

$$\widehat{(\mathbf{e}_1, \mathbf{e}_4)} = \widehat{(\mathbf{e}_4, \mathbf{e}_6)}. \quad (6.54)$$

Cette structure contient un ensemble de points caractéristiques comme les extrémités de l'arc de parabole, le sommet, le foyer et les pôles du polygone caractéristique. Certains bipoints représentent des propriétés importantes de l'arc de parabole comme les tangentes aux extrémités ou la distance focale de la parabole. L'utilisateur dispose donc d'une définition complète d'un arc parabole qu'il peut donc librement contraindre.

6.4.3 Arc d'ellipse

Définition mathématique

Une ellipse est une conique dont l'excentricité e est inférieure à 1. Elle admet deux axes de symétrie : l'axe focal aussi appelé *grand axe* et la droite, appelée *petit axe*, perpendiculaire à l'axe focal et passant par le milieu du segment reliant les deux sommets A_1 et A_2 (Figure 6.13). Soit \mathbf{C} le centre de l'ellipse. Soit B_1 et B_2 , les deux sommets du petit axe, on définit alors les scalaires a , b et c tels que :

$$a = \|\mathbf{CA}_1\| = \|\mathbf{CA}_2\| ; \quad (6.55)$$

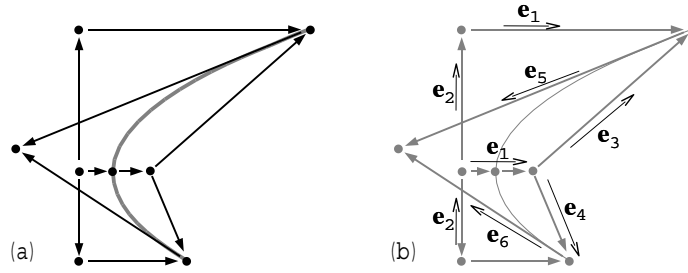


FIG. 6.12: Structure d'une parabole : (a) ensemble des points et des bipoints , (b) ensemble des vecteurs .

$$b = \|\mathbf{CB}_1\| = \|\mathbf{CB}_2\| ; \quad (6.56)$$

$$c = \|\mathbf{CF}\| = \|\mathbf{CF}'\| . \quad (6.57)$$

Ces scalaires vérifient les relations :

$$b^2 = a^2 - c^2 ; \quad (6.58)$$

$$e = \frac{c}{a} . \quad (6.59)$$

En raison de cette symétrie, un second foyer \mathbf{F}' et une nouvelle directrice \mathcal{D}' peuvent être identifiés. Une ellipse est alors le lieu des points \mathbf{M} dont la somme des distances aux deux foyers \mathbf{F} et \mathbf{F}' est constante soit :

$$\|\mathbf{MF}\| + \|\mathbf{MF}'\| = 2a. \quad (6.60)$$

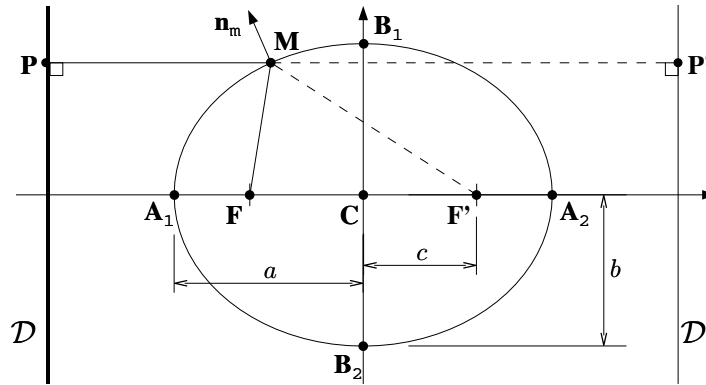


FIG. 6.13: Une ellipse

La normale \mathbf{n}_m à la courbe en un point \mathbf{M} est portée par la bissectrice de $(\widehat{MF, MF'})$ (Figure 6.13).

Modèle géométrique

Une ellipse est une courbe de Bézier rationnelle quadratique telle que $4\kappa > 1$ soit avec nos conventions, $h_1 < 1$. Partant de la courbe paramétrique, les caractéristiques d'une ellipse sont identifiables à partir d'un ensemble de relations

proposées par Lee [Lee 1987] :

$$\mathbf{S}_0\mathbf{C} = \nu(\mathbf{u} + \mathbf{v}) ; \quad (6.61)$$

$$a = \sqrt{\frac{\nu}{\lambda_1}} ; \quad (6.62)$$

$$b = \sqrt{\frac{\nu}{\lambda_2}} ; \quad (6.63)$$

$$e = 1 - \frac{\lambda_1^2}{\lambda_2^2}. \quad (6.64)$$

Structure

Comme pour la parabole, la structure d'une ellipse (Figure 6.14) regroupe les propriétés mathématiques des points extrémités et des sommets principaux ainsi que le polygone caractéristique de la courbe paramétrique. À cela est ajouté un ensemble de relations assurant la cohérence entre les différents éléments.

$$\|\mathbf{CF}\| = \|\mathbf{CF}'\| = e = 1 - \frac{\lambda_1^2}{\lambda_2^2} ; \quad (6.65)$$

$$\|\mathbf{CA}_1\| = \|\mathbf{CA}_2\| = a = \sqrt{\frac{\nu}{\lambda_1}} ; \quad (6.66)$$

$$\|\mathbf{M}_1\mathbf{F}\| + \|\mathbf{M}_1\mathbf{F}'\| = 2a ; \quad (6.67)$$

$$\|\mathbf{M}_2\mathbf{F}\| + \|\mathbf{M}_2\mathbf{F}'\| = 2a ; \quad (6.68)$$

$$(\widehat{\mathbf{e}_2, \mathbf{e}_4}) = (\widehat{\mathbf{e}_4, \mathbf{e}_3}) ; \quad (6.69)$$

$$(\widehat{\mathbf{e}_5, \mathbf{e}_7}) = (\widehat{\mathbf{e}_7, \mathbf{e}_6}) ; \quad (6.70)$$

$$(\widehat{\mathbf{e}_4, \mathbf{e}_8}) = (\widehat{\mathbf{e}_9, \mathbf{e}_7}) = \frac{\pi}{2} ; \quad (6.71)$$

$$h_1 = \sqrt{\frac{1}{4\kappa}}. \quad (6.72)$$

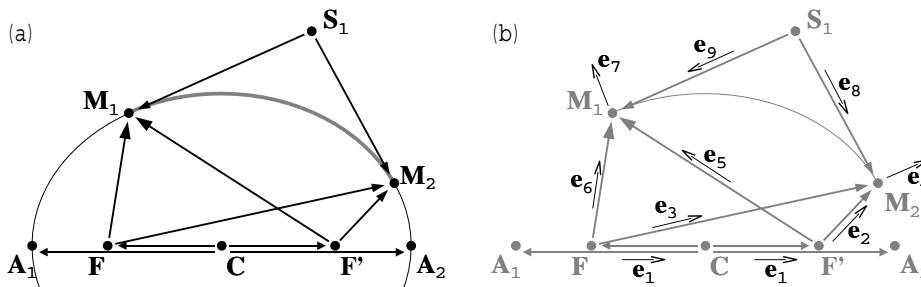


FIG. 6.14: Structure d'une ellipse : (a) ensemble des points et des bipoints, (b) ensemble des vecteurs

6.4.4 Arc d'hyperbole

Définition mathématique

Une hyperbole est une conique dont l'excentricité e est supérieure à 1. Comme l'ellipse, elle admet deux axes de symétrie : l'axe focal aussi appelé *grand axe* et la droite perpendiculaire à l'axe focal, passant par le milieu du segment reliant les deux sommets A_1 et A_2 (Figure 6.15). Soit C le centre de l'hyperbole. On définit alors les scalaires a , b et c tels que :

$$a = \|CA_1\| = \|CA_2\| ; \quad (6.73)$$

$$c = \|CF\| = \|CF'\| . \quad (6.74)$$

Ces scalaires vérifient les relations :

$$b^2 = c^2 - a^2 ; \quad (6.75)$$

$$e = \frac{c}{a} . \quad (6.76)$$

Comme l'ellipse, un second foyer F' et une nouvelle directrice D' peuvent être définis. Une hyperbole est alors le lieu des points M dont la différence des distances aux deux foyers F et F' est constante, soit :

$$\|MF\| - \|MF'\| = 2a. \quad (6.77)$$

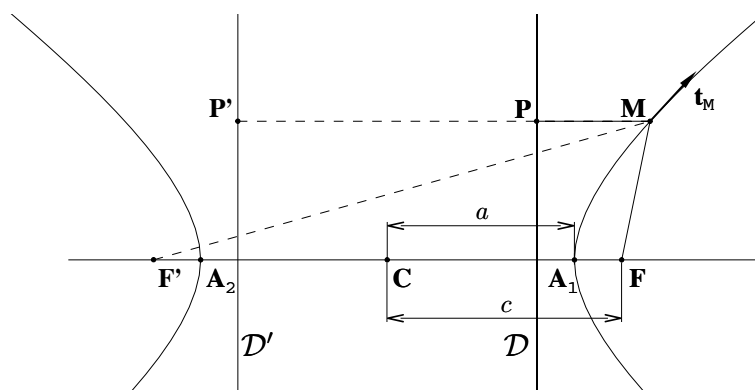


FIG. 6.15: Une hyperbole

La tangente t_m à la courbe en un point M est portée par la bissectrice de $(\widehat{MF, MF'})$ (Figure 6.15).

Modèle géométrique

Une hyperbole est une courbe de Bézier rationnelle quadratique telle que $4\kappa < 1$ soit, avec nos conventions, $h_1 < 1$. Comme pour l'ellipse, de la courbe paramétrique, les caractéristiques d'une hyperbole sont identifiables à partir des relations (6.61), (6.62) et (6.64).

Structure

Comme pour la parabole et l'ellipse, la structure d'une hyperbole regroupe les éléments caractéristiques d'un arc d'hyperbole et du réseau de la courbe paramétrique associée. À cela est ajouté un ensemble de relations assurant la cohérence entre les différents éléments :

$$\|\mathbf{CF}\| = \|\mathbf{CF}'\| = e ; \quad (6.78)$$

$$\|\mathbf{CA}_1\| = \|\mathbf{CA}_2\| = a = \sqrt{\frac{\nu}{\lambda_1}} ; \quad (6.79)$$

$$\| \|\mathbf{M}_1\mathbf{F}\| - \|\mathbf{M}_1\mathbf{F}'\| \| = 2a ; \quad (6.80)$$

$$\| \|\mathbf{M}_2\mathbf{F}\| - \|\mathbf{M}_2\mathbf{F}'\| \| = 2a ; \quad (6.81)$$

$$(\widehat{\mathbf{e}_2, \mathbf{e}_4}) = (\widehat{\mathbf{e}_4, \mathbf{e}_3}) ; \quad (6.82)$$

$$(\widehat{\mathbf{e}_5, \mathbf{e}_7}) = (\widehat{\mathbf{e}_7, \mathbf{e}_6}) ; \quad (6.83)$$

$$h_1 = \sqrt{\frac{1}{4\kappa}}. \quad (6.84)$$

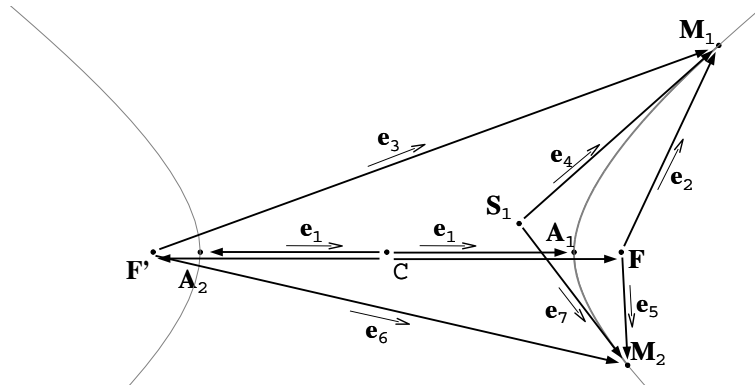


FIG. 6.16: Structure d'une hyperbole

6.4.5 Arc de cercle

Définition mathématique

Un cercle est une ellipse d'excentricité $e = 0$. Les deux foyers et le centre sont donc confondus. C'est alors le lieu des points \mathbf{M} équidistants au centre \mathbf{C} :

$$\|\mathbf{MC}\| = r \quad (6.85)$$

Modèle géométrique

Un arc de cercle peut être parfaitement représenté par une courbe de Bézier rationnelle de degré 2 (Figure 6.17) telle que :

$$\mathbf{u} = \mathbf{v} ; \quad (6.86)$$

$$h_0 = h_2 = 1 ; \tag{6.87}$$

$$h_1 = \cos \alpha, \alpha \in [0, \pi/2] . \tag{6.88}$$

Le centre \mathbf{O} du cercle est alors connu par :

$$\mathbf{S}_1\mathbf{O} = \frac{1}{2 \sin^2 \alpha} (\mathbf{v} - \mathbf{u}) . \tag{6.89}$$

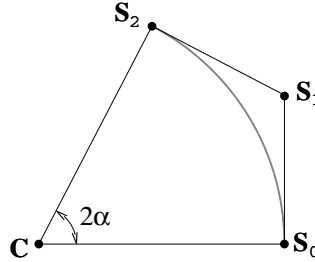


FIG. 6.17: Un arc de cercle défini par une courbe de Bézier rationnelle quadratique.

Structure

Pour définir les arcs de cercle quelque soit la valeur de α , ils sont modélisés par trois courbes de Bézier rationnelles. En effet, trois est le nombre minimum de courbes permettant de définir un cercle sans avoir de sommets rejetés à l'infini et/ou de coordonnées homogènes négatives (Figure 6.18). Afin de simplifier le traitement, les trois courbes sont identiques (Figure 6.19), ce qui évite des changements de topologie de l'entité.

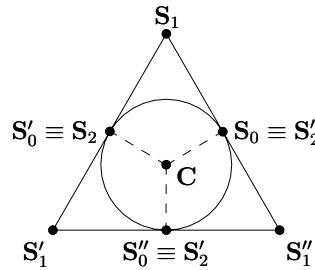


FIG. 6.18: Modélisation d'un cercle avec trois courbes de Bézier rationnelles.

6.5 Conclusion

Ce chapitre a présenté la structure et la toile de différents objets géométriques. Certains objets ont une structure complexe afin de permettre à l'utilisateur d'avoir une totale liberté sur leur spécification. Pour cela, le chapitre suivant

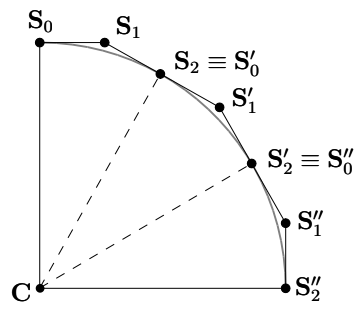


FIG. 6.19: Modélisation d'un arc de cercle avec trois courbes de Bézier rationnelles identiques

va exposer la structure des contraintes géométriques. En d'autres termes, nous allons voir comment les spécifications géométriques et les spécifications dimensionnelles peuvent être représentées avec le même formalisme.

Chapitre 7

L'expression des contraintes géométriques

Ce chapitre présente la structure des différentes contraintes géométriques. Il sera donc exposé dans ce chapitre, le moyen de décrire les contraintes géométriques avec le même formalisme que pour les éléments géométriques.

7.1 Introduction

Un objet géométrique est composé d'une toile et d'une structure. Nous avons vu que ce formalisme permet de le décrire de façon naturelle, c'est-à-dire que sa description ne s'appuie pas sur un quelconque repère cartésien mais sur les relations intrinsèques entre ses différents éléments.

Le chapitre précédent a décrit un ensemble d'éléments géométriques. L'assemblage de ces éléments permet de construire la toile mais surtout la structure des différents objets.

Dans une approche de construction géométrique sous contraintes, l'utilisateur définit son objet par un ensemble d'éléments et de contraintes géométriques qui doivent être vérifiés. Ces contraintes imposent la dimension et/ou la position relative de certains éléments. Dans ce modèle, c'est donc la dimension et/ou la position relative des éléments de la structure qui doivent vérifier les contraintes géométriques.

La structure contient des éléments définis par une mesure. Les contraintes géométriques vont donc imposer la mesure de certains éléments. Les éléments mesurables, autres que le **multipoint**, sont le **bipoint** et l'**angle**. Les contraintes géométriques élémentaires sont donc la distance entre deux points et l'angle entre deux vecteurs. Les autres contraintes géométriques sont des combinaisons de ces deux contraintes élémentaires. Ce chapitre va donc présenter la structure

des différentes contraintes géométriques. Avant tout, quelques définitions sont données afin de préciser les différents termes utiles pour cet exposé. Dans un second temps, la structure des contraintes sera présentée dans un espace 2D. Puis la généralisation au 3D des contraintes classiques sera proposée.

7.2 Définitions

Variable Certains éléments de la structure comme le **bipoint** ou l'**angle** ont une mesure. Cette mesure est enregistrée dans une **variable**.

Variable source Une **variable source** est une **variable** dont la valeur est imposée. Par exemple, lorsque l'utilisateur impose que la distance entre deux points a une valeur d , la **variable** du **bipoint** définie entre les deux **points** supports prend la valeur d et devient une **variable source**.

Variable puits Variable dont la valeur n'est pas imposée directement, mais déduite des valeurs des **variables sources**. En effet, pour que la structure reste cohérente après la modification de la mesure d'un de ses éléments contraints, les **variables puits** doivent s'ajuster en fonction de la valeur immuable des **variables sources**.

Variable libre Une **variable libre** est une **variable puits** que l'utilisateur encourage à être modifiée. Un problème sous-contraint admet une infinité de solutions, l'ensemble des **variables sources** ne suffit donc pas à déduire de façon rigoureuse la valeur des **variables puits**. Mais l'utilisateur ne souhaite pas voir toutes les **variables puits** modifier leur mesure. Il préfère que les modifications soient *naturelles*, c'est-à-dire conformes à ses prévisions. Or, dès que la taille du problème devient importante, le comportement naturel est difficile à spécifier. Avec les **variables libres**, l'utilisateur va pouvoir informer le modèle de ses préférences.

Lorsque les figures le nécessitent, une entité dont la **variable** est **source** est représentée en traits épais alors qu'une entité dont la **variable** est **puits** est en traits fins. Quant aux entités dont les **variables** sont **libres**, ils sont représentés en traits pointillés.

7.3 Contraintes angulaires 2D

7.3.1 Angle "droite – droite"

Une droite est définie par deux points ou par un point et une direction. Si la direction d'une droite définie par deux points est contrainte alors un **bipoint** doit être construit entre les deux **points** supports des points. Spécifier l'angle entre deux droites est alors équivalent à spécifier la direction de la première par

rapport à la seconde. Il suffit donc de préciser la mesure de l'angle composé par les deux vecteurs supports. Par exemple, contraindre l'angle entre les droites \mathcal{D}_1 et \mathcal{D}_4 de la figure 7.1 revient à imposer la mesure de l'angle $(\widehat{e_1, e_3})$. La variable de cet angle devient donc une variable source.

7.3.2 Parallélisme “droite – droite”

Cette contrainte peut-être établie de deux façons différentes. En effet, deux droites sont parallèles si :

- leurs vecteurs supports sont colinéaires : c'est une contrainte temporaire. Deux vecteurs sont colinéaires si l'angle qu'ils forment est égal à $0[\pi]$ (Figure 7.1). Cette contrainte est donc traitée numériquement. Par exemple, contraindre le parallélisme des droites \mathcal{D}_1 et \mathcal{D}_2 de la figure 7.1 revient à imposer la mesure de l'angle $(\widehat{e_1, e_2})$ égale à $0[\pi]$;
- elles sont portées par le même vecteur support : la contrainte est définitive. Par exemple, les droites \mathcal{D}_2 et \mathcal{D}_3 sont parallèles (Figure 7.1). Cette contrainte est donc traitée logiquement.

Le choix est fonction de la nature de la contrainte : soit elle est définitive, soit elle peut être supprimée. Par exemple, certaines contraintes comme la contrainte de coïncidence “point – droite”, utilisent la contrainte définitive de parallélisme pour lier des éléments lui étant propres.

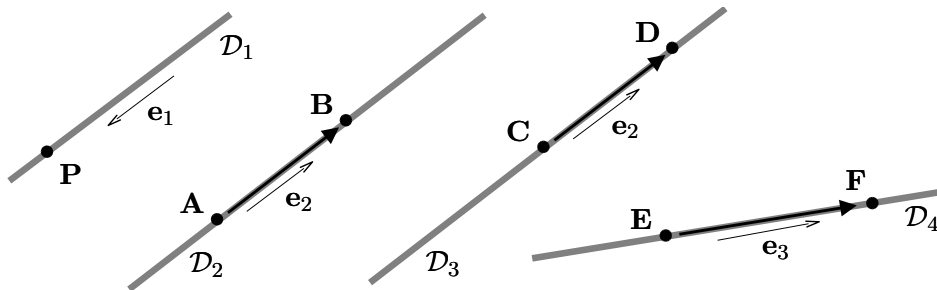


FIG. 7.1: Les différentes façons de définir l'angle entre deux droites

7.4 Contraintes de coïncidence 2D

7.4.1 Coïncidence “point – point”

Comme pour le parallélisme entre deux droites, deux types de coïncidence “point – point” sont définies : la coïncidence absolue et la coïncidence d'assemblage. Le choix de la coïncidence est basé sur un critère temporel. La coïncidence absolue sera préférée lorsqu'elle est définitive. En effet, deux points **A** et **B** sont absolument coïncidents si et seulement s'ils ne sont qu'un. En d'autres termes, ces deux

points sont portés par le même **spoint** \mathbf{p} . La figure 7.2-a illustre la coïncidence absolue des extrémités des deux **bipoints**. Lorsque la coïncidence n'est spécifiée que temporairement (comme pour une application d'assemblage) elle est traduite par un **copoint** liant deux **spoints** \mathbf{a} et \mathbf{b} . La figure 7.2-b illustre cette coïncidence d'assemblage pour la liaison entre deux objets. Ces coïncidences sont logiques. En effet, elles ne sont jamais évaluées comme peut l'être une distance.

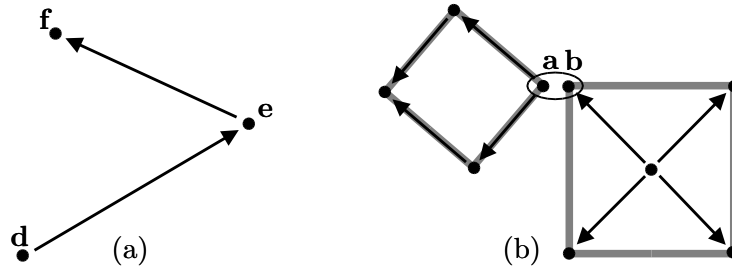


FIG. 7.2: Les différentes façons de définir la coïncidence entre deux points

7.4.2 Coïncidence “point – droite”

La structure d'une droite \mathcal{D} est constituée soit de deux **spoints** \mathbf{a} , \mathbf{b} , soit d'un **spoint** \mathbf{a} et un **vecteur** $\mathbf{e}_{\mathcal{D}}$, alors que la structure d'un point \mathbf{P} est un **spoint** \mathbf{p} . Pour que le point \mathbf{P} soit coïncident à la droite \mathcal{D} , il suffit de dire que le **bipoint** \mathbf{ap} est colinéaire au **vecteur** $\mathbf{e}_{\mathcal{D}}$ ou au **bipoint** \mathbf{ab} (Figure 7.3).

Pour cette contrainte, la colinéarité du **bipoint** \mathbf{ap} avec le **vecteur** $\mathbf{e}_{\mathcal{D}}$ ou le **vecteur** $\mathbf{e}_{\mathbf{ab}}$ du **bipoint** \mathbf{ab} est définie de façon absolue (le **bipoint** \mathbf{ap} est porté par le **vecteur** $\mathbf{e}_{\mathcal{D}}$ ou le **vecteur** $\mathbf{e}_{\mathbf{ab}}$). La raison est que la partie du squelette spécifique à cette contrainte ne sera jamais utilisée pour d'autres contraintes.

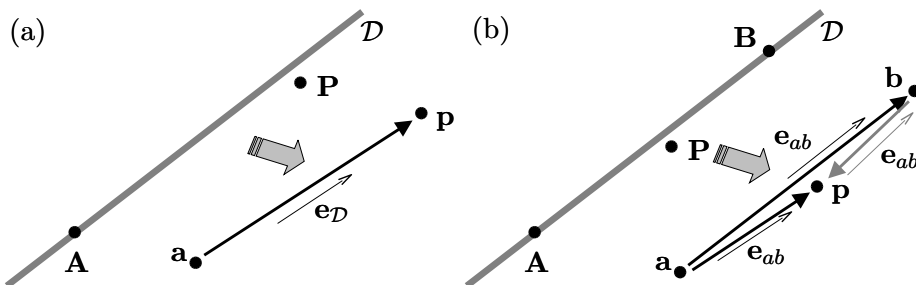


FIG. 7.3: Structure de la coïncidence “point – droite”

Dans le cas où la droite est portée par deux **spoints** \mathbf{a} et \mathbf{b} , la contrainte est doublée, c'est-à-dire qu'en plus du **bipoint** \mathbf{ab} , le **bipoint** \mathbf{bp} porté par le **vecteur** $\mathbf{e}_{\mathbf{ab}}$ est créé (Figure 7.3). La raison de ce doublement est de ne pas bloquer la résolution par un mauvais choix de structure. En effet, prenons l'exemple

de la figure 7.4-a. Le problème de cette figure est un carré de côté d_1 posé contre deux segments orthogonaux entre eux dont les longueurs ne sont pas spécifiées. Les sommets P_1 et P_2 de ce carré doivent rester coïncidents aux segments. De plus, la distance du sommet P_1 au point B , intersection des segments, doit être égale à d_2 .

Une structure (partielle) possible pour ce problème est représenté à la figure 7.4-b. Les données du problème permettent de calculer facilement la mesure des bipoints P_1P_2 et P_1P_4 . La résolution du problème avec cette structure ne peut pas se poursuivre. En effet, comme la mesure des bipoints BC et CP_2 n'est pas connue, il n'existe plus de structure partielle résolvable. Or, si l'on utilise la structure complète de la contrainte de coïncidence, le bipoint BP_2 existe. Les données sont alors suffisantes pour calculer la structure triangulaire $\{B, P_1, P_2\}$, ce qui permet de positionner le carré contre les deux segments sans connaître leurs longueurs.

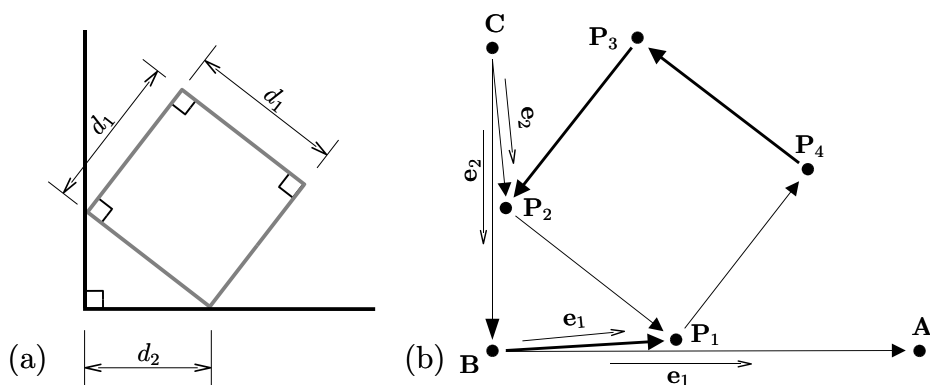


FIG. 7.4: Exemple de structure contenant une coïncidence “point-droite”.

7.4.3 Coïncidence “droite – droite”

La structure de la coïncidence d'une droite \mathcal{D} à une droite \mathcal{D}' peut être construite de différentes manières pour un résultat final identique, mais le choix de la structure a des conséquences sur la qualité du système d'équations.

Comme une droite a deux structures différentes, il y a trois combinaisons possibles. Pour simplifier, seule la plus complexe sera étudiée, c'est-à-dire le cas dans lequel les deux droites sont définies par deux points.

Soit la droite \mathcal{D} portée par les **spoints** a et b et la droite \mathcal{D}' portée par les **spoints** c et d . Les **bipoints** ab et cd sont portés respectivement par les **vecteurs** e et e' . La coïncidence des droites \mathcal{D} et \mathcal{D}' est équivalente :

- soit à la coïncidence des deux points de la droite \mathcal{D}' avec la droite \mathcal{D} . Les points connus de la droite \mathcal{D} sont les points portés par les **spoints** c et

d. Il suffit alors de construire pour chacun de ces points, la structure de la coïncidence “point – droite” comme le propose la figure 7.5-a ;

- soit à la coïncidence d'un point de la droite \mathcal{D}' avec la droite \mathcal{D} et au parallélisme des deux droites \mathcal{D} et \mathcal{D}' (Figure 7.5-b).

Pour les mêmes raisons que pour la coïncidence d'un point à une droite, la structure de la contrainte de coïncidence entre deux droites est sur-contrainte. En effet, la structure retenue (Figure 7.5-c) est composée de deux contraintes de coïncidence “point – droite” et d'une contrainte de parallélisme “droite – droite”.

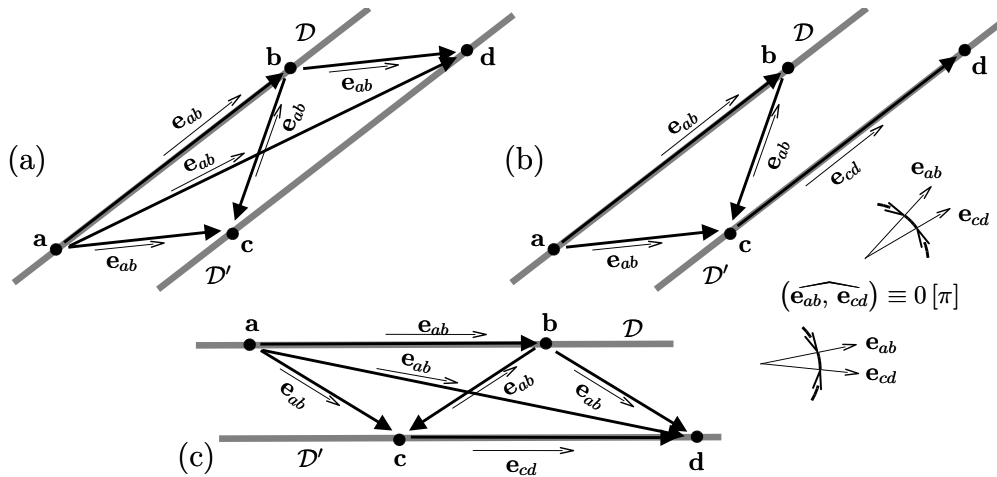


FIG. 7.5: différentes structures pour la coïncidence “droite – droite”

7.5 Contraintes dimensionnelles 2D

7.5.1 Distance “point – point”

La contrainte de distance “point – point” se traduit aisément dans ce formalisme. En effet, spécifier la distance entre deux points revient à spécifier le module d'un bipoint dont les extrémités sont les spoints supports des points considérés. La variable de ce bipoint devient une variable source.

7.5.2 Distance “point – droite”

Par définition, la distance d'un point \mathbf{P} à une droite \mathcal{D} est la distance $d(\mathbf{P}, \mathbf{X}_p)$, avec \mathbf{X}_p , projection orthogonale de \mathbf{P} sur la droite \mathcal{D} . Spécifier que la distance d'un point \mathbf{P} à une droite \mathcal{D} doit être de d revient donc à dire que :

- le point \mathbf{X}_p est coïncident à la droite \mathcal{D} ;
- la droite portée par les points \mathbf{X}_p et \mathbf{P} est orthogonale à la droite \mathcal{D} ;
- la distance entre les points \mathbf{X}_p et \mathbf{P} est celle recherchée.

La structure de cette contrainte est schématisée à la figure 7.6.

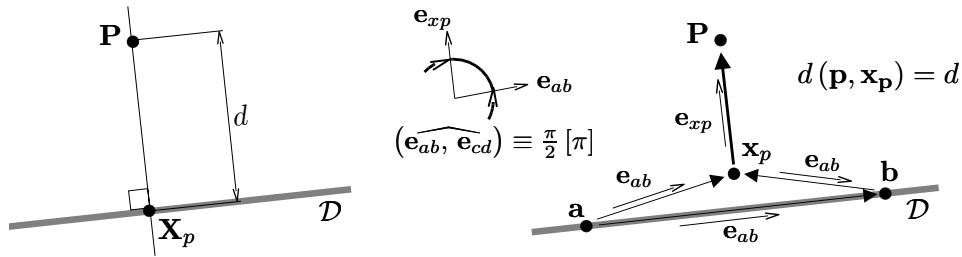


FIG. 7.6: Structure de la contrainte de distance “point – droite”

7.5.3 Distance “droite – droite”

La distance “droite – droite” pose les mêmes problèmes que la contrainte de coïncidence “droite – droite”. En effet, plusieurs décompositions en contraintes élémentaires existent.

Une première construction consiste à positionner un point C à la distance spécifiée de la droite \mathcal{D} portée par les points A et B , et par ce point C faire passer la droite \mathcal{D}' parallèlement à la première. Cette approche se traduit par l’association d’une contrainte de distance “point – droite” entre le point C et la droite \mathcal{D} et de la contrainte de parallélisme entre les droites \mathcal{D} et \mathcal{D}' (Figure 7.7-a).

L’association de deux contraintes de distance “point – droite” entre les points C et D et la droite \mathcal{D} permet aussi d’obtenir le résultat. En effet, par deux points distincts ne passe qu’une droite (Figure 7.7-b), les points C et D étant équidistants à la droite \mathcal{D} , les droites \mathcal{D} et \mathcal{D}' sont donc parallèles.

Finalement, la structure retenue pour cette spécification est aussi rendue redondante par l’association de deux contraintes de distance “point – droite” et de la contrainte de parallélisme “droite – droite”.

7.6 Contraintes de liberté

Les contraintes de liberté sont des spécifications permettant d’indiquer au modèle les éléments dont la valeur est encouragée à être modifiée. En théorie, toutes les contraintes dimensionnelles peuvent être dérivées en contraintes de liberté. En effet, tout comme une distance ou un angle sont impossibles, une distance ou un angle peuvent être désignés comme “libre”. La structure des contraintes de liberté est donc identique à celle des contraintes dimensionnelles. La différence apparaît au niveau du type de la **variable** modifiable de la contrainte. Elle devient donc une **variable libre**. La figure 7.8 représente leurs différentes structures.

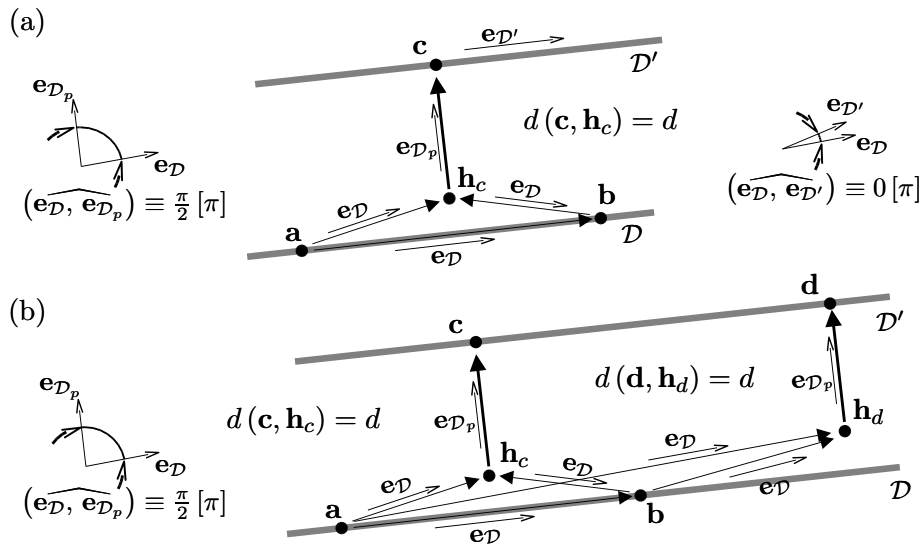


FIG. 7.7: Structure de la contrainte de distance “droite – droite”

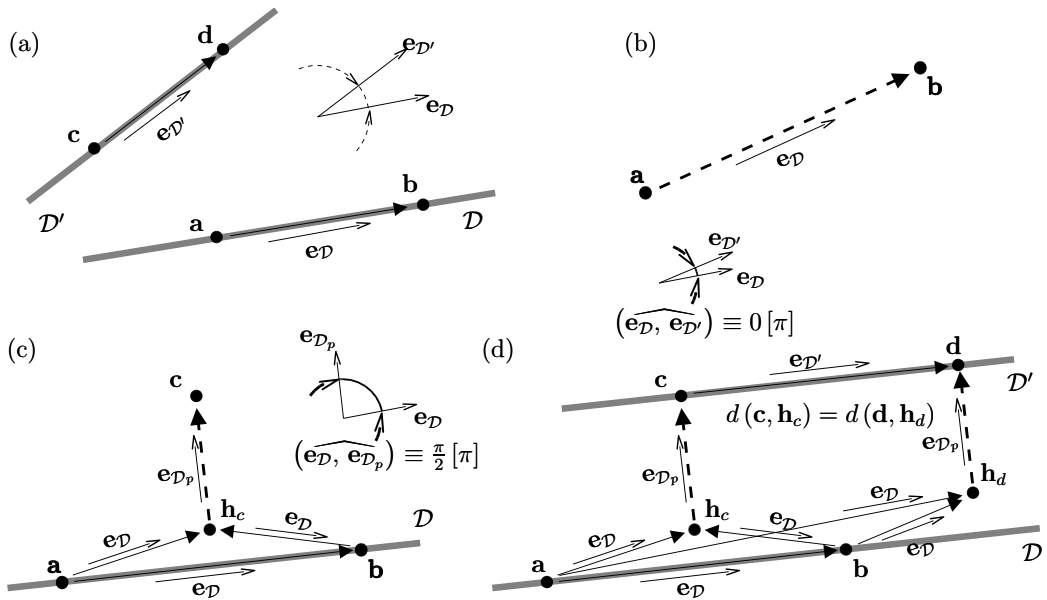


FIG. 7.8: Structure des contraintes de liberté : (a) angle “droite – droite”; (b) distance “point – point”; (c) distance “point – droite”; (d) distance “droite – droite”;

7.7 Les contraintes sur les courbes

Les contraintes classiquement utilisées en CAO sont des contraintes définies entre deux éléments de type point, droite ou plan. Or, l'utilisateur peut être amené à contraindre la position d'un élément par rapport à une courbe. Entre points et droites peuvent être définies des contraintes de coïncidence, de distance et des contraintes angulaires. Avec une courbe, les mêmes familles de contraintes peuvent être définies. Cette section propose donc un ensemble de contraintes liant une courbe d'une part à un point, une droite ou une courbe d'autre part. La structure de ces contraintes est alors présentée.

7.7.1 Contraintes angulaires

Angle "courbe – droite"

Soit une courbe \mathcal{C} , un point de la courbe $\mathbf{P}(u)$ et une droite \mathcal{D} . Dire que la courbe fait un angle α en $\mathbf{P}(u)$ avec la droite \mathcal{D} signifie que la tangente à la courbe \mathcal{C} en ce point fait un angle α avec la droite \mathcal{D} .

Afin de spécifier une telle contrainte, le **multipoint** définissant la tangente du point courant est créé. Il suffit alors de contraindre l'**angle** entre le **vecteur support** de ce **multipoint** et le **vecteur support** de la droite (Figure 7.9).

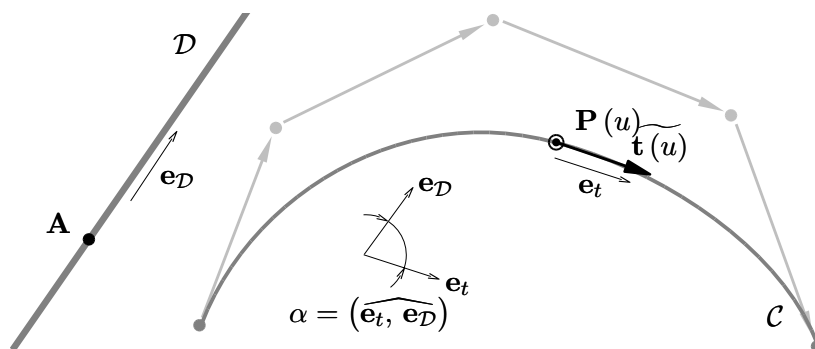


FIG. 7.9: Structure de la contrainte angulaire entre une droite \mathcal{D} et la tangente en un point $\mathbf{P}(u)$ d'une courbe \mathcal{C}

Angle "courbe – courbe"

Soit deux courbes \mathcal{C}_1 et \mathcal{C}_2 , deux points $\mathbf{P}(u)$ et $\mathbf{Q}(v)$ appartenant respectivement à chaque courbe. Dire que la courbe \mathcal{C}_1 fait un angle α en $\mathbf{P}(u)$ avec la courbe \mathcal{C}_2 en $\mathbf{Q}(v)$ signifie que la tangente à la courbe \mathcal{C}_1 en $\mathbf{P}(u)$ fait un angle α avec la courbe \mathcal{C}_2 en $\mathbf{Q}(v)$.

Pour spécifier une telle contrainte, l'**angle** entre les **vecteurs supports** des **multipoints** définissant les tangentes respectives des deux points est imposé (Figure 7.10).

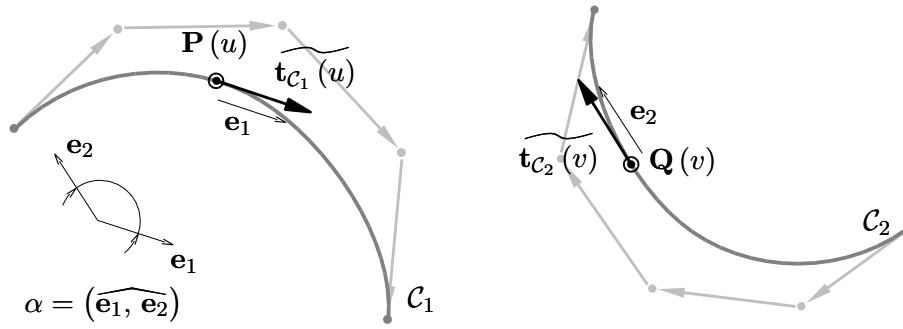


FIG. 7.10: Angle entre deux points appartenant à deux courbes

7.7.2 Contrainte de coïncidence

Coïncidence “courbe – point”

Soit un point \mathbf{A} extérieur à la courbe, soit un arc de courbe \mathcal{C} . La coïncidence du point \mathbf{A} avec la courbe consiste à spécifier qu'il existe un point $\mathbf{P}(u)$ de la courbe \mathcal{C} coïncident avec le point \mathbf{A} .

La structure d'une telle contrainte est composée d'un multipoint $\widetilde{\mathbf{S}_0\mathbf{P}(u)}$ permettant de localiser le point courant $\mathbf{P}(u)$. La contrainte de coïncidence est alors traduite par la coïncidence absolue entre $\mathbf{P}(u)$ et \mathbf{A} . L'extrémité du multipoint est le spoint support du point \mathbf{A} (Figure 7.11).

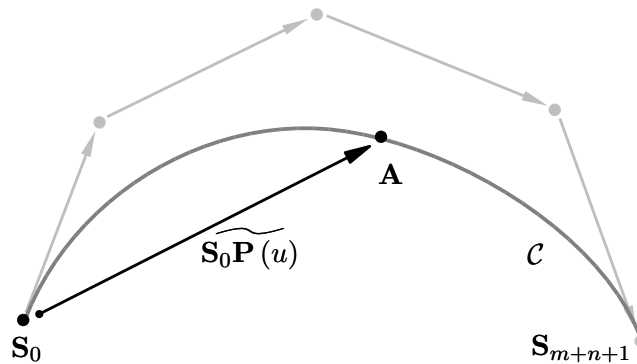


FIG. 7.11: Coïncidence entre un point \mathbf{A} et une courbe \mathcal{C}

Pour un arc de courbe quelconque deux points caractéristiques existent : les deux extrémités ($\mathbf{P}(u_0)$ et $\mathbf{P}(u_{m+n+1})$). Le cas particulier de la spécification de la coïncidence d'un point extérieur avec un de ces points caractéristiques se traduit simplement par la coïncidence entre le point extérieur et le spoint support du premier ou dernier pôle de la courbe.

Tangence “courbe – droite”

La contrainte de tangence de la courbe \mathcal{C} à la droite \mathcal{D} signifie que la ligne coupe la courbe en un point \mathbf{P} et en ce point la tangente à la courbe est parallèle à la

droite.

Pour définir cette contrainte, deux **multipoints** sont créés. Le premier, $\widetilde{S_0P}(u)$, définit le point courant $P(u)$ de la courbe et le second, $\widetilde{t_C}(u)$, la tangente courante. Ensuite, il suffit de définir une coïncidence entre le point courant et la droite (Cf. sous-section 7.4.1) et le parallélisme entre la droite et la tangente courante (Cf. sous-section 7.3.2). La structure de cette contrainte est représentée à la figure 7.12.

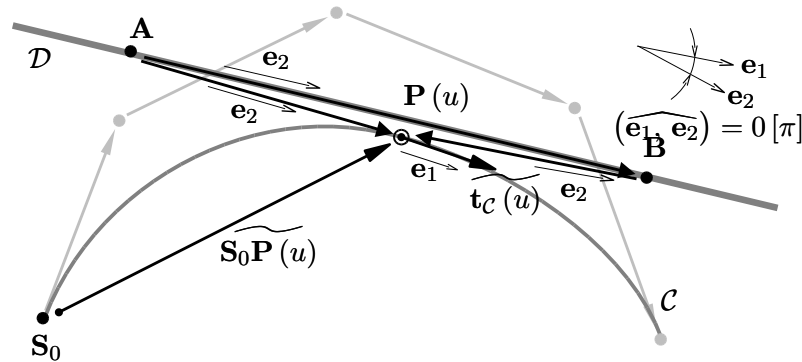


FIG. 7.12: Structure de la contrainte de tangence entre une courbe et une droite

Tangence “courbe – courbe”

La contrainte de tangence entre deux courbes C_1 et C_2 signifie que les courbes se coupent en un point A et en ce point leurs tangentes sont parallèles.

Pour définir cette contrainte, quatre **multipoints** sont créés. Les deux premiers, $\widetilde{S_0P}(u)$ et $\widetilde{S'_0Q}(v)$, définissent les points courants $P(u)$ et $Q(v)$ de chaque courbe et les seconds, $\widetilde{t_{C_1}}(u)$ et $\widetilde{t_{C_2}}(v)$, les tangentes courantes en ces points. Ensuite, il suffit de définir une coïncidence absolue entre les points courants et le point A ainsi que le parallélisme entre les tangentes courantes (Figure 7.13).

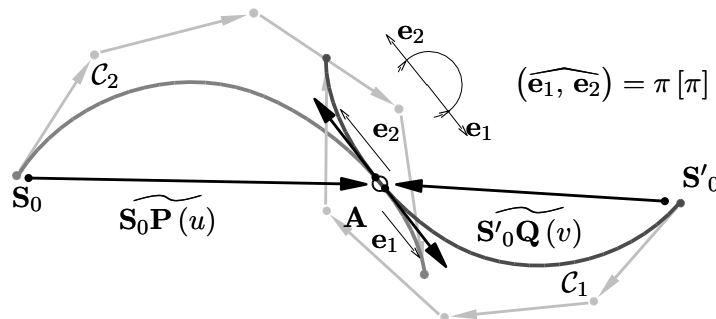


FIG. 7.13: Structure de la contrainte de tangence entre deux courbes

7.7.3 Contrainte de distance

Distance “courbe – point”

La contrainte de distance entre un point extérieur \mathbf{A} et la courbe \mathcal{C} consiste à spécifier que la distance minimale entre le point et la courbe est :

$$d = \min_{\forall u \in [u_0, u_{m+n+1}]} (d(\mathbf{A}, \mathbf{P}(u))). \quad (7.1)$$

La structure de cette contrainte est composée d'un multipoint $\widetilde{\mathbf{S}_0\mathbf{P}(u)}$ afin de définir le point courant $\mathbf{P}(u)$ de la courbe et d'un bipoint $\mathbf{P}(u)\mathbf{A}$ dont le module doit être égal à d (Voir figure 7.14).

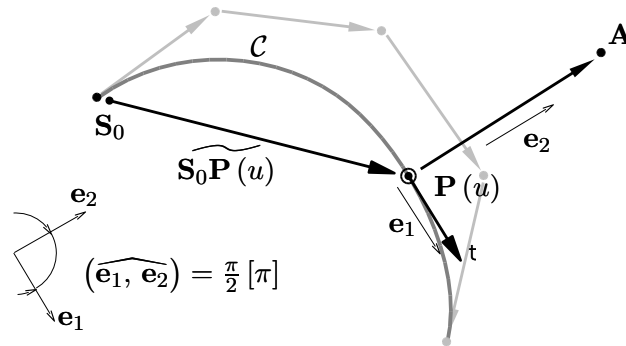


FIG. 7.14: Contrainte de distance entre un point et une courbe

Distance “courbe – droite”

La contrainte de distance entre la droite \mathcal{D} et la courbe \mathcal{C} consiste à spécifier que la distance minimale entre la droite et la courbe est d . Or, par les points appartenant respectivement à la droite et à la courbe, passe une droite perpendiculaire à la droite \mathcal{D} et portée par la normale à la courbe en ce point.

La structure de cette contrainte est donc composée :

- d'un multipoint $\widetilde{\mathbf{S}_0\mathbf{P}(u)}$ afin de définir le point courant $\mathbf{P}(u)$ de la courbe ;
- de la structure de la contrainte de distance d'un point à une droite entre le point $\mathbf{P}(u)$ et la droite \mathcal{D} ;
- d'un angle de $\pi/2 [\pi]$ entre la tangente $\widetilde{\mathbf{t}(u)}$ à la courbe au point courant et le vecteur support du bipoint liant le point courant à sa projection orthogonale sur la droite \mathcal{D} .

Distance “courbe – courbe”

La contrainte de distance entre deux courbes \mathcal{C}_1 et \mathcal{C}_2 spécifie que la distance minimale entre les courbes est d . Or, la droite, passant par les points courants

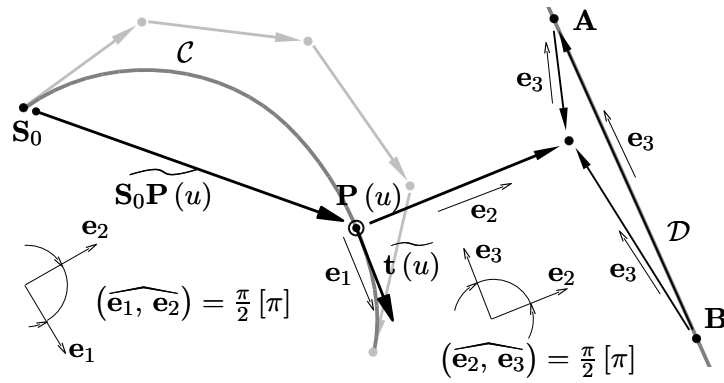


FIG. 7.15: Structure de la contrainte de distance entre une courbe et une droite

des courbes vérifiant la distance minimale, est portée par les normales des courbes en ces points.

La structure de cette contrainte est donc composée :

- d'un multipoint $\widetilde{S_0P}(u)$ afin de définir le point courant $P(u)$ de la courbe C_1 ;
- d'un multipoint $\widetilde{S'_0Q}(v)$ afin de définir le point courant $Q(v)$ de la courbe C_2 ;
- d'un multipoint $\widetilde{t_1}(u)$ afin de définir la tangente au point courant $P(u)$ de la courbe C_1 ;
- d'un multipoint $\widetilde{t_2}(v)$ afin de définir la tangente au point courant $Q(v)$ de la courbe C_2 ;
- du bipoint $P(u)Q(v)$ de module d ;
- d'un angle de $\pi/2 [\pi]$ entre la tangente $\widetilde{t_1}(u)$ et le bipoint $P(u)Q(v)$;
- d'un angle de $\pi/2 [\pi]$ entre la tangente $\widetilde{t_2}(v)$ et le bipoint $P(u)Q(v)$.

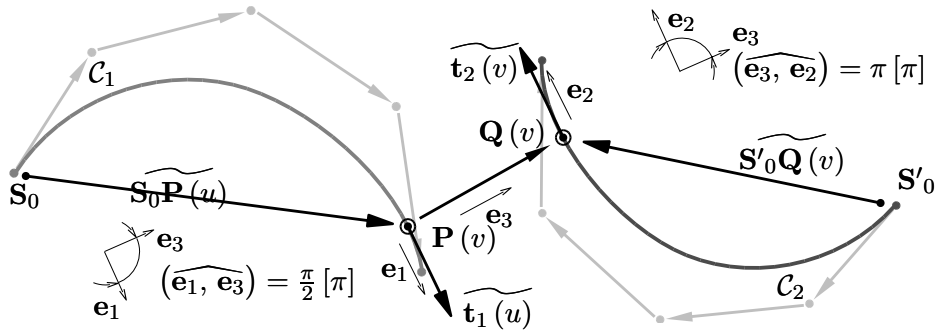


FIG. 7.16: Structure de la contrainte de tangence entre deux courbes

7.7.4 Contrainte de continuité “courbe – courbe”

Continuité “ G^1 ”

La continuité “ G^1 ” entre deux courbes adjacentes se traduit par un ensemble de contraintes géométriques liant les polygones caractéristiques des deux courbes. Soit deux courbes \mathcal{C} et \mathcal{C}' dont les polygones caractéristiques sont respectivement $\{\mathbf{S}_0, \dots, \mathbf{S}_m\}$ et $\{\mathbf{S}'_0, \dots, \mathbf{S}'_n\}$. La continuité “ G^1 ” en $\mathbf{P}(0)$ pour \mathcal{C} et en $\mathbf{P}'(0)$ pour \mathcal{C}' est obtenue si et seulement si :

- les deux courbes ont une continuité “ C^0 ” entre $\mathbf{P}(0)$ et $\mathbf{P}'(0)$, c’est-à-dire qu’il y a une coïncidence “point – point” entre les spoints \mathbf{S}_0 et \mathbf{S}'_0 des polygones caractéristiques ;
- les deux courbes sont continues en tangence entre $\mathbf{P}(0)$ et $\mathbf{P}'(0)$, c’est-à-dire que les vecteurs \mathbf{e}_1 et \mathbf{e}'_1 support des bipoints \mathbf{a}_1 et \mathbf{a}'_1 des polygones caractéristiques sont colinéaires.

Continuité “ G^2 ”

La continuité “ G^2 ” entre deux courbes adjacentes se traduit par une continuité “ G^1 ” complétée de relations supplémentaires. La continuité “ G^2 ” entre $\mathbf{P}(0)$ et $\mathbf{P}'(0)$ des deux courbes \mathcal{C} et \mathcal{C}' est donc obtenue si et seulement si :

- les deux courbes ont une continuité “ G^1 ” en $\mathbf{P}(0)$ et $\mathbf{P}'(0)$;
- il y a une continuité de l’orientation du plan osculateur au franchissement des courbes, c’est-à-dire que :

$$\sin(\widehat{\mathbf{e}_1, \mathbf{e}_2}) \sin(\widehat{\mathbf{e}'_1, \mathbf{e}'_2}) \geq 0,$$

avec \mathbf{e}_2 et \mathbf{e}'_2 les vecteurs supports respectivement des bipoints \mathbf{a}_2 et \mathbf{a}'_2 des polygones caractéristiques ;

- il y a égalité des rayons de courbure aux extrémités $\mathbf{P}(0)$ et $\mathbf{P}'(0)$, (Cf. éq. (6.40) et éq. (6.41)).

7.7.5 Conclusion

Les contraintes proposées sont des contraintes qui portent sur un point quelconque de chaque courbe. Elles peuvent être énoncées de la manière suivante : *il existe un point $\mathbf{P}(u)$ de la courbe \mathcal{C} tel que (la courbe est tangente en ce point à ...) ou (la distance de ce point à ... doit être égale à d) ou (l’angle entre la tangente de la courbe en ce point avec ... doit être égale à α).* Or, les points d’application de ces contraintes peuvent être précisés par leur coordonnée paramétrique u^* ou localisées par un intervalle de coordonnées admissibles. Ces contraintes dérivées s’énoncent alors : *le point $\mathbf{P}(u^*)$ de la courbe \mathcal{C} (tel que $u^* \in [a, b]$) est (coïncident) ou (tangente) ou ... à ...* Elles peuvent aussi être précisées par un ensemble de

contraintes. La contrainte globale devient alors : *le point \mathbf{P} de la courbe \mathcal{C} (tangent à ...) et/ou (dont la distance à ...) et/ou (dont l'angle avec ...) est (coïncident) ou (tangent) ou ... à* Ainsi, l'utilisateur peut spécifier avec une certaine liberté la forme des éléments curvilignes.

7.8 Les Contraintes 3D

Cette section propose la structure des contraintes conventionnelles entre un plan et un élément de type point, droite ou plan. Mais avant tout, rappelons la structure d'un plan (Cf. sous-section 6.2.3).

Un plan \mathcal{P} est défini par une surface de Bézier de degré un dont les quatre pôles sont portés par des **spoints** coplanaires. Ces quatre **spoints** sont donc coïncidents à un plan affine \mathcal{A} porté par un plan vectoriel \mathcal{V} de normale \mathbf{n} .

7.8.1 Contraintes angulaires 3D

Angle “plan – plan”

Contraindre l'angle α entre deux plans \mathcal{P}_1 et \mathcal{P}_2 revient à contraindre l'angle β entre les normales des plans vectoriels \mathcal{A}_1 et \mathcal{A}_2 (Figure 7.17). En fonction de l'orientation de ces plans, l'angle spécifié β vaut α (la valeur souhaitée), $(\pi - \alpha)$, $(\alpha - \pi)$ ou $-\alpha$ selon l'orientation initiale des plans.

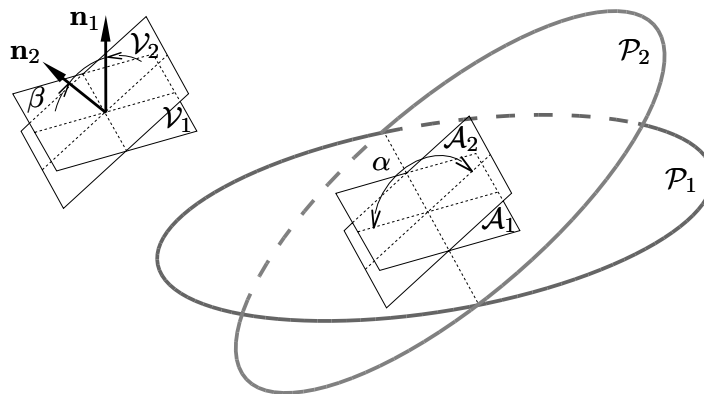


FIG. 7.17: Contrainte angulaire entre deux plans

Parallélisme “plan – plan”

La contrainte de parallélisme entre deux plans n'est pas simplement une contrainte angulaire dont la valeur spécifiée β est égale à $0 [\pi]$ (Figure 7.18).

En effet, le parallélisme de deux plans sous-entend la coïncidence de leurs plans vectoriels respectifs. Les différents éléments vectoriels (vecteurs et

angles) sont coïncident avec les deux plans vectoriels. L'important pour cette contrainte est que les plans connaissent leurs éléments propres et leurs éléments coïncidents. En effet, en cas de suppression de cette contrainte, il faut alors redistribuer les éléments selon leurs plans respectifs.

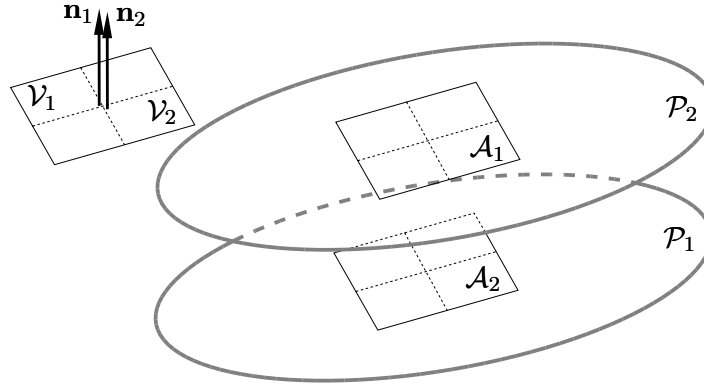


FIG. 7.18: Parallélisme entre deux plans

Angle “plan – droite”

L'angle entre un plan \mathcal{P} et une droite \mathcal{D} pourrait être facilement contraint. En effet, il suffirait de contraindre l'angle entre la droite et le vecteur normal au plan. Or, il n'est pas possible de définir un angle entre un vecteur normal et un vecteur unitaire (Cf. sous-section 6.2.3). Il reste donc à définir cet angle dans le plan orthogonal à \mathcal{P} contenant la droite \mathcal{D} . Pour cela, il faut construire les entités de la structure nécessaire à une telle contrainte (Figure 7.19), soit :

- un **spoint** $\mathbf{A} \in \mathcal{A}$, point d'intersection de la droite \mathcal{D} et du plan \mathcal{P} ;
- le **plan affine** \mathcal{A}^* porté par le **plan vectoriel** \mathcal{V}^* contenant la droite \mathcal{D} et orthogonal à \mathcal{A} . Les **normales** \mathbf{n} et \mathbf{n}^* des plans respectifs \mathcal{V} et \mathcal{V}^* sont par constructions orthogonaux entre eux et doivent le rester ;
- la droite $\mathcal{D}_{\mathcal{P}}$ passant par \mathbf{A} portée par le **vecteur** $\mathbf{e}_{\mathcal{D}_{\mathcal{P}}}$. Ce **vecteur** appartient aux plans vectoriels \mathcal{V} et \mathcal{V}^* ;
- l'**angle** $(\widehat{\mathbf{e}_{\mathcal{D}}, \mathbf{e}_{\mathcal{D}_{\mathcal{P}}}})$ dont la valeur doit être égale à α est défini dans le **plan vectoriel** \mathcal{V}^* .

Perpendicularité “plan – droite”

La contrainte de perpendicularité n'est autre qu'une contrainte angulaire dont la valeur de l'angle est égal à $\pi/2$. Si la droite n'est initialement pas perpendiculaire au plan \mathcal{P} , la définition des plans \mathcal{V}^* et \mathcal{A}^* est directe. Dans le cas contraire, ces plans sont choisis en fonction des autres données du problème.

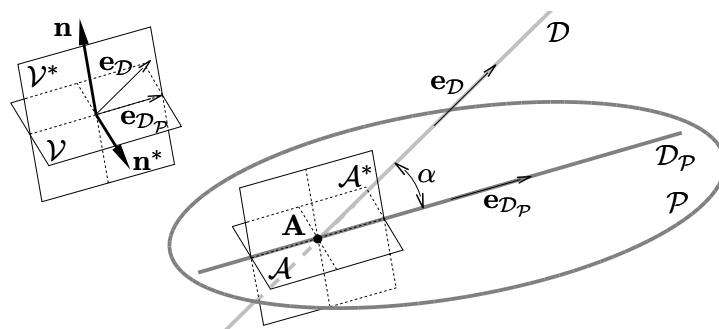
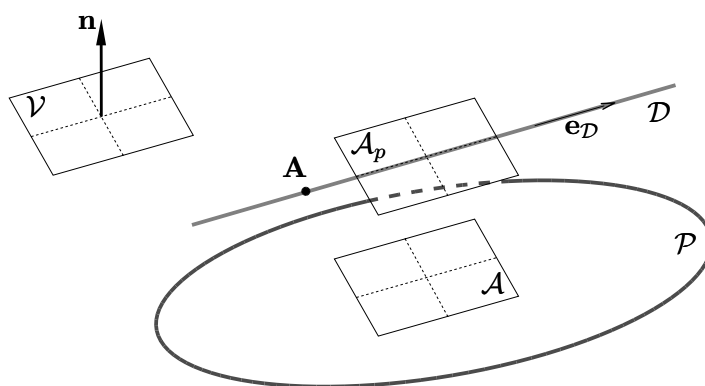


FIG. 7.19: La structure de la contrainte angulaire entre un plan et une droite

Parallélisme “plan – droite”

La contrainte de parallélisme “plan – droite” ne peut pas être définie de la même façon que la contrainte angulaire entre un plan et une droite. En effet, soit la droite \mathcal{D} coïncide avec \mathcal{P} , soit il n’y a pas d’intersection. Par contre, cette droite appartient à un plan parallèle au plan \mathcal{P} . La structure d’une telle contrainte (Figure 7.20) est donc définie par les éléments suivants :

- un plan affine \mathcal{A}_p porté par le plan vectoriel \mathcal{V} . Le parallélisme entre les plans \mathcal{A} et \mathcal{A}_p est donc absolu ;
- une contrainte de coïncidence entre la droite \mathcal{D} et le plan \mathcal{A}_p . Ce plan contient donc les **spoints** de \mathcal{D} comme le **spoint** \mathbf{A} . Le vecteur \mathbf{e}_D est donc inclus dans le plan vectoriel \mathcal{V} .

FIG. 7.20: Parallélisme entre un plan \mathcal{P} et une droite \mathcal{D}

7.8.2 Contraintes de coïncidence 3D

Coïncidence “plan – point”

La contrainte de coïncidence d’un point avec un plan est une contrainte logique. En effet, un plan \mathcal{P} est porté par un plan affine \mathcal{A} . Le plan affine connaît l’ensemble des **spoints** support des points coplanaires au plan \mathcal{P} . Pour spécifier la

coïncidence entre un point \mathbf{P} et le plan \mathcal{P} , il suffit de spécifier la coïncidence du **spoint** \mathbf{p} support du point \mathbf{P} avec le **plan affine** \mathcal{A} .

Coïncidence “plan – droite”

La structure d'une droite \mathcal{D} est constituée de **spoints**, de **bipoints** et d'un **vecteur** $\mathbf{e}_{\mathcal{D}}$. Pour spécifier la coïncidence de la droite avec le plan, il suffit de spécifier la coïncidence des éléments affines (**spoints** et **bipoints**) de \mathcal{D} avec le **plan affine** \mathcal{A} et la coïncidence des éléments vectoriels de \mathcal{D} avec le **plan vectoriel** \mathcal{V} .

Coïncidence “plan – plan”

La contrainte de coïncidence entre deux plans est, elle aussi, une contrainte logique. En effet, il suffit de dire que tous les éléments vectoriels et affines de l'un appartiennent respectivement aux plans **vectoriel** et **affine** de l'autre et inversement. Comme pour la contrainte de parallélisme entre plans, l'important pour cette contrainte est que les différents éléments connaissent leurs origines en cas de suppression de la contrainte.

7.8.3 Contraintes dimensionnelles 3D

Distance “plan – point”

Par définition, la distance d'un point \mathbf{X} à un plan \mathcal{P} est la distance $d(\mathbf{X}, \mathbf{X}_p)$, avec \mathbf{X}_p , projection orthogonale de \mathbf{X} sur le plan. Spécifier que la distance d'un point \mathbf{X} à un plan \mathcal{P} consiste donc à imposer la distance entre le point \mathbf{X} et sa projection orthogonale \mathbf{X}_p . Pour cela, il suffit de définir la droite \mathcal{D} perpendiculaire à \mathcal{P} passant par \mathbf{P} et le point \mathbf{X}_p , intersection de cette droite avec le plan. Ensuite, il suffit de spécifier la contrainte de perpendicularité entre le plan et la droite (Cf. sous-section 7.8.1), de contraindre la coïncidence du point \mathbf{X}_p au plan (Cf. sous-section 7.8.2) et la distance entre les points \mathbf{X} et \mathbf{X}_p (Cf. sous-section 7.5.1). Cet ensemble d'éléments définit la structure de cette contrainte schématisée à la figure 7.21.

Distance “plan – plan”

La contrainte de distance “plan – plan” sous-entend une contrainte de parallélisme “plan – plan” et une distance “point – plan”. Ainsi, la structure de cette contrainte représentée à la figure 7.22 est l'association de la structure des deux contraintes citées précédemment.

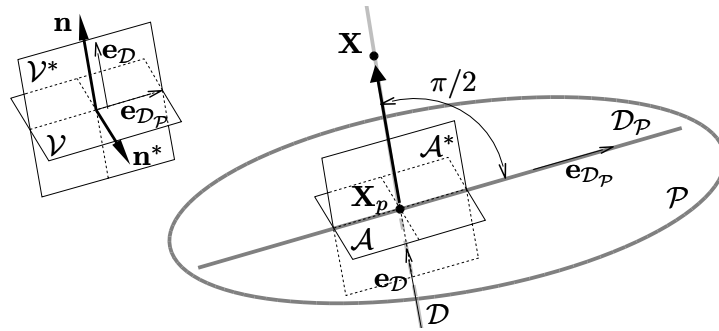


FIG. 7.21: Structure de la contrainte de distance d'un point à un plan

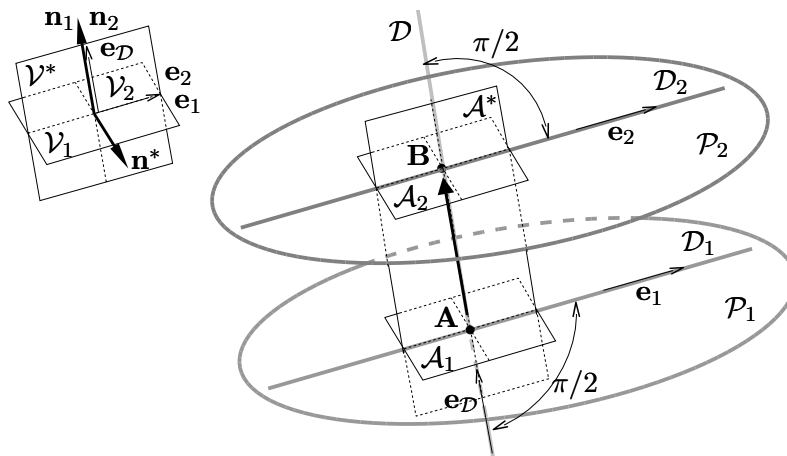


FIG. 7.22: Structure de la contrainte de distance entre plans

Distance “plan – droite”

La structure de la contrainte de distance d'un plan à une droite est très similaire à la structure de la contrainte de distance “plan – plan” (Figure 7.23).

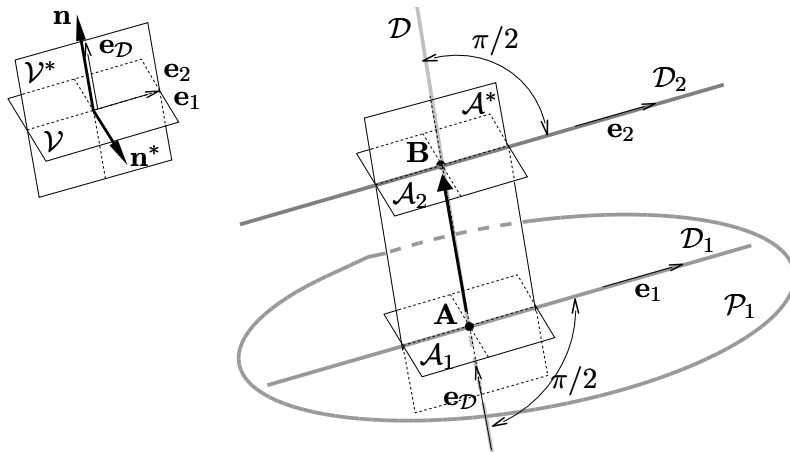


FIG. 7.23: La structure de la contrainte de distance d'une droite à un plan

7.8.4 Conclusion

Les structures des contraintes 3D ne sont pour l'instant que des propositions. Seule, une structure par contrainte a été présentée mais comme pour le 2D, plusieurs structures sont possibles pour la même contrainte et pour les mêmes raisons, il sera avantageux de la rendre sur-contrainte. Certaines structures peuvent sembler être complexes, mais elles sont présentées seules, hors contexte. Placées dans la structure d'une pièce, elles apparaîtront plus intégrées.

7.9 Synthèse

Après que la structure et la toile des éléments géométriques aient été présentées au chapitre 6, celui-ci a décrit la structure des contraintes géométriques. Ceci a permis de montrer que ces deux familles de spécifications sont, dans le modèle géométrique, représentées avec le même formalisme. Cela n'a d'intérêt que pour leur mise en équations qui sera présentée dans la deuxième section du prochain chapitre. Il doit, bien sûr, commencer par la présentation du modèle mathématique.

Chapitre 8

Le modèle algébrique

Le modèle algébrique est le dernier niveau du modèle d'information. Ici, est reformulé l'ensemble des spécifications dans un langage commun qui autorise l'analyse et la résolution du problème en incluant des contraintes d'ingénierie. Il sera donc rapidement présenté. Dans une seconde partie, le module de transcription du modèle géométrique au modèle algébrique sera présenté.

8.1 Introduction

Nous avons vu le modèle d'information regroupant les spécifications de l'utilisateur. Ce niveau public, constamment enrichi de spécifications, contient la description d'une partie du produit. En raison des possibilités offertes par une approche variationnelle, la cohérence de ces spécifications n'est pas garantie. Un module capable de gérer et d'analyser cet ensemble est nécessaire. À ce niveau, la description du produit, exprimée par l'utilisateur, n'est pas adéquate pour un tel module. Les informations doivent alors être traduites dans un langage plus adapté et enregistrées dans un nouveau modèle. Ce dernier est appelé *modèle algébrique* du fait de la nature de ce module. En effet, un langage général permettant de traduire un problème est celui des systèmes d'équations algébriques. Quel que soit le problème, son expression sous la forme d'un ensemble de relations algébriques est la plus adaptée à une résolution. Cependant, un ensemble d'équations algébriques ne met pas nécessairement en évidence la structure de ce système. Il est essentiel de faire appel à des outils complémentaires et donc à une représentation complémentaire pour réaliser des analyses plus fines.

Les chapitres précédents ont présenté le modèle géométrique ainsi que le dictionnaire entre ce modèle et le modèle de spécifications. Or, ce modèle géométrique a été mis en place pour simplifier le passage des spécifications

géométriques au modèle algébrique. La deuxième section de ce chapitre présente ce passage.

Le modèle de spécifications ne contient pas seulement des contraintes géométriques. Les contraintes d'ingénierie doivent elles aussi être réécrites sous la forme du *modèle algébrique*. Nous verrons en fin de chapitre que cette transformation est directe.

8.2 Le modèle algébrique

Le modèle algébrique est la traduction de la description par l'utilisateur d'un objet dans un langage plus adapté au module d'ingénierie. Cette description est composée d'un ensemble de spécifications. Ces spécifications peuvent être réparties en deux familles : la première contient des entités et la seconde est formée par les relations entre ces entités. Les entités sont des spécifications dont la particularité est d'être associée à un ou plusieurs paramètres. Les relations entre les entités deviennent alors des relations entre paramètres.

Un autre classement peut aussi être proposé. En effet, une partie des spécifications sont géométriques alors que les autres ne le sont pas. Les spécifications géométriques sont pré-traitées lors du passage du modèle de spécifications au modèle géométrique. Le traitement final est réalisé lors du passage au modèle algébrique. À ce niveau, les relations géométriques sont regroupées pour former des relations plus adaptées à la résolution. Les autres spécifications (spécifications d'ingénierie) n'ont pas besoin de ce pré-traitement.

Finalement, un ensemble de relations est formulé à partir de la description du produit. La résolution et les analyses ne portent alors plus sur le problème initial mais sur cette reformulation. Ainsi, pour permettre une analyse et des diagnostics pertinents, une relation doit connaître sa provenance. Pour permettre une partie de l'analyse, une relation doit être écrite sous une forme logique qui traduit le lien entre elle et les différents paramètres impliqués. Pour cela, elle est liée à un nœud "relation". Pour finaliser l'analyse et permettre la résolution, une relation doit pouvoir être évaluée. Elle peut donc faire appel à son "estimateur". Le diagramme de classes de la figure 8.1 représente une relation et ses dépendances.

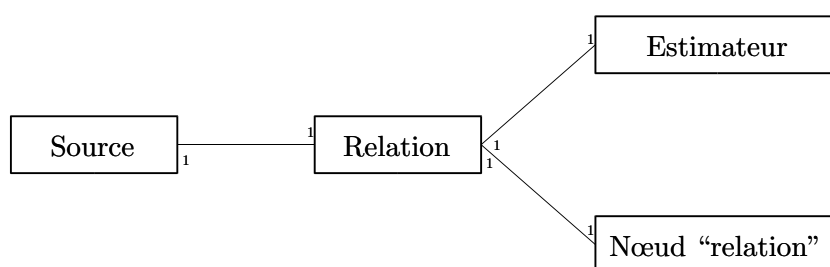


FIG. 8.1: Diagramme de classes d'une relation et de ses dépendances

8.2.1 L'estimateur

Une relation est généralement une expression algébrique liant un ensemble de paramètres. Or, dans certains cas, la formulation de certaines relations directement sous cette forme n'est pas réalisée. Les raisons sont multiples. Par exemple, bien que l'expression algébrique de la longueur d'une courbe, de l'aire d'une surface, ou du volume d'un solide puisse facilement être définie sous la forme d'intégrales, leurs estimations sont réalisées par des approches numériques. Dans d'autres cas, les relations ne peuvent simplement pas être explicitées. Par exemple, lorsqu'une relation est liée à une équation différentielle. Dans les cas les plus rares, une relation ne peut être évaluée que par une "boîte noire" extérieure à l'application. Or, nous verrons dans la partie suivante que l'important pour la résolution ou l'analyse du problème est de pouvoir estimer les relations et leurs premières dérivées à une précision près.

8.2.2 Le nœud "relation"

Une partie de l'analyse est une analyse structurelle. Pour cela, chaque relation est exprimée sous la forme d'un sous-graphe biparti (Cf. annexe C.6). Les nœuds "paramètres", liés aux paramètres du problème, composent le premier ensemble de nœuds. Le second contient des sommets référençant les relations. Le cocycle d'un nœud "relation" (Cf. annexe C.2) le relie à l'ensemble des nœuds de ses paramètres.

8.3 Du modèle géométrique au modèle algébrique

Les chapitres précédents ont introduit le modèle géométrique dans son ensemble et ont détaillé la forme des différents éléments et contraintes géométriques. Ces éléments permettent de construire la structure et la peau d'un objet. Ceux-ci regroupent un ensemble suffisant d'informations pour avoir une description complète de sa géométrie. De plus, nous avons vu qu'afin de gérer et d'analyser la cohérence des spécifications, celles-ci doivent être reformulées dans le module algébrique. Les spécifications géométriques ont été préalablement retranscrites dans le module géométrique. Ces données vont donc être à la base du passage de ces spécifications au modèle algébrique.

La problématique de cette section est d'identifier un ensemble de relations algébriques permettant la description complète des informations géométriques. Cet ensemble doit être nécessaire pour que toutes les spécifications géométriques se retrouvent dans le modèle algébrique et suffisante pour que l'état de contrainte du modèle algébrique soit identique à celui du modèle d'information.

Pour cela, l'idée est de rechercher un ensemble nécessaire et suffisant de relations permettant de vérifier l'existence de l'objet géométrique. Ainsi, dans cette section, la structure de l'objet est dans un état stable, c'est-à-dire que la mesure de chaque élément est connue. Elle est donc soit à l'état initial – les mesures des éléments ont été définies à partir du sketch initial –, soit à l'état final – les constituants vérifient les spécifications de l'utilisateur. Comme la structure supporte la toile, c'est donc elle qui définit la forme globale et les dimensions de l'objet. L'existence de cet objet dans l'espace de travail est alors dépendante de l'existence de la structure, elle-même dépendante de la mesure de ses différents constituants. En effet, il ne suffit pas de dire qu'une structure triangulaire est constituée de trois côtés et de trois angles pour qu'elle existe. Il faut aussi que la mesure de ses éléments vérifie un ensemble de relations. Par exemple, pour le triangle de la figure 8.2, ces relations sont, entre autres :

$$\alpha + \beta + \gamma = \pi [2\pi] ; \quad (8.1)$$

$$\frac{a}{\sin \alpha} = \frac{b}{\sin \beta} = \frac{c}{\sin \gamma} ; \quad (8.2)$$

$$a^2 = b^2 + c^2 - 2bc \cos \alpha ; \quad (8.3)$$

$$b^2 = c^2 + a^2 - 2ca \cos \beta ; \quad (8.4)$$

$$c^2 = a^2 + b^2 - 2ab \cos \gamma, \quad (8.5)$$

et permettent de contrôler la compatibilité des mesures des différents éléments, à ne pas confondre avec la cohérence des spécifications. Effectivement, la problématique de la première est de savoir, connaissant la mesure de chaque élément de la structure, si celle-ci existe dans l'espace de travail. Alors que la seconde cherche à savoir si, à un problème décrit par un ensemble de spécifications, il existe une ou plusieurs solutions.

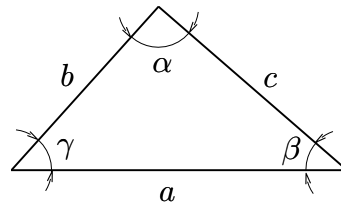


FIG. 8.2: Les éléments d'un triangle

Le problème revient donc à chercher un ensemble de relations permettant de s'assurer de la compatibilité des dimensions des éléments de la structure. En effet, s'il permet de vérifier son existence lorsque celle-ci est dans un état stable, alors il permet aussi de lui trouver un état stable lorsque celle-ci est bien-contrainte. Ainsi, ce système fournit une description de l'objet géométrique duale de celle obtenue par la structure du modèle géométrique.

La description de la structure étant définie sous forme de graphes, les différentes relations générées à partir de ces données sont présentées. Ensuite, nous montrerons qu'à partir de ces relations, un ensemble nécessaire et suffisant de relations peut être construit pour représenter la structure correspondante.

8.3.1 Les relations 2D

L'approche choisie consiste à vérifier l'existence de la structure d'un objet à partir des paramètres internes de ses éléments. Une autre solution serait de positionner les différents éléments dans un repère. Avec cette approche, dès qu'une incompatibilité est identifiée, l'inexistence de la structure peut alors être directement annoncée. Un exemple d'incompatibilité est : deux bipoints théoriquement adjacents qui ne le sont pas du fait de leurs coordonnées. La solution proposée est plus en harmonie avec les données de la structure, puisque la raison même de son existence est de proposer une description relative et non pas une description absolue dans un repère quelconque.

Dans un espace de dimension deux, la structure est composée d'une partie définie exclusivement dans un espace vectoriel composé de **vecteurs** et d'**angles**. Les relations topologiques entre ces composants sont définies dans un graphe, dit vectoriel, où les nœuds définissent les **vecteurs** et les arcs représentent les **angles**. L'autre partie, définie dans un espace affine, composé de **spoints** et de **bipoints**, est portée par les éléments de l'espace vectoriel. Ici aussi, les relations topologiques entre les éléments sont organisées dans un graphe dit affine.

Les relations affines

Un graphe affine est composé de sommets représentant des points de l'espace affine. L'arc (a, b) représente un **bipoint**, un **copoint** ou un **multipoint** d'extrémité initiale a et d'extrémité finale b . En d'autres termes, il représente le vecteur **ab** dans l'espace affine associé.

Soit une chaîne d'origine A et d'extrémité B d'un graphe affine non orienté, orientons les arêtes de la chaîne telles que cette chaîne soit un chemin. Alors, ce chemin représente une décomposition du vecteur AB en vecteurs associés à des arêtes du graphe. Un chemin d'un graphe affine représente donc une somme de vecteurs dans l'espace affine. Par conséquent, un circuit d'un graphe affine exprime une somme nulle de vecteurs.

Soit un cycle du graphe orienté, définissons le sens de parcours du cycle par l'orientation d'un arc du cycle. Les autres arcs sont alors parcourus dans le sens positif s'ils sont de même orientation et dans le sens négatif, s'ils sont d'orientation contraire. Un cycle d'un graphe affine orienté permet donc d'écrire une relation vectorielle entre ses arcs. En effet, la somme des vecteurs associés aux arcs pondérés par leur orientation est égale au vecteur nul.

Dans un espace affine de dimension d , la projection d'un circuit sur d vecteurs formant une base permet d'écrire d relations indépendantes entre les coordonnées de ses vecteurs. De plus, dans cet espace il n'est possible d'écrire que d relations indépendantes.

Par exemple, prenons le cycle $\{(a, b), (c, b), (c, d), (a, d)\}$ d'un graphe, ce cycle représente la somme vectorielle $(\mathbf{ab} - \mathbf{cb} + \mathbf{cd} - \mathbf{ad})$ égale au vecteur nul. Or, dans notre modèle, un **bipoint** (c'est-à-dire un vecteur) est défini par un module et par un **vecteur support**. La relation vectorielle (8.6) est donc plus adaptée au modèle de données :

$$\|\mathbf{ab}\| \mathbf{e}_{ab} - \|\mathbf{cb}\| \mathbf{e}_{cb} + \|\mathbf{cd}\| \mathbf{e}_{cd} - \|\mathbf{ad}\| \mathbf{e}_{ad} = \mathbf{0}. \quad (8.6)$$

Soit \mathbf{e}_i les d vecteurs formant une base. La projection de cette somme vectorielle sur les vecteurs de la base produit donc le système d'équations indépendantes (8.7) :

$$\begin{cases} \mathbf{ab} \cdot \mathbf{e}_1 - \mathbf{cb} \cdot \mathbf{e}_1 + \mathbf{cd} \cdot \mathbf{e}_1 - \mathbf{ad} \cdot \mathbf{e}_1 = 0 \\ \dots \\ \mathbf{ab} \cdot \mathbf{e}_d - \mathbf{cb} \cdot \mathbf{e}_d + \mathbf{cd} \cdot \mathbf{e}_d - \mathbf{ad} \cdot \mathbf{e}_d = 0 \end{cases}. \quad (8.7)$$

Chaque équation est équivalente à la relation (8.8) et donc à la relation (8.9) :

$$\|\mathbf{ab}\| \mathbf{e}_{ab} \cdot \mathbf{e}_i - \|\mathbf{cb}\| \mathbf{e}_{cb} \cdot \mathbf{e}_i + \|\mathbf{cd}\| \mathbf{e}_{cd} \cdot \mathbf{e}_i - \|\mathbf{ad}\| \mathbf{e}_{ad} \cdot \mathbf{e}_i = 0 ; \quad (8.8)$$

$$\begin{aligned} \|\mathbf{ab}\| \cos(\widehat{\mathbf{e}_{ab}, \mathbf{e}_i}) - \|\mathbf{cb}\| \cos(\widehat{\mathbf{e}_{cb}, \mathbf{e}_i}) + \|\mathbf{cd}\| \cos(\widehat{\mathbf{e}_{cd}, \mathbf{e}_i}) \\ - \|\mathbf{ad}\| \cos(\widehat{\mathbf{e}_{ad}, \mathbf{e}_i}) = 0. \end{aligned} \quad (8.9)$$

Ainsi, les paramètres d'une structure ne sont pas compatibles entre eux s'il existe un cycle dans le graphe affine où l'une des n relations générées n'est pas vérifiée.

Ces équations sont construites à partir des **variables des bipoints** mais aussi à partir des **variables des angles** entre les **vecteurs support** et les **vecteurs de projection**. Or, le graphe vectoriel ne contient pas tous les **angles** possibles entre les **vecteurs** existants. Les **angles** manquants doivent alors être créés dans ce graphe. En effet, l'existence de l'objet passe par l'existence des différents **vecteurs** dans l'espace vectoriel, et celle-ci ne peut être vérifiée que si toutes les relations connues (les **angles**) entre ces **vecteurs** apparaissent dans le graphe vectoriel.

Les relations vectorielles

Un graphe vectoriel associé à un espace vectoriel de dimension 2 permet sur le même principe d'écrire des relations entre les éléments. En effet, les arcs définissent des angles orientés entre les vecteurs unitaires représentés par les

noeuds. Pour des angles orientés coplanaires, la relation de Chasles peut être appliquée. Un circuit dans le graphe vectoriel de dimension 2 représente donc une somme d'angles orientés égale à $0 [2\pi]$. Par analogie avec les cycles affines, un cycle pondéré par l'orientation des arcs définit une somme angulaire nulle modulo 2π . Soit, par exemple, $\{(e_1, e_2), (e_2, e_3), (e_4, e_3), (e_4, e_5), (e_1, e_5)\}$ un cycle du graphe. Ce cycle permet d'obtenir la relation angulaire :

$$(\widehat{e_1, e_2}) + (\widehat{e_2, e_3}) - (\widehat{e_4, e_3}) + (\widehat{e_4, e_5}) - (\widehat{e_1, e_5}) = 0 [2\pi]. \quad (8.10)$$

La dimension d'un espace vectoriel 2D dont les vecteurs sont tous unitaires n'est finalement que de un. En effet, comme le module de chaque vecteur est connu, il ne reste à préciser que son orientation, soit un paramètre. Ainsi, un cycle ne permet d'obtenir qu'une seule relation indépendante.

En conclusion, les paramètres vectoriels d'une structure ne sont pas compatibles entre eux, s'il existe un cycle dans le graphe vectoriel 2D qui ne vérifie pas la relation de Chasles.

8.3.2 Ensemble nécessaire et suffisant de relations

La sous-section précédente a proposé différentes relations permettant de contrôler la compatibilité entre les différents éléments de la structure. L'objectif de cette sous-section est de définir un ensemble de relations nécessaire et suffisant à ce contrôle.

Ensemble nécessaire de relations 2D

Pour un problème 2D, le graphe vectoriel définit les relations entre les vecteurs et les angles appartenant à un espace vectoriel 2D. La compatibilité des données de ce graphe avec le système d'équations algébriques est alors assurée lorsque la relation issue de chaque cycle de ce graphe est effectivement égale à un angle nul modulo 2π . En effet, si cette relation est nulle, alors la relation de Chasles est vérifiée pour les angles d'un cycle, ce qui implique que l'ensemble des vecteurs associés peut être disposé dans un plan vectoriel. En conséquence, si la relation de chaque cycle est nulle, l'ensemble des vecteurs du graphe peut être réparti dans un unique espace vectoriel de dimension deux.

Le graphe affine définit les relations entre les `pointlinks` et les `points` appartenant à un espace affine. Les `points` sont adjacents à des `pointlinks` et les extrémités des `pointlinks` sont des `points`. Le graphe affine contient des données compatibles entre elles à partir du moment où les paramètres des éléments vérifient les conditions de fermeture de chaque cycle du graphe. Or, un cycle affine est porté par un ensemble de vecteurs. Par conséquent, la compatibilité entre les éléments de ce cycle affine dépend en premier lieu de l'adéquation

entre les éléments du graphe vectoriel. Ceux-ci doivent être contenus dans leur espace de définition. La vérification de l'existence de la structure consiste alors à vérifier que tous les cycles du graphe affine forment effectivement un vecteur nul. Pour cela, il suffit de vérifier que les projections de chaque cycle sur tous les vecteurs d'une base de l'espace affine soient nulles.

Un graphe fini contient une base de cycles de dimension finie. Tout cycle du graphe peut alors être obtenu par une combinaison linéaire des cycles de la base. Par conséquent, le plus grand ensemble de relations indépendantes que l'on peut obtenir à partir d'un graphe est composé de toutes les relations indépendantes obtenues à partir de chaque cycle d'une base de cycles.

En conclusion, nous venons de montrer que dans un espace de dimension deux, toutes les relations obtenues à partir des cycles des graphes affines et vectoriels forment un ensemble de relations qui doit être nécessairement vérifié afin d'assurer l'existence géométrique de la structure qu'il représente.

Ensemble suffisant de relations vectorielles

Dans un espace vectoriel de dimension deux, à partir de la relation de Chasles, des relations angulaires sont obtenues. Elles traduisent la compatibilité de la valeur des angles des cycles du graphe vectoriel et donc des éléments de la partie vectorielle de la structure.

Nous allons montrer que dans un espace vectoriel de dimension deux, l'ensemble des relations obtenues à partir des cycles d'une base (base de cycles du graphe vectoriel) est suffisant pour assurer la compatibilité entre les éléments de cette partie de la structure.

Nous rappelons que cette partie est indépendante du reste de la structure. De plus, nous supposons que le graphe vectoriel est connexe. Soient n le nombre de **vecteurs** (et donc de nœuds) et m le nombre des angles (et donc d'arcs). Le graphe est connexe alors $(m + 1) \geq n$.

Comme les **vecteurs** sont des vecteurs unitaires, ils sont parfaitement définis par la donnée d'un seul paramètre. Un ensemble de n vecteurs a donc $(n - 1)$ degrés de liberté. Il suffit donc de connaître la mesure de $(n - 1)$ angles indépendants pour que cet ensemble soit rigide, la mesure des autres angles devant s'en déduire.

Nous avons supposé que le graphe contient m angles. Pour qu'il soit rigide, $(n - 1)$ angles doivent être connus. Il est donc nécessaire de connaître $(m - n + 1)$ relations indépendantes entre les angles afin d'en déduire la mesure des autres.

Supposons que ces relations soient connues ainsi que la mesure de tous les angles, alors, la partie vectorielle de la structure existe si elles sont toutes vérifiées. Il nous reste donc à montrer qu'à partir du graphe, $(m - n + 1)$ relations indépendantes peuvent être obtenues.

Or, le nombre cyclomatique¹ d'un graphe connexe contenant n nœuds et m arcs est $(m - n + 1)$ (Cf. annexe C.2). Le graphe contient donc $(m - n + 1)$ cycles indépendants qui permettent d'obtenir $(m - n + 1)$ relations indépendantes entre les paramètres angulaires. Cet ensemble de relations suffit donc pour contrôler l'existence de la partie vectorielle de la structure.

Cette première partie montre qu'à partir du graphe vectoriel, un ensemble suffisant de relations peut être généré afin de vérifier la compatibilité des données vectorielles. La même réflexion va être réalisée pour vérifier qu'un ensemble suffisant de relations peut être généré à partir du graphe affine.

Ensemble suffisant de relations affines

Nous allons montrer que dans un espace affine de dimension deux, dans lequel l'ensemble des mesures des éléments de l'espace vectoriel support sont compatibles entre elles, il est possible de définir un ensemble suffisant de relations permettant d'assurer la compatibilité des mesures de la totalité des éléments de la structure. Ces relations sont obtenues à partir de la projection de cycles du graphe affine sur une base quelconque de l'espace affine.

Nous rappelons que, contrairement à l'espace vectoriel qui est indépendant, l'espace affine est dépendant de l'espace vectoriel. Soit une structure connexe composée de n **spoints** et de m **bipoints**. Le graphe affine associé est connexe et contient donc n sommets et m arcs avec $(m + 1) \geq n$. Les bipoints de cette structure sont portés par un ensemble de **vecteurs**. Nous supposons que chaque **vecteur** ne porte qu'un seul **bipoint**. L'espace vectoriel support de cet espace affine contient donc m **vecteurs**. Pour différencier le nombre de **vecteurs** du nombre de **bipoints**, nous appelons ν le nombre des vecteurs, donc $m = \nu$. Le graphe vectoriel associé contient μ angles et est lui aussi connexe donc $(\mu + 1) \geq \nu$.

Un **spoint** dans un espace affine de dimension deux est un élément à deux degrés de liberté. Une structure de n **spoints** dans un espace de dimension deux a donc $(2n - 3)$ degrés de liberté. Pour que cette structure soit rigide, il est donc nécessaire de connaître la mesure de $(2n - 3)$ paramètres indépendants parmi les $(m + \mu)$ paramètres qu'elle contient. La mesure des autres paramètres doit en être déduit. Pour cela, il faut $(m + \mu - 2n + 3)$ relations indépendantes. Autrement dit, la connaissance de ces $(m + \mu - 2n + 3)$ relations vérifiées par les différents paramètres des graphes affines et vectoriels est suffisante pour démontrer l'existence de la structure. Il nous reste donc à montrer qu'à partir de ces deux graphes, ces relations indépendantes peuvent être obtenues.

Le nombre cyclomatique du graphe affine est $(m - n + 1)$. À partir de ce graphe, en 2D, $2(m - n + 1)$ relations indépendantes sont donc identifiables.

¹dimension de la base de cycles d'un graphe.

Le graphe vectoriel associé, quant à lui, permet d'écrire $(\mu - \nu + 1)$ relations indépendantes. Comme $m = \nu$:

$$(2(m - n + 1) + (\mu - \nu + 1)) = (m + \mu - 2n + 3).$$

Finalement, $(m + \mu - 2n + 3)$ relations ont donc été obtenues. Cet ensemble de relations est donc suffisant pour vérifier l'existence d'une structure 2D dans un état stable.

Cette réflexion a été faite en supposant que les graphes ne comportaient qu'une composante connexe et que les arcs du graphe affine ne sont que des **bipoints**. De plus, chaque **bipoint** est supposé être porté par son propre **vecteur support**. La démonstration sans ces hypothèses est seulement un peu plus complexe. En effet :

- si deux **bipoints** sont portés par un même **vecteur** commun, le nombre de degrés de liberté du système est diminué de 1. Effectivement, les couples de points portés par les deux vecteurs sont initialement définis dans un espace de dimension deux alors que cette contrainte les plonge dans le même espace de dimension un. Comme le nombre de **vecteurs** est alors diminué de un, le nombre de degrés de liberté du graphe vectoriel est diminué de un. Or le nombre d'équations générées par ce graphe est alors augmenté de un, le décompte précédent reste donc équilibré ;
- si un arc du graphe affine est un **copoint**, le nombre de degrés de liberté du système est diminué de *deux*. En effet, une configuration avec deux **spoints** coïncidents ne possède pas de degré de liberté relatif, mais comme un **copoint** n'a pas de mesure (ce qui équivaut à une mesure nulle) et n'est pas porté par un **vecteur**, le nombre de paramètres est diminué de deux : le décompte précédent reste donc équilibré ;
- si le nombre de composantes connexes du graphe affine est p , alors le nombre de composantes connexes du graphe vectoriel est au plus p . En effet, la structure d'un solide connexe (Cf. sous-section 1.3.1) doit être définie par un graphe affine et un graphe vectoriel connexes. En effet, pour que l'objet soit consistant, il faut qu'entre chaque couple de **spoints** puisse être défini un chemin, d'où une composante affine connexe. De plus, la compatibilité des éléments du graphe affine passe par celle du graphe vectoriel. Pour vérifier cela, il est donc nécessaire que les **vecteurs supports** des **bipoints** d'une composante connexe appartiennent à une unique composante vectorielle connexe. Par contre, deux composantes connexes d'un graphe affine peuvent être portées par une seule composante connexe du graphe vectoriel. Par exemple, lorsque les premières spécifications d'assemblage sont déclarées entre deux solides. Dans ce cas, le décompte précédent reste équilibré. En effet, le degré de liberté d'un couple de graphes vectoriels (anciennement connexe) lié par x arcs, tels que chaque arc ait un nœud dans chaque graphe,

est augmenté de un. Le nombre de paramètres augmente dans le même temps de x , alors que le nombre d'équations indépendantes n'augmente que de $(x - 1)$. En effet, sur les x arcs, le premier lie les deux composantes connexes en une composante connexe. Le nombre de cycles n'augmente pas. L'ajout de chaque nouvel arc crée ensuite un nouveau cycle. Comme $(x - 1)$ arcs sont ajoutés, $(x - 1)$ cycles sont créés. Le décompte précédent reste donc équilibré.

8.3.3 Conclusion

Pour un problème en dimension deux, à partir des cycles des bases des graphes affines et vectoriels, un ensemble nécessaire et suffisant de relations peut être formulé afin de vérifier la compatibilité de l'ensemble des paramètres géométriques. Or ce système d'équations est aussi le système nécessaire et suffisant traduisant l'ensemble des spécifications géométriques en vue de gérer la cohérence du problème global. Pour cette raison, les différentes étapes de la génération des équations suivent certaines règles afin d'obtenir un système de bonne qualité. Ceci est le thème de la section suivante.

8.4 Le générateur d'équations 2D

Contrairement aux solveurs géométriques présentés dans la première partie (Cf. chapitre 3) qui sont basés soit sur une mise en équations élémentaire (Cf. section 3.2) pouvant entraîner un système très imbriqué, soit sur une mise en équations guidée par un plan de résolution obtenu durant les étapes précédentes (Cf. section 3.3), l'approche proposée, qualifiée d'heuristique, essaie d'obtenir la meilleure mise en équations du problème géométrique pour le problème global. En effet, le système n'est obtenu que par l'ensemble des données géométriques. Les spécifications d'ingénierie ne sont, à ce niveau, pas prises en compte. Il n'est alors pas possible d'assurer la meilleure mise en équations d'un problème d'ingénierie, comme le garantissent les méthodes géométriques pour les problèmes purement géométriques (Cf. section 3.3).

Afin de présenter la méthode, la qualité d'un système sera définie. Ce qui permettra de clarifier et justifier l'aspect heuristique de la mise en équations. Ces deux précisions permettent alors l'exposé du principe.

8.4.1 Qualité d'un système

Une définition intuitive d'un système d'équations adapté est un système qui peut être résolu par étapes. En effet, il est toujours plus rapide de résoudre n équations

l'une après l'autre plutôt que de les résoudre simultanément. Cette définition a motivé la décomposition du système.

Pour un même problème, un bon système d'équations est donc un système qui peut être décomposé en blocs de plus petites dimensions. En effet, plus les PPSI sont de petites dimensions, plus les analyses qu'ils subiront seront pertinentes.

Pour être plus précis, on peut caractériser la qualité d'un système par sa complexité de résolution : plus cette complexité est faible, plus le système peut être décomposé en petites composantes, plus la qualité du système est "bonne". La complexité de résolution peut être définie par la somme de la complexité de résolution de chaque bloc. Celle-ci est donnée par la complexité de l'algorithme de résolution utilisée.

Une autre définition d'un bon système, toute aussi intuitive, peut être donnée à partir d'une comparaison avec la résolution d'un problème sur une planche à dessin. En effet, en 2D, il est courant de dire qu'un problème géométrique est bien paramétré s'il peut être construit avec les outils classiques du dessinateur industriel. Un système d'équations lié à un problème géométrique bien contraint est donc un très bon système s'il est dual d'une construction à *la règle*, *au compas et au rapporteur*. Cette définition est valable pour les mises en équations classiques (Cf. section 3.2), mais n'est pas réellement applicable ici. En effet, dans ces approches, chaque équation traduit une contrainte. Un bloc traduit donc une procédure de construction vérifiant un ensemble de contraintes. Par exemple, un point positionné par deux distances est constructible au compas et le système associé est composé de deux équations liées.

Les équations générées pour l'approche proposée ont une autre signification. En effet, soit l'exemple du problème de la figure 3.1-b dont le graphe affine ne comporte qu'un cycle où tous les **bipoints** ont des mesures connues (Figure 8.3-a), le graphe vectoriel, quant à lui, contient trois angles (représentés en pointillés à la figure 8.3-b) dont les valeurs ne sont pas connues. La cohérence de ces cycles traduit l'assemblage des trois parties rigides entre elles. En effet, ces parties sont déjà parfaitement définies par la structure puisqu'il est possible de connaître pour chacune la position interne de chacun de leurs points sans aucun calcul, alors que la position de l'une par rapport à l'autre n'est pas connue.

Comme la cohérence de la structure s'apparente à la cohérence de l'assemblage des différentes parties rigides, une étape élémentaire de résolution doit être duale d'une étape élémentaire de l'assemblage de parties rigides. Un bon système d'équations doit donc se décomposer en sous-systèmes tels que l'assemblage associé ne porte que sur le plus petit ensemble de parties déjà rigides.

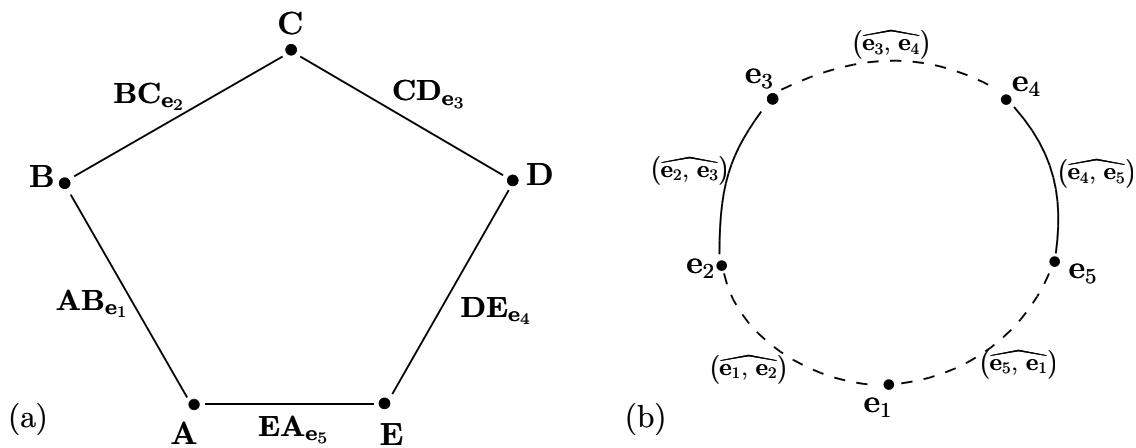


FIG. 8.3: Graphes affine et vectoriel du problème de la figure 3.1-b

8.4.2 Aspect heuristique

Dans les approches classiques, les méthodes permettant d'obtenir une bonne décomposition du problème sont basées sur des constatations géométriques. La décomposition obtenue n'est pas celle d'un système d'équations mais d'un problème géométrique. Nous avons vu que ces méthodes ne sont pas en mesure de traiter les problèmes où la géométrie est couplée à des contraintes d'ingénierie (Cf. section 3.4). Les approches algébriques (Cf. section 3.2) qui peuvent résoudre ce type de configurations ne sont pas plus performantes que dans le domaine purement géométrique pour les mêmes raisons : leurs générateurs d'équations ne fournissent pas de bons systèmes. Pour un problème donné, la résolution ne peut généralement être réalisée que sur le système global. Celui-ci est alors trop grand pour des temps de résolution et une fiabilité acceptables. L'incapacité de ces méthodes à offrir à l'utilisateur une analyse et des diagnostics dans de telles configurations n'en font donc pas des solutions acceptables.

L'idée du générateur proposé ici, est de baser la mise en équation du problème sur une série d'heuristiques favorisant la génération d'un bon système. Ainsi, que le problème soit purement géométrique ou couplé à des contraintes d'ingénierie, le système généré est, en raison des règles appliquées, probablement un bon système. Comme ces règles n'ont aucune origine géométrique, rien n'assure, pour un problème purement géométrique, la génération du meilleur système. En contrepartie, la qualité du système ne change pas dès que des relations d'ingénierie sont utilisées. Et en pratique, les systèmes obtenus sont de bonne qualité.

8.4.3 Stratégie du générateur d'équations

La qualité du système d'équations dépend, d'un point de vue subjectif, du nombre de variables et du nombre de variables puits par équation. En effet, plus une

équation lie de paramètres, plus il y a de risques qu'elle se lie à une grande partie du problème géométrique, moins il y a de chances qu'elle corresponde à l'assemblage d'un petit ensemble de parties rigides. De plus, pour qu'une équation permette de réaliser un assemblage élémentaire, elle ne doit contenir qu'une minorité de **variables puits**. En effet, en 2D un assemblage élémentaire se résout avec un système comportant au plus deux modules de **bipoints**.

Les différentes heuristiques proposées pour le choix des relations à générer sont donc toutes basées sur cette remarque.

Choix des cycles

Le premier choix influant pour la qualité du système d'équations est le choix des bases de cycles.

Pour le graphe affine, un cycle produit en 2D deux équations non linéaires qui sont généralement fortement imbriquées. En effet, tous les modules des **bipoints** du cycle apparaissent dans les deux équations. Or, pour un problème géométrique bien structuré, un bon système d'équations se décompose en composantes équivalentes à une procédure élémentaire d'assemblage. En d'autres termes, chaque composante doit permettre de résoudre les équations obtenues à partir d'un seul cycle affine. Pour obtenir en priorité de telles composantes, il faut donc s'assurer que chaque cycle affine contienne un minimum de variables comportant un maximum de variables sources. Pour cela, le graphe affine est pondéré. En effet, les arcs du graphe ne sont pas tous équivalents. Un **copoint**, qui est de mesure nulle, n'a pas d'influence sur les équations. Il ne doit donc pas avoir d'influence sur le poids d'un cycle. De même, un **bipoint** dont la **variable** est source ne doit pas avoir le même poids qu'un **bipoint** dont la mesure est inconnue. Ainsi, les arcs du graphe affine sont pondérés en fonction de leur nature et de leur **variable** attachée. La pondération proposée, basée sur les degrés de liberté de chaque élément dans l'espace de travail, devient :

- un **copoint** positionne parfaitement ses deux extrémités l'une par rapport à l'autre. Son poids est donc 0 ;
- un **bipoint** dont la **variable** est **source** a au moins un degré de liberté fixé dans son système composé de ses deux **spoints** extrémités. En 2D, il lui reste donc au plus un DDL. Son poids est donc 1 ;
- un **bipoint** dont la **variable** est **puits** a au plus deux degrés de liberté libres dans son système en 2D. Son poids est donc 2.

Dans ce graphe pondéré, on recherche alors une base de cycles de poids minimaux. L'algorithme développé est proposé en annexe C.5.2. Ainsi, comme les **copoints** ont un poids nul, leur nombre n'a pas d'importance. Ensuite, la différence de la pondération des **bipoints** en fonction de la nature de leur **variable** permet de favoriser les cycles composés d'un ensemble minimal de **variables** avec un

maximum de **variables sources**.

Le graphe vectoriel est pondéré sur les mêmes principes mais avec des poids différents. En effet, alors que le graphe affine en 2D est de dimension 2, le graphe vectoriel est en 2D de dimension 1. La pondération proposée, basée sur les degrés de liberté, devient :

- un **angle** dont la **variable** est **source** positionne parfaitement ses deux vecteurs l'un par rapport à l'autre. Son poids est donc 0 ;
- un **angle** dont la **variable** est **puits** définit un système de deux vecteurs indépendants entre eux. Ce système a donc 1 degré de liberté. Son poids est donc 1.

La pondération des graphes pourrait être plus précise. En effet, un **bipoint** dont le **vecteur** support est lié à un **angle** dont la variable est spécifiée, a sa direction imposée par rapport à un autre vecteur. Il a donc un degré de liberté en moins par rapport à un **bipoint** dont le vecteur n'est lié à aucun **angle** spécifié. De plus, nous verrons que les **angles** peuvent être de différentes natures. En effet, la génération d'une équation affine impose la création d'un ensemble d'angles. Ces angles ont donc moins de signification géométrique dans un problème que les autres angles. Ils devraient donc avoir des poids plus importants.

La base de projection

Alors que le passage d'un cycle vectoriel à une équation est direct et unique, la génération des équations associées à un cycle affine est fonction de la base de projection. Le choix de cette base est important. En effet, l'apport du modèle géométrique défini par une structure et une toile est de ne pas imposer de repère(s) global et/ou locaux, ces choix de repères n'ayant aucun intérêt pour la description d'un objet. Or ces repères sont nécessaires pour la mise en équations et le choix d'une base de projection n'est autre que le choix d'un repère local.

Un bon système d'équations, d'un point de vue numérique, est un système qui permet de minimiser les erreurs de troncature. En effet, toute cette approche est destinée à un système informatique qui ne travaille qu'à une certaine précision près. La base de projection choisie doit donc être la plus proche d'une base orthogonale car les imprécisions numériques sont fonctions des angles entre les vecteurs de la base et leur minimum est obtenu avec des axes orthogonaux. De plus, pour une base quelconque, les erreurs numériques peuvent devenir tellement importantes que les résultats ne sont plus significatifs.

La projection d'un cycle affine entraîne la création d'un ensemble d'angles supplémentaires pour le graphe vectoriel. Ces angles n'ont donc aucune raison d'exister autre que pour la projection de l'équation. Au contraire, les angles initiaux ont une véritable signification géométrique pour la structure. En effet, soit ils définissent l'angle entre deux bipoints adjacents, soit ils sont support d'une

spécification de l'utilisateur. Pour que les équations gardent une signification géométrique forte, il faut donc créer un minimum d'éléments complémentaires et donc un minimum d'angles de projection.

Pour répondre aux deux remarques précédentes, la première action visant à choisir la base de projection la plus orthogonale possible, tout en minimisant la création d'angles, consiste à choisir un premier **vecteur** de projection et à construire à partir de ce **vecteur** une base orthonormée directe. En effet, la projection d'un **bipoint** **AB** porté par un vecteur \mathbf{e}_i sur une base orthonormée directe $(\mathbf{O}, \mathbf{e}_x, \mathbf{e}_y)$ donne les composantes :

$$\begin{cases} \|\mathbf{ab}\| \cos(\widehat{\mathbf{e}_i, \mathbf{e}_x}) \\ \|\mathbf{ab}\| \cos(\widehat{\mathbf{e}_i, \mathbf{e}_y}) \end{cases} . \quad (8.11)$$

Comme la base est orthonormée directe en dimension 2 :

$$\cos(\widehat{\mathbf{e}_i, \mathbf{e}_y}) = \sin(\widehat{\mathbf{e}_i, \mathbf{e}_x}).$$

Les composantes deviennent finalement :

$$\begin{cases} \|\mathbf{ab}\| \cos(\widehat{\mathbf{e}_i, \mathbf{e}_x}) \\ \|\mathbf{ab}\| \sin(\widehat{\mathbf{e}_i, \mathbf{e}_x}) \end{cases} . \quad (8.12)$$

La conséquence est immédiate, il n'y a qu'un ensemble d'angles à construire par cycle et la base est orthonormée. En effet, supposons que le cycle contienne n **bipoints** , la projection de ce cycle sur deux vecteurs quelconques nécessite la connaissance de $2n$ angles, alors que la projection sur les vecteurs d'une base orthonormée ne demande que la connaissance de n angles.

Choix du premier vecteur de la base

Une série d'heuristiques a été proposée pour choisir le **vecteur** de projection. Les règles sont à suivre dans l'ordre tant qu'un choix est possible. À la fin des quatre règles, le **vecteur** est choisi arbitrairement parmi la sélection finale. Le **vecteur** de projection doit être :

- présent dans le cycle. Ce critère permet de favoriser l'utilisation d'angles significatifs pour l'équation. En effet, les angles définis par les vecteurs supports de **bipoints** du cycle ont, a priori, une signification géométrique locale alors que les angles définis par un seul **vecteur** appartenant au cycle n'en ont généralement pas ;
- commun à d'autres cycles. Ce critère permet de favoriser la projection de plusieurs cycles sur le même axe. Il minimise ainsi le nombre d'angles à ajouter dans le graphe vectoriel, et donc le nombre de cycles vectoriels ;
- utilisé par le maximum d'angles spécifiés. En effet, plus un **vecteur** est utilisé par des angles spécifiés, plus l'équation a de chances d'avoir un minimum d'inconnues angulaires ;

- utilisé par le maximum d'angles. Ce dernier critère permet, lorsque plusieurs vecteurs ont passé l'étape précédente, de sélectionner celui qui semble demander la création d'un minimum d'angles.

Ce choix du **vecteur** de projection permet donc de définir un repère local adapté pour la projection de chaque cycle. L'utilisation d'une véritable base orthonormée assure la minimisation des erreurs numériques. De plus, la création d'un ensemble minimal d'angles complémentaires favorise la génération d'un système angulaire composé d'un ensemble restreint d'équations. Toutes ces raisons permettent de favoriser la génération d'un système d'équations autorisant une résolution par étapes.

8.4.4 Prise en compte des multipoints

Le point courant d'une courbe est l'extrémité d'un **multipoint**. Si ce point est spécifié par un ensemble de contraintes géométriques, le **multipoint** apparaîtra dans un cycle qui sera lui aussi projeté dans une base bien adaptée. La projection de ce **multipoint** sur un vecteur \mathbf{e}_x est donc simplement donnée par la relation :

$$\widetilde{\mathbf{S}_0\mathbf{P}}(u) \cdot \mathbf{e}_x = \sum_{i=0}^n \|\mathbf{a}_i\| F_{i,m}^*(u) \mathbf{e}_i \cdot \mathbf{e}_x. \quad (8.13)$$

En effet, un **multipoint** $\widetilde{\mathbf{S}_0\mathbf{P}}(u)$ d'une courbe dont le réseau caractéristique est défini par les bipoints \mathbf{a}_i et dont les fonctions de formes sont les fonctions $F_{i,m}^*(u)$ est un vecteur qui vérifie la relation :

$$\begin{aligned} \widetilde{\mathbf{S}_0\mathbf{P}}(u) &= \sum_{i=0}^n \mathbf{a}_i F_{i,m}^*(u) ; \\ \widetilde{\mathbf{S}_0\mathbf{P}}(u) &= \sum_{i=0}^n \|\mathbf{a}_i\| F_{i,m}^*(u) \mathbf{e}_i. \end{aligned} \quad (8.14)$$

8.4.5 Génération dynamique des équations

Afin de pouvoir effectuer les différentes analyses en temps réel. Le système de relations doit être constamment remis à jour. Pour cela, il n'est pas concevable de devoir, à chaque modification, retrouver tous les cycles de chaque graphe. De plus une telle approche ne permet pas d'avoir une traçabilité des différentes analyses. En effet, les liens entre deux systèmes d'équations différents mais comportant un ensemble de relations communes, bien que pouvant être formulés différemment, sont alors très difficilement identifiables.

Pour répondre à ce besoin, la base des cycles n'est affectée que par des mises à jour. Au lieu de rechercher une base de cycles pour le graphe complet, la recherche est restreinte aux zones concernées par les modifications. Les algorithmes

développés permettant cette opération sont proposés à la sous-section C.5.3 en annexe.

Ainsi, après chaque modification, le plus petit sous-graphe affine contenant les cycles affines modifiés est identifié. Dans ce sous-graphe, une partie des cycles de la base sont supprimés et d'autres créés. La suppression d'un cycle entraîne la suppression d'un ensemble de relations affines. Or, ces relations ont nécessité la création d'un ensemble d'angles pour les projections. Ces angles sont alors supprimés à leur tour du graphe vectoriel. L'apparition des nouveaux cycles affines entraîne à son tour la création de nouveaux angles. Dans le graphe vectoriel, le même travail est réalisé, ce qui permet de réduire au maximum la modification des bases de cycles et donc l'impact des modifications dans le modèle algébrique.

8.5 Généralisation au 3D

Durant cette thèse, très peu de réflexions ont été réalisées sur la génération des équations pour un problème 3D. Cette section se contente de proposer une première ébauche de solution qui a l'avantage d'être parfaitement compatible avec le modèle géométrique.

La structure distingue d'un côté les éléments vectoriels coplanaires et, de l'autre côté, les éléments affines coplanaires. Grâce à ce découpage, pour chaque ensemble d'éléments coplanaires, un ensemble de relations peut être écrit mais cet ensemble de relations n'est pas suffisant. En effet, la compatibilité des éléments dans chaque plan affine et vectoriel est assurée mais la compatibilité dans l'espace ne l'est pas.

Supposons qu'il existe une base de cycles du graphe affine telle que chaque cycle soit contenu dans un plan. Cette base de cycles permet d'écrire un ensemble nécessaire de relations affines pour le problème global. Or, comme les cycles sont contenus dans des plans, au maximum deux équations par cycle sont générées puisque la troisième est dans tout les cas identiquement nulle. Cette configuration est obtenue pour les solides à surfaces planes, cylindriques ou sphérique, voir à surfaces quadriques.

De la même façon, il est possible de rechercher une base de cycles du graphe vectoriel comportant un maximum de cycles plans. On peut alors écrire un ensemble nécessaire de relations vectorielles exprimées dans ces plans.

Il reste alors un ensemble de cycles vectoriels qui n'est pas encore traité. De plus, en 3D, il existe un troisième graphe : le graphe sphérique. Ce graphe contient une base de cycles complémentaire aux cycles vectoriels non plans. Cet ensemble de cycles permet de compléter l'ensemble des relations 2D pour que le système global puisse permettre de résoudre une partie des problèmes 3D. De plus, il est possible de sélectionner parmi ces équations complémentaires un ensemble tel que

le système d'équations global soit nécessaire et suffisant pour traduire l'ensemble des spécifications géométriques dans le modèle algébrique. Les nouvelles relations sont écrites, issues des cycles d'une base du graphe sphérique, à partir des relations de la trigonométrie sphérique. La forme de ces relations est donc présentée dans un premier temps.

8.5.1 La trigonométrie sphérique

Définitions

La trigonométrie sphérique définit les relations entre les paramètres d'un triangle sphérique. Comme son nom l'indique, un triangle sphérique est un triangle dessiné sur une sphère unité. C'est donc la portion de la surface d'une sphère limitée par les arcs de trois grands cercles distincts, un grand cercle étant une section plane d'une sphère passant par son centre. On définit alors :

- un dièdre : angle formé par deux plans ;
- un trièdre : ensemble de trois plans vectoriels sécants en un unique point ;
- un grand cercle : section plane d'une sphère passant par son centre ;
- un angle sphérique : l'angle formé par l'intersection de deux arcs de grands cercles sur une sphère (angle Φ sur la figure 8.4). C'est donc l'angle entre les deux demi-plans supports des arcs. Par définition, un angle sphérique est défini entre $[0, \pi]$;
- un côté (sphérique) : arc de grand cercle joignant deux sommets d'un triangle sphérique (côté ϕ sur la figure 8.4). La longueur de cet arc est égal à l'angle entre les droites joignant les extrémités du côté au centre de la sphère.

Relations entre paramètres

Un triangle sphérique est défini par six paramètres : trois angles sphériques et trois côtés. Il suffit d'en connaître trois pour qu'il soit parfaitement défini. Les trois autres se déduisent à partir d'un ensemble de relations de la trigonométrie sphérique. Dans un premier temps, les principales formules [Korn *et al.* 1961], [Lefebure de fourcy 1836] sont rappelées :

$$\frac{\sin \delta}{\sin \Delta} = \frac{\sin \phi}{\sin \Phi} = \frac{\sin \psi}{\sin \Psi} ; \quad (8.15)$$

$$\cos \delta = \cos \phi \cos \psi + \sin \phi \sin \psi \cos \Delta ; \quad (8.16)$$

$$\cos \Delta = -\cos \Phi \cos \Psi + \sin \Phi \sin \Psi \cos \delta ; \quad (8.17)$$

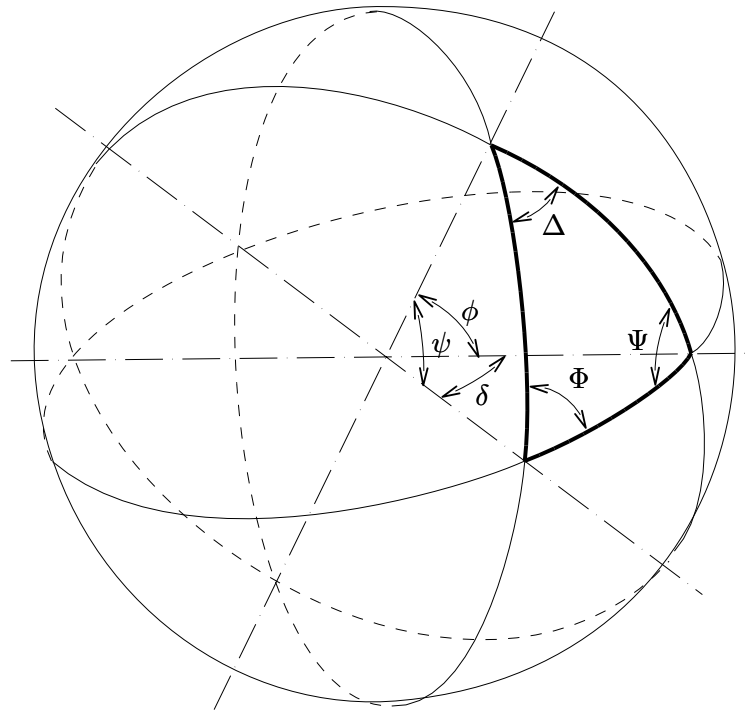


FIG. 8.4: Un triangle sphérique

$$\left\{ \begin{array}{l} \tan \frac{\phi + \psi}{2} \cos \frac{\Phi + \Psi}{2} = \tan \frac{\delta}{2} \cos \frac{\Phi - \Psi}{2} \\ \tan \frac{\phi - \psi}{2} \sin \frac{\Phi + \Psi}{2} = \tan \frac{\delta}{2} \sin \frac{\Phi - \Psi}{2} \\ \tan \frac{\Phi + \Psi}{2} \cos \frac{\phi + \psi}{2} = \cot \frac{\Delta}{2} \cos \frac{\phi - \psi}{2} \\ \tan \frac{\Phi - \Psi}{2} \sin \frac{\phi + \psi}{2} = \cot \frac{\Delta}{2} \sin \frac{\phi - \psi}{2} \end{array} \right. ; \quad (8.18)$$

$$\left\{ \begin{array}{l} \sin \frac{\Delta}{2} \sin \frac{\phi + \psi}{2} = \sin \frac{\delta}{2} \cos \frac{\Phi - \Psi}{2} \\ \sin \frac{\Delta}{2} \cos \frac{\phi + \psi}{2} = \cos \frac{\delta}{2} \cos \frac{\Phi + \Psi}{2} \\ \cos \frac{\Delta}{2} \sin \frac{\phi - \psi}{2} = \sin \frac{\delta}{2} \sin \frac{\Phi - \Psi}{2} \\ \cos \frac{\Delta}{2} \cos \frac{\phi - \psi}{2} = \cos \frac{\delta}{2} \sin \frac{\Phi + \Psi}{2} \end{array} \right. ; \quad (8.19)$$

$$\left\{ \begin{array}{l} s = \frac{\delta + \phi + \psi}{2} \\ S = \frac{\Delta + \Phi + \Psi}{2} \\ \sin \frac{\Delta}{2} = \sqrt{\frac{\sin(s - \phi) \sin(s - \psi)}{\sin \phi \sin \psi}} \\ \cos \frac{\Delta}{2} = \sqrt{\frac{\sin s \sin(s - \delta)}{\sin \phi \sin \psi}} \\ \sin \frac{\delta}{2} = \sqrt{\frac{-\cos S \cos(S - \Delta)}{\sin \Phi \sin \Psi}} \\ \cos \frac{\delta}{2} = \sqrt{\frac{\sin(S - \Phi) \sin(S - \Psi)}{\sin \Phi \sin \Psi}} \end{array} \right. . \quad (8.20)$$

Les polygones sphériques

Par analogie avec les triangles sphériques, des polygones sphériques peuvent être définis. Un polygone sphérique est donc un polygone dessiné sur une sphère dont les côtés sont des arcs de grands cercles.

Le problème est que les paramètres d'un polygone sphérique ne sont pas liés, comme ceux des triangles sphériques, par des relations connues. Pour pallier cela, un polygone sphérique sera décomposé en triangles sphériques afin de lier les différents paramètres qui lui sont associés. Les règles de cette décomposition sont présentées au chapitre 11 et leur justification est abordée à la section suivante.

8.5.2 Ensemble nécessaire et suffisant de relations 3D

Nous avons vu qu'un ensemble d'informations n'a pas encore été exploité pour la génération des équations. Cet ensemble est le graphe sphérique dont les nœuds sont les **normales** aux plans **vectoriels**. Comme pour les autres graphes, une base de cycles indépendants est recherchée, mais contrairement aux deux catégories précédentes, les cycles de ces graphes ne peuvent pas être contenus dans des plans. La relation de Chasles ne peut donc pas servir de base à la génération des équations entre les angles. Or, l'angle entre deux **normales** est dual à un dièdre. Par conséquent, un cycle d'angles entre vecteurs normaux est le dual d'un polyèdre sphérique². De la même façon qu'un trièdre définit un triangle sphérique, ce polyèdre définit un polygone sphérique. Or, les côtés d'un polygone sphérique ne sont autres que les **angles** entre les vecteurs définis par l'intersection des plans vectoriels. On retrouve donc les cycles du graphe vectoriel qui n'ont pas encore été pris en compte puisqu'ils ne sont pas plans. Ces informations doivent donc permettre de générer un ensemble nécessaire et suffisant de relations permettant de vérifier la compatibilité des éléments d'une structure de dimension trois.

Un polygone sphérique composé de n côtés et n angles est une structure comportant $(2n - 3)$ degrés de liberté. Il suffit donc de trois équations indépendantes pour assurer la compatibilité des éléments d'une telle structure. Les relations de la trigonométrie sphérique permettent d'obtenir ces trois équations pour les triangles. Pour les polygones sphériques, il n'existe pas de telles relations. Par contre, si l'on décompose un polygone sphérique en triangles sphériques et que pour chaque triangle les relations de la trigonométrie sont vérifiées alors on peut conclure.

8.5.3 Génération des relations de la trigonométrie sphérique

Afin de repousser au dernier moment la génération des équations permettant de décrire les polygones sphériques, ces relations ne sont définies que par le *nœud relation*. L'estimateur reste virtuel. En effet, ce n'est qu'à partir du plan de résolution que la stratégie de résolution d'un polygone peut être optimisée. Pour permettre la génération du plan de résolution, sachant qu'un polygone sphérique à $(2n - 3)$ degrés de liberté pour $2n$ paramètres, trois relations sont générées par polygones.

²Dans l'espace vectoriel, les faces d'un polyèdre sphérique sont sécantes en un point unique.

8.6 Des spécifications d'ingénierie au modèle algébrique

Nous avons vu que les contraintes d'ingénierie permettent de lier différents paramètres de différentes natures par un ensemble de relations. Tout comme pour les relations trigonométriques, il suffit de créer pour chaque spécification d'ingénierie le nombre nécessaire et suffisant de relations permettant de la transcrire au mieux. Chaque relation est alors composée d'un nœud relation liant les différents paramètres et d'un estimateur qui, la plupart du temps, représente simplement une équation algébrique alors que dans certains cas il est associé à une "boîte noire".

Le modèle géométrique apparaît pour les spécifications d'ingénierie comme un point fort de l'approche. En effet, les relations d'ingénierie relient généralement les paramètres géométriques du type longueur, angle, surface aux autres paramètres du problème. Dans cette approche, ces paramètres géométriques apparaissent directement dans le modèle géométrique. Dans les approches classiques, la géométrie est définie par des coordonnées de points dans un repère cartésien. La distance entre deux points n'est donc pas un paramètre du problème. Pour utiliser un tel paramètre dans une relation il est alors nécessaire de construire l'équation le liant aux coordonnées des points concernées.

8.7 Conclusion

Ce chapitre a permis de présenter le modèle algébrique. Ce modèle contient une description complète de l'objet sous une forme adaptée au maintien et à l'analyse de la cohérence des spécifications. Dans un deuxième temps, la transcription de l'ensemble des spécifications du modèle d'information au modèle algébrique a été proposée. Ce chapitre permet donc de clore la description d'un modèle dynamique de spécifications. La cohérence de ce modèle est gérée par deux modules complémentaires : le modeleur et le solveur. La partie suivante a pour objectif donc décrire le second de ces modules.

Troisième partie
Le solveur d'ingénierie

Chapitre 9

Le module de résolution

Ce chapitre présente le rôle du module de résolution pour le modèle dynamique d'expression des spécifications. Dans un premier temps, quelques termes sont introduits, permettant de clarifier les différents concepts énoncés. Ce chapitre aborde alors la présentation de l'algorithme global de résolution.

9.1 Introduction

Nous avons vu (Cf. section 4.3) que le modèle d'information regroupe l'ensemble de la description d'un produit et surtout que cet ensemble doit être maintenu cohérent par deux modules dont l'action est complémentaire. Le premier, couramment appelé "modeleur géométrique", maintient la cohérence topologique du modèle géométrique. Le second, appelé "solveur d'ingénierie", gère la cohérence de l'ensemble des paramètres d'un modèle en fonction de relations issues du modèle géométrique d'une part mais aussi en fonction des relations d'ingénierie complétant la description globale du modèle d'autre part.

Cette partie va donc présenter le "solveur d'ingénierie" qui, en raison des spécifications du modèle d'information, peut être vu comme un module de résolution indépendant. En effet, le modèle d'information contient un niveau de description algébrique du problème. C'est ce niveau qui est utilisé par ce module. En entrée, il attend les variables du problème spécifiées à travers un ensemble de relations algébriques. Une de ses sorties est alors un ensemble de valeurs admissibles pour les inconnues du problème.

Mis à part le rôle nécessaire de solveur, ce module répond à un ensemble de conditions permettant d'offrir à l'utilisateur le maximum d'informations sur l'état de son problème, comme l'état de contrainte ou de cohérence. Avant de présenter ce module, quelques termes sont définis. L'algorithme global de résolution sera ensuite détaillé.

9.2 Analyses et diagnostics

9.2.1 Analyse structurelle

Le module de résolution reçoit du modèle dynamique de spécification des relations essentiellement algébriques, mais pouvant être données sous la forme d'une "boîte noire". Or, nous savons qu'une résolution ne sera performante que si elle est réalisée par étapes. Le rôle de l'analyse structurelle consiste donc à rechercher une décomposition du système d'équations en sous-systèmes indépendants. Chaque sous-système est constitué d'un ensemble de relations mais aussi de paramètres dont il va permettre de préciser les valeurs admissibles. On dira qu'il est structurellement :

- bien contraint, lorsque le nombre d'équations est identique au nombre de variables ;
- sous-contraint, si le nombre d'équations est inférieur au nombre de variables ;
- sur-contraint, dans le cas où le nombre d'équations est supérieur au nombre de variables.

Le but de cette analyse est, d'une part, d'obtenir une décomposition du problème afin de rendre la résolution plus rapide, mais d'autre part, de pouvoir analyser l'état de contrainte d'un problème pour aider l'utilisateur dans sa tâche.

Cette décomposition est basée sur une analyse structurelle, c'est-à-dire que seules les dépendances des équations sont analysées. L'outil utilisé est le graphe biparti dont une description théorique détaillée est disponible en annexe C.6. Ce graphe est aussi utilisé par de nombreuses autres approches (Cf. section 3.2.4). Un graphe biparti est un graphe dont l'ensemble des sommets peut être partitionné en deux classes de sorte que deux sommets de la même classe ne soient jamais adjacents. Ce graphe est intéressant pour modéliser un système d'équations : le premier ensemble de nœuds regroupe les équations et le second les variables. Ce graphe a l'avantage de permettre une étude qualitative rapide.

L'information sur l'état de contrainte d'un système d'équations n'est que qualitative. En effet, la connaissance des propriétés structurelles d'un graphe biparti associé à un système d'équations n'est pas suffisante pour analyser complètement le système. Par exemple :

- un système d'équations structurellement sur-contraint est : soit redondant dans le cas où une équation est combinaison des autres équations, soit contradictoire, mais le graphe ne permet pas de distinguer ces deux cas ;
- un système structurellement sous-contraint est généralement vérifié par une infinité de solutions. Quoiqu'il en soit, un système apparemment sous-contraint peut admettre un ensemble dénombrable de solutions. Par exemple, le système sous-contraint $x^2 + y^2 = 0$ n'admet qu'une solution

dans $\mathbb{R}^2 : \{x = 0, y = 0\}$;

- de même, un système structurellement bien contraint peut contenir des équations redondantes ou contradictoires.

Cette approche logique permet donc une première analyse qui divise le système initial en un ensemble de petits systèmes. Dans tous les cas, la décomposition obtenue ne fausse pas l'état réel d'un problème. Une analyse plus fine est ensuite réalisée pour chaque PPSI¹. Cette analyse est l'analyse qualitative.

9.2.2 Analyse qualitative

L'analyse structurelle a permis de décomposer le système initial en PPSI. La critique de cette analyse a fait ressortir que l'aspect structurel du système n'est qu'une information sur son état réel. Une analyse plus approfondie est donc nécessaire. Cette analyse va devoir définir si un PPSI structurellement bien contraint est ou non correct, c'est-à-dire s'il a un nombre fini de solutions. Pour les mêmes raisons qui ont amené à décomposer le problème en sous-systèmes, l'analyse qualitative est réalisée sur chaque sous-système plutôt que sur le système global.

L'analyse qualitative est basée sur un ensemble d'hypothèses simplificatrices. En effet, nous supposons que :

- les équations sont des équations algébriques à coefficients réels ;
- un système de n équations algébriques à coefficients réels à n inconnues dont le jacobien est (presque partout) de rang n , a un nombre fini de solutions ;
- le rang du jacobien est le même presque partout ;
- le rang du jacobien en la solution est le même que presque partout ailleurs.

Les systèmes d'équations vérifiant ces hypothèses sont donc bien contraints si et seulement si [Lamure *et al.* 1998] :

- le déterminant du jacobien du système n'est pas identiquement nul ;
- le rang du jacobien du système en la solution est égal au rang du système.

L'analyse qualitative doit donc vérifier ces deux points. Un système ne vérifiant pas l'un des critères est mal contraint. En effet, un tel système contient au moins une équation redondante ou contradictoire. Le PPSI associé est alors localement sous-contraint puisqu'il a moins d'équations indépendantes que d'inconnues.

Lorsque l'analyse constate qu'un sous-système est mal contraint, il faut alors identifier une équation redondante dans le système. Par définition, cette équation peut être supprimée. Le plan de résolution doit alors être corrigé. En effet, le sous-système alors sous-contraint doit chercher à se regrouper avec d'autres sous-systèmes du problème. Cette étape sera présentée à la sous-section 10.3.4.

¹Plus Petit Système Indépendant

9.2.3 Diagnostics post- ou pré-résolution

Un diagnostic consiste à faire comprendre à l'utilisateur l'état de contrainte de son problème (suite à une analyse pré-résolution) ou les raisons d'un échec de résolution (suite à une analyse post-résolution) à partir d'un ensemble d'informations. Ces informations doivent être comprises par l'utilisateur et doivent utiliser son vocabulaire. Ces diagnostics sont en partie liés au modèle de spécifications. Les informations fournies à l'utilisateur ne sont qu'une traduction du plan de résolution dans son langage de spécifications. En effet, la résolution d'un PPSI permet de trouver des valeurs admissibles pour un ensemble de variables, c'est-à-dire pour un ensemble de paramètres du problème. Suite à la résolution d'un PPSI, la forme d'une partie de la géométrie est alors fixée, tout comme certains paramètres d'ingénierie voient leur valeur définie. Ainsi, il est possible d'illustrer les différentes étapes de résolution, les parties sur- et sous-contraintes, et la zone dont la résolution a échoué.

Un diagnostic pré-résolution est la conclusion d'une analyse réalisée pendant la construction du modèle afin de fournir à l'utilisateur le maximum d'informations sur l'état de contrainte local de son modèle.

Le diagnostic post-résolution est constitué, en cas d'échec de la résolution, d'une analyse des causes. Elle est définie par l'ensemble des spécifications mises en jeu et des parties précédemment résolues qui apparaissent comme des parties rigides pour le sous-système mis en échec. La cause d'un échec peut être soit :

- un problème d'existence. L'exemple typiquement est un triangle spécifié par la mesure de ces trois côtés mais ne vérifiant pas l'inégalité triangulaire ;
- un problème de cohérence. Ce type de cause apparaît dans les problèmes sur-contraints inconsistants ;
- un problème numérique. Lorsque l'algorithme utilisé est numérique, la convergence de la résolution dépend de la régularité du système. Il est donc possible qu'elle ne converge pas alors que des solutions existent (Cf. comportement chaotique de la méthode de Newton-Raphson).

9.3 Résolution

9.3.1 Plan de résolution

La tâche première du module de résolution est de résoudre un système d'équations. Pour les problèmes traités, ce système peut comporter un nombre important de relations. Or, les algorithmes de résolution numérique sont des algorithmes dont la complexité est au mieux en $O(n^3)$. Simplement, en raison de cette première remarque, il est nécessaire de chercher à résoudre le système par étapes. En effet, il est toujours plus rapide de résoudre n systèmes indépendants

les uns après les autres que de les résoudre en une seule fois.

La sous-section 8.4.1 a présenté la qualité d'un système d'équations. Il a été dit qu'un système d'équations adapté à la résolution est un système qui peut être résolu par étapes, où seule une sous partie est résolue. Or, en raison des dépendances entre les différentes variables et équations, l'ordonnancement des étapes de résolution est uniquement déterminé. La résolution d'un bloc ne peut donc se faire qu'à un moment donné.

Un plan de résolution décrit l'ordonnancement des différents blocs, c'est-à-dire des sous-systèmes indépendants. Pour cela, il sera donc représenté par un simple graphe orienté où les noeuds représentent les sous-systèmes composés d'un ensemble de relations permettant de déterminer un ensemble de variables et les arcs, les dépendances entre sous-systèmes (Figure 9.1).

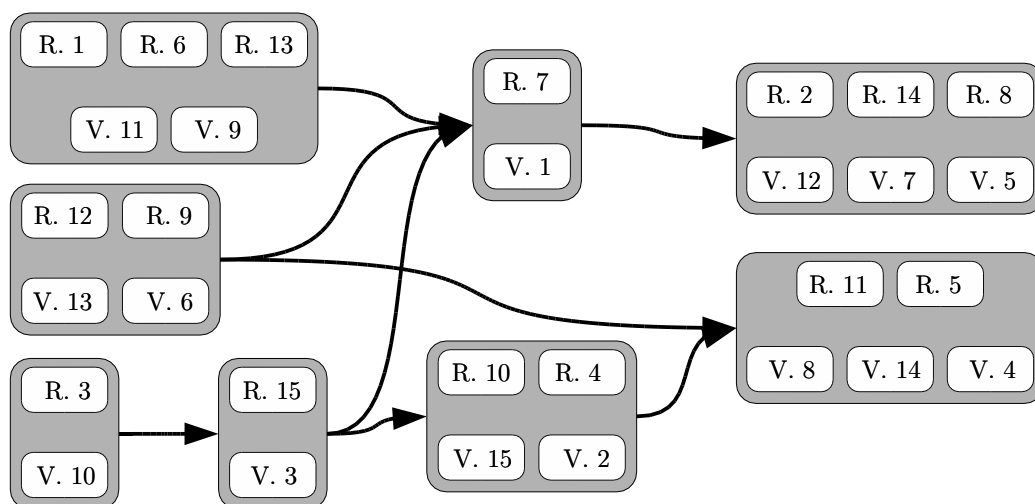


FIG. 9.1: Un plan de résolution représenté par un simple graphe orienté.

Pour des raisons évidentes, chaque sous-système doit être le plus petit possible. C'est pourquoi, un sous-système du plan de résolution sera appelé *PPSI* pour "Plus Petit Système Indépendant" [Ait-Aoudia *et al.* 1993].

9.3.2 Critère de choix des algorithmes de résolution

La résolution d'un PPSI consiste à trouver toutes ou une partie significative de ses solutions. En effet, la présentation de l'arbre de solutions a fait ressortir qu'une solution d'un PPSI n'est pas obligatoirement celle du problème global. Afin d'assurer une solution finale, il faut donc être en mesure d'en trouver plusieurs pour chaque PPSI.

Cette résolution doit aussi être la plus fiable possible. Pour ne pas rechercher en vain ou annoncer un faux diagnostic, l'idéal serait de savoir si un PPSI admet ou non des solutions. Ceci permet de différencier l'échec numérique de l'inexistence de solution et de gagner un temps précieux.

La solution attendue n'a pas besoin d'être exacte. En effet, la plupart des variables sont associées à des mesures de pièces mécaniques "classiques" dont la précision de fabrication est de l'ordre du centième de millimètre voire du micron. La précision de la définition des pièces peut donc se contenter d'être légèrement meilleure. En revanche, l'utilisateur conçoit un mécanisme d'une façon interactive. Le temps de réponse du solveur doit être le plus rapide possible. C'est pourquoi les solveurs sont basés sur des algorithmes de résolution numérique.

9.4 Arbre de solutions

La résolution d'un problème géométrique peut sembler simple à partir du moment où le plan de résolution est bien contraint. Il suffit de résoudre chaque PPSI dans l'ordre du plan de résolution. Malheureusement, dans la pratique ce n'est pas le cas. En effet, chaque PPSI composé d'un ensemble d'équations non linéaires admet soit aucune solution, soit une solution, soit un nombre fini de solutions. Dans le cas général, il n'est pas possible de connaître ce nombre contrairement à quelques configurations courantes où une solution algébrique est connue.

Un arbre de solutions est un graphe sans cycle dans lequel chaque nœud représente l'état de résolution d'un problème et les arcs définissent les chemins entre les différents états et l'état initial. La notion d'arbre de solutions a été introduite par Mathis [Mathis 1997].

L'état de résolution représente l'état de l'ensemble des variables du problème. En effet, une variable est soit **source**, soit **puits**. Lorsqu'elle est **puits** sa valeur est soit connue, soit inconnue si elle appartient à un PPSI non encore résolu. À l'état initial, seules les variables sources ont une valeur définie. La résolution d'un PPSI change l'état de résolution du problème puisque la valeur d'un nouvel ensemble de variables devient connue.

Un niveau de nœuds de l'arbre de solutions est associé à un PPSI. Il contient autant de nœuds qu'il existe de solutions différentes au système composé des relations des PPSI traités (incluant le PPSI courant).

Les équations d'un PPSI sont dépendantes de l'état de résolution précédent. En effet, ces équations contiennent des paramètres, c'est-à-dire des variables dont les valeurs ont été calculées grâce aux PPSI précédents. Les arcs relient donc les valeurs des paramètres d'un PPSI (solutions des PPSI précédents) aux solutions de ce PPSI, ceux-ci deviennent alors paramètres pour les PPSI suivants. Il y a autant d'arcs liant un état de résolution (un jeu de valeurs de paramètres) que de solutions à un PPSI muni de ce jeu de valeurs.

Un deuxième arbre peut être défini : un arbre dans le corps des complexes. Alors que les branches de l'arbre de solutions réelles se divisent en fonction du nombre de solutions réelles de chaque PPSI, les branches de l'arbre complexe se

divisent en fonction du nombre de solutions complexes. Cet arbre permet d'obtenir un majorant du nombre de solutions à un problème (Cf. sous-section 9.4.1).

Pour illustrer cette définition, l'arbre de solutions théoriques et réelles (en traits épais) du plan de résolution séquentiel de la figure 9.2-a est présenté à la figure 9.2-b. Le premier PPSI du plan de résolution n'admet qu'une solution. Il n'y a donc qu'une branche entre les paramètres et l'état de résolution du PPSI 1. Le PPSI 2 admet deux solutions. À ce niveau, l'arbre de solutions se divise en deux branches. Supposons que le PPSI 3 soit une équation du second degré, il peut donc avoir zéro, une ou deux solutions. En raison du premier état de résolution associé au PPSI 1, ce PPSI n'admet qu'une solution, la première branche de l'arbre ne se divise donc pas, alors qu'en théorie elle se divise en deux. Par contre, la seconde branche se divise, ce PPSI admet ici, deux solutions. Le dernier PPSI est un système non linéaire qui admet aussi jusqu'à trois solutions. Dans cet exemple, en fonction de la branche de l'arbre, ce PPSI admet zéro, une ou trois solutions.

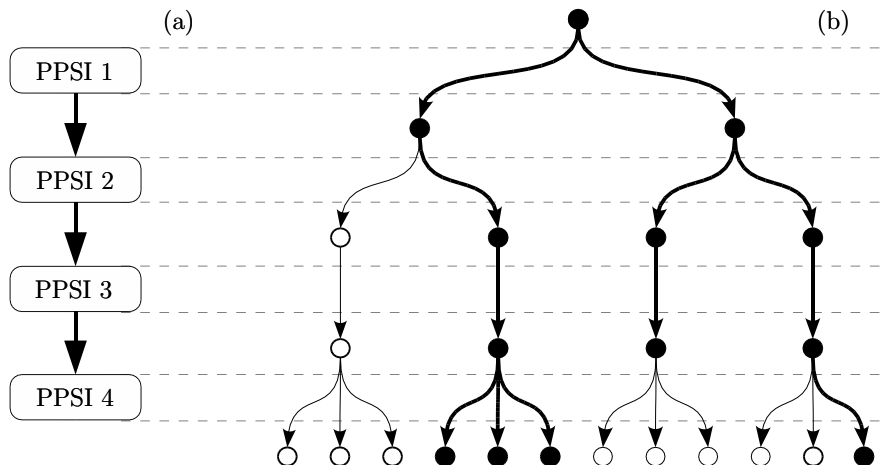


FIG. 9.2: (a) un plan de résolution séquentiel, (b) l'arbre de solutions associé

Un arbre de solutions permet de bien illustrer les difficultés que l'on peut rencontrer lors de la résolution d'un problème. En effet, il représente l'évolution du nombre (théorique et réel) de solutions pendant le processus de résolution. On comprend bien que pour certains problèmes, il est possible qu'une solution globale ne soit obtenue qu'après avoir parcouru une grande partie de l'arbre de solutions. Ceci impose au module de résolution d'être en mesure de scruter l'arbre de solutions, et a plusieurs conséquences :

- il faut être capable de trouver différentes solutions à un PPSI donné ;
- il faut connaître le nombre de solutions d'un PPSI donné pour ne pas chercher indéfiniment une nouvelle solution lorsqu'il n'y en a pas ;
- il faut connaître les branches parcourues.

La difficulté principale d'une telle approche est donc l'aspect combinatoire de la

recherche de solution(s).

9.4.1 Approximation du nombre de solutions

Le plan de résolution permet, sous certaines conditions, d'estimer le nombre maximum de solutions à un problème. La première condition est d'être en mesure de connaître le nombre théorique de solutions d'un PPSI. Nous verrons que dans les cas courants, ce nombre est connu. Pour les autres PPSI, ce nombre pourra, dans le meilleur des cas, être estimé. Donc, connaissant pour chaque PPSI son nombre théorique de solutions, le nombre global de solutions peut alors être estimé par :

$$S_{max} = \prod_i S(PPSI_i), \quad (9.1)$$

avec $S(PPSI_i)$ le nombre de solutions théoriques pour le PPSI d'indice i .

Ce nombre n'a pas réellement d'intérêt en soi puisqu'en pratique il est difficilement estimable. En effet, le nombre de solutions théoriques d'un système quelconque n'est pas connu. Il est évoqué ici car il montre l'importance de l'aspect combinatoire dans la recherche d'une solution.

9.4.2 Multiplicité des solutions

La possibilité d'offrir des solutions à un problème est un point non négligeable. En effet, pour des raisons de cohérence topologique ou simplement pour des raisons propres à l'utilisateur, une solution peut être refusée. Le système doit alors être capable de fournir des solutions supplémentaires.

Pour répondre à ce besoin, le plan de résolution est un apport majeur. En effet, le problème initial est décomposé en sous-problèmes. De la même façon que la décomposition facilite la résolution, elle favorise la recherche de plusieurs solutions. Et ce, pour plusieurs raisons :

- dans la majorité des problèmes, pour une partie des PPSI extraits, une résolution algébrique est possible. Ce point est détaillé aux sous-sections 11.2.5 et 11.2.6. Ainsi, pour ces PPSI, toutes les solutions analytiques sont connues ;
- pour les autres PPSI, la recherche de plusieurs solutions est possible par des approches numériques (Cf. sous-section 11.2.4). Or, pour rester viable du point de vue temps de résolution, cette recherche doit être lancée sur des PPSI de petites dimensions.

9.4.3 Existence d'une solution

Un système d'équations "bien contraint" admet un nombre fini de solutions. En effet, prenons l'exemple d'un système traduisant les problèmes géométriques de la

figure 9.3. Chaque triangle est spécifié par un ensemble nécessaire et suffisant de contraintes dimensionnelles. Le système d'équations associé a donc théoriquement une ou plusieurs solutions. Or, il existe des conditions d'existence d'une solution (Cf. [Serré 2000]). Par exemple, le premier triangle doit vérifier l'inégalité triangulaire (c'est-à-dire que la longueur de chaque côté doit être inférieure à la somme des deux autres). Ces problèmes sont simples. L'inexistence d'une solution dans le cas général est beaucoup plus difficile à prouver.

La décomposition du problème global en PPSI permet tout de même de réduire la dimension du problème. Ainsi, les éventuels problèmes d'existence peuvent être localisés à des sous-parties. Démontrer l'inexistence d'une solution consiste alors à montrer qu'il n'existe pas de chemin dans l'arbre des solutions atteignant une solution finale.

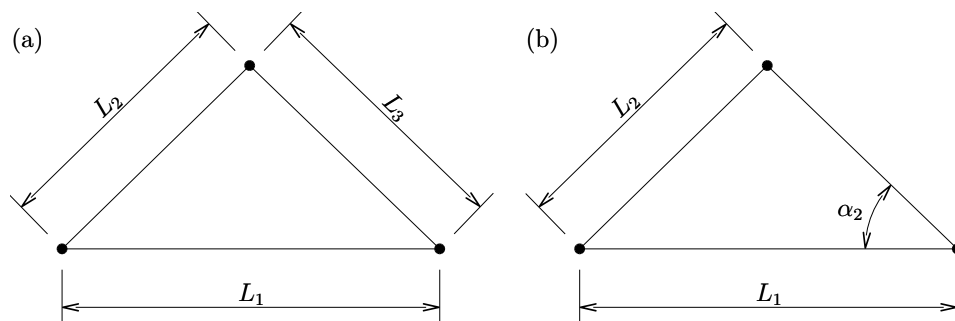


FIG. 9.3: Conditions d'existence de deux problèmes géométriques simples : (a) un triangle dont la longueur des trois côtés est spécifiée doit vérifier l'inégalité triangulaire sur chacun des côtés, (b) un triangle dont deux côtés L_1 et L_2 et l'angle opposé α_2 sont spécifiés, doit vérifier la relation : $L_1 \sin \alpha_2 < L_2$.

9.5 Algorithme global de résolution

Le module de résolution présenté ici, est un module globalement indépendant du modèle de spécifications. En effet, sa tâche première est de résoudre un système d'équations. Nous verrons, d'autre part, que pour certaines tâches, il reste tout de même fortement dépendant de ce modèle.

Ce module est un “solveur dynamique” pour l'ingénierie. Effectivement, contrairement aux solveurs géométriques, (Cf. chapitre 3) qui sont avant tout destinés à résoudre un problème géométrique, celui-ci se contente de résoudre un système d'équations pouvant être de grande taille. L'aspect géométrique du problème a complètement disparu à ce niveau. Il est qualifié de “dynamique” car, en plus de résoudre le système, il doit assister l'utilisateur tout au long de sa conception et préparer continûment la résolution. Ceci est dû à l'enrichissement progressif du modèle. Le système d'équations évolue en parallèle. Afin de

résoudre rapidement et aussi de jouer son rôle d'assistant, le solveur doit donc constamment mettre à jour le plan de résolution qui est à la base de toutes ses missions. La première d'entre elles est de trouver une solution avec un temps de réponse permettant au système de prétendre être interactif. La décomposition du problème est une des clés de cette spécificité. Elle autorise la mise en évidence de plusieurs solutions pour un même problème. En effet, il est toujours plus facile de trouver plusieurs solutions à un système de petite dimension possédant des caractéristiques particulières pour lequel différents algorithmes peuvent être lancés, que pour un grand système quelconque où les algorithmes basiques sont déjà difficilement maîtrisables. Toujours dans cette même idée, l'inexistence d'une solution pourra dans certains cas et grâce à cette décomposition, être prouvée. Une fois que l'inexistence d'une solution est prouvée, une explication va devoir être fournie par le module pour que l'utilisateur puisse la comprendre et facilement la corriger.

Le solveur connaît l'ensemble des relations du modèle mathématique et surtout ses modifications successives. À partir de ces données, il réalise un ensemble de tâches décrites par l'algorithme de la figure 9.4 afin de mettre en permanence à jour le plan de résolution et autoriser une résolution rapide et aussi rigoureuse que possible.

Suite à une *mise à jour* du système d'équations (ajout et suppression d'un ensemble de relations et de variables), la première étape consiste à réaliser une analyse structurelle. Pour des questions de performance, cette analyse n'est réalisée que sur le sous-système affecté par les modifications. Cette analyse permet de modifier en conséquence le plan de résolution. Une analyse qualitative valide le plan ou lance une procédure de corrections. Suite à ce travail, un diagnostic de pré-résolution sur l'état de contrainte local du problème peut alors être fourni à l'utilisateur. Le système est alors prêt pour une résolution et pourra être suivi d'un diagnostic post-résolution en cas d'échec.

9.6 Conclusion

Ce chapitre a présenté le module de résolution de façon globale. Ses différentes tâches y ont été introduites. L'algorithme global a été proposé afin de donner une vision générale du fonctionnement de ce module. Les chapitres suivants vont détailler les trois grandes tâches de ce module, c'est-à-dire : la préparation du plan de résolution, la résolution, et l'élaboration de diagnostics.

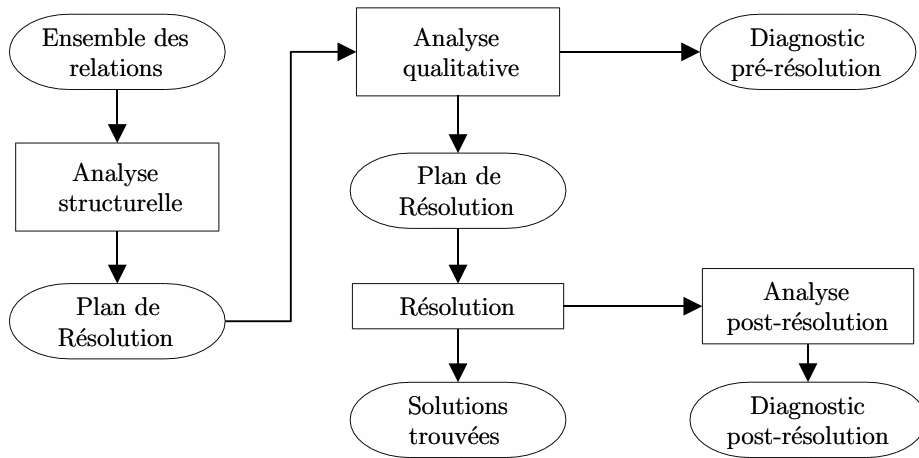


FIG. 9.4: Algorithme global de résolution.

Chapitre 10

Plan de résolution

Le plan de résolution précise les étapes de résolution d'un problème. Il définit donc le découpage du système global et l'organisation des sous-systèmes à résoudre. Ce plan est obtenu à la suite d'une analyse structurale puis approfondie par une analyse qualitative. La présentation de ces deux analyses ainsi que la stratégie de gestion dynamique du plan de résolution constituent les trois grands thèmes de ce chapitre.

10.1 Introduction

Un système d'équations défini dans $E \subset \mathbb{R}^n$ admet un ensemble dénombrable de solutions s'il est composé de n équations non-dégénérées au voisinage des solutions. Nous appellerons un tel système, un système bien contraint.

La démonstration de cette propriété fait appel à des éléments de géométrie différentielle dont les définitions nécessaires sont rappelées en annexe E. En effet, soit E un ensemble vectoriel de dimension finie $\dim(E) = n$, V une partie de E et ϕ un système de n équations non-dégénérées en V . D'après l'annexe E, cet ensemble d'éléments sont précisément les conditions de la définition d'une sous-variété. En conséquence, V définit par ϕ une sous-variété de E .

De plus, comme ϕ est un système de n équations, la codimension de cette variété est n . Or, comme la dimension de l'espace de définition E est aussi n , nous sommes dans le cas particulier où la dimension de V est 0. En conséquence, la partie V de E est une partie discrète de E , la solution de ϕ est constituée de points isolés. V est donc un ensemble de points dénombrables.

La condition de non-dégénérescence est essentielle [Demazure 1989]. En effet, en l'absence d'une telle condition, l'ensemble des zéros communs d'une famille de fonctions de classe C^∞ peut-être extrêmement pathologique.

Rien n'empêche cependant le système d'équations d'être dégénéré ailleurs. En effet, on peut montrer qu'un système d'équations non-dégénéré au voisinage de

ses zéros ne sera dégénéré que sur un ensemble de mesure nulle.

La préparation du plan de résolution est basée sur les conséquences de ce théorème. En effet, tout sous-système d'un ensemble de relations, non dégénéré au voisinage des solutions et de codimension maximale, admet lui aussi un ensemble fini de solutions. La première étape de la préparation du plan de résolution doit donc décomposer le système initial en sous-systèmes structurellement bien contraints. La seconde étape va valider la décomposition par une analyse qualitative des sous-systèmes.

Ce chapitre va présenter dans une première section l'analyse structurelle puis il introduira l'analyse qualitative. Cette dernière analyse, en cas de diagnostic négatif, entraîne la correction du plan de résolution. Celle-ci est réalisée grâce à une gestion dynamique du plan de résolution qui sera présentée dans une dernière section.

10.2 Analyse structurelle

L'analyse structurelle permet d'obtenir une décomposition d'un système d'équations en composants structurellement bien contraints. Elle utilise pour cela la théorie des graphes bipartis (Cf. annexe C.6). Le graphe utilisé est un graphe où les nœuds sont pondérés en fonction de leur nature. La raison de cette pondération est présentée dans un deuxième temps afin de bien apprécier son impact.

10.2.1 Algorithme

L'analyse structurelle procède en trois grandes étapes (Figure 10.1). Dans un premier temps, un couplage maximum prenant en compte la pondération des nœuds est réalisé.

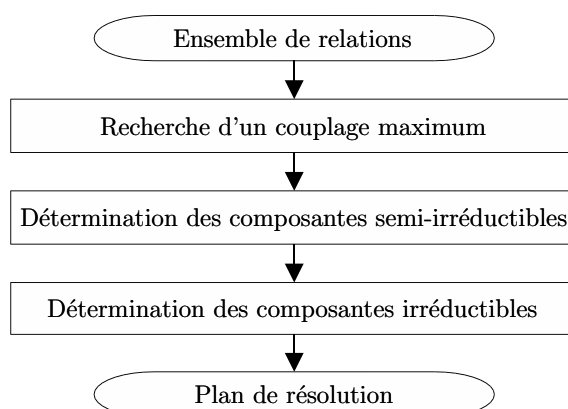


FIG. 10.1: Algorithme d'analyse structurelle

Recherche d'un couplage maximum

Un couplage maximum prenant en compte la pondération des nœuds est réalisé. L'algorithme utilisé n'est pas celui de Hopcroft et Krap présenté en annexe (Cf. annexe C.6.1). Il se décompose en deux phases :

- rechercher un couplage complet ;
- rendre le couplage maximum.

La recherche d'un couplage complet d'un graphe $G = [S \cup T, E]$, c'est-à-dire un couplage qui ne laisse pas deux sommets adjacents insaturés, est assez simple. En effet, il suffit de parcourir tous les sommets d'une des classes S ou T et de choisir dans les sommets adjacents au sommet courant un candidat pour le couplage.

L'intérêt de cette première étape est d'orienter le choix des sommets à saturer en fonction de leurs caractéristiques. Les sommets de l'ensemble des relations sont ordonnés afin de saturer dans un premier temps les équations de plus fort poids¹ et de plus faible degré. Ainsi, les équations qui ne seront pas couplées, ont une forte probabilité d'être des équations qui imbriquent fortement un système.

Pour chaque équation, la variable choisie dans la liste des variables adjacentes sera celle de plus fort poids et de plus faible degré relatif. Le degré d'un nœud est le nombre d'arêtes adjacentes (Cf. annexe C.2.2). Afin d'assurer la saturation d'un maximum de variables, l'idée est de réaliser le couplage dans un sous-graphe ne contenant que les nœuds pas encore saturés. Ainsi, l'algorithme a une bonne vision de l'ensemble des arêtes encore candidates. Pour cela, l'idée a été de définir le degré relatif d'un nœud par le degré de ce nœud dans le sous-graphe défini ci-dessus.

Pour finir, afin de rendre maximum le couplage, il suffit, selon le théorème de Berge (Cf. annexe C.6.1), de rechercher s'il existe ou non des chaînes alternées augmentantes.

Complexité algorithmique Soit un graphe biparti $G = [S \cup T, E]$ composé de $|S| + |T| = 2n$ nœuds et $|E| = m$ arêtes. On suppose que chaque nœud connaît son degré et son degré relatif, ce qui lui impose de modifier ces nombres dès que nécessaire. On suppose que le degré des nœuds est proche du degré moyen, soit m/n .

Les sommets *équations* doivent être ordonnés, soit une complexité en $O(\log n)$. Pour chaque *équation* il faut choisir le sommet *variable* à saturer parmi m/n variables, soit une complexité en $O(m)$. Saturer un arc impose aux sommets adjacents de l'*équation* saturée de mettre à jour leur degré relatif, soit une complexité totale en $O(m/n)$. Finalement, la première partie de cette étape a une complexité en $O(2m + \log n)$. Pour un graphe très fortement imbriqué, m est de l'ordre de

¹Le poids des nœuds est introduit aux sous-sections 10.2.2 et 10.2.3.

n^2 (le cocycle d'un nœud de S est T). La complexité est alors de la forme $O(n^2)$. Il faut savoir que pour un graphe moyennement imbriqué, m est de l'ordre de kn , $k \in \mathbb{N}$, $k \ll n$, la complexité est alors en $O(kn)$. Il faut ajouter à cela la complexité de la recherche des chaînes alternées augmentantes.

Détermination des composantes semi-irréductibles

La détermination des composantes semi-irréductibles consiste à rechercher à partir du graphe, les parties structurellement sous et sur-contraintes du problème. L'algorithme utilisé est décrit en annexe (Cf. annexe C.6.2).

Une composante semi-irréductible est une composante structurellement irréductible dont un des ensembles de nœuds contient plus d'éléments que l'autre. Le système d'équations associé est donc obligatoirement dégénéré à cause de la structure de sa matrice jacobienne. En fonction des cas, une composante semi-irréductible est associée à un sous-problème sur ou sous-contraint.

La détermination des composantes irréductibles est une étape nécessaire pour offrir à l'utilisateur un diagnostic sur l'état de contrainte d'un problème. Quoiqu'il en soit, une composante irréductible doit aussi être résolue afin d'offrir des solutions aux problèmes sous- et sur-contraints consistants ou de diagnostiquer l'inexistence de solution. Pour cela, les nœuds insaturés sont temporairement supprimés du graphe. Le graphe devient un graphe restreint.

Décomposition du graphe en PPSI

La détermination des composantes irréductibles consiste à rechercher dans le graphe restreint celles fortement connexes (Cf. annexe C.6.2). Elles forment donc des sous-graphes composés de $2n$ nœuds (n variables pour n équations) définissant les Plus Petits Systèmes Indépendants (PPSI) structurellement non-dégénérés.

Synthèse de l'analyse structurelle

L'analyse structurelle a permis dans un premier temps d'obtenir les informations nécessaires à la construction d'un diagnostic de pré-résolution. En effet, la recherche des composantes semi-irréductibles a permis de localiser les zones :

- sur-contraintes (Figure 10.2-a) ;
- bien-contraintes (Figure 10.2-b) ;
- sous-contraintes (Figure 10.2-c).

Dans un deuxième temps, elle a permis de construire un plan de résolution à partir du graphe restreint. Or, ce graphe est un sous-graphe du graphe global. Le plan de résolution est donc partiel et structurellement bien contraint. Effectivement, comme les nœuds non-saturés par le couplage ont été supprimés, le plan associé est composé de PPSI structurellement bien contraints.

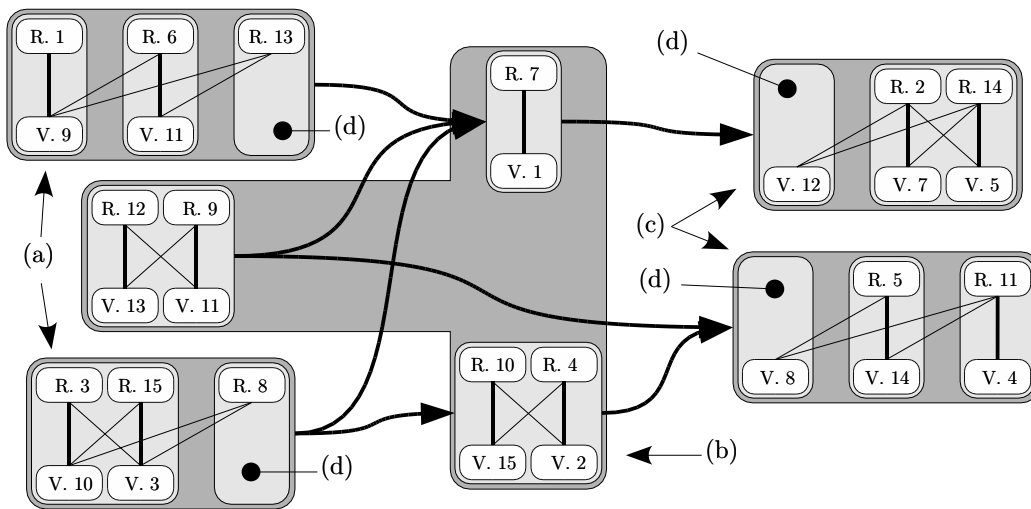


FIG. 10.2: Plan de résolution

Ce plan de résolution est ensuite complété avec les nœuds non-saturés. Chaque nœud apparaît, dans le plan, isolé dans un PPSI (Figure 10.2-d) qui sera appelé soit *PPSI sous-contraignant* lorsque le nœud est une variable, soit *PPSI sur-contraignant*.

10.2.2 Les PPSI sur-contraignants

Afin de résoudre un problème sur-contraint, le principe retenu est de rendre le problème bien contraint et de vérifier que la solution obtenue soit solution du problème global. Cette approche est viable. En effet, l'ensemble des solutions d'un problème sur-contraint est un sous-ensemble de celui du problème bien-contraint.

En pratique, le plan de résolution d'un problème sur-contraint contient des PPSI sur-contraignants (Figure 10.3-a : PPSI S-C.). Ces PPSI ne sont pas utilisés par les algorithmes de résolution, mais doivent être vérifiés par l'état de résolution. Ils interdisent donc la formation de certaines branches de l'arbre de solutions. Par exemple, le PPSI sur-contraignant de la figure 10.3-a refuse deux solutions du PPSI 2, ce qui supprime les deux branches de solutions grisées (Figure 10.3-b). Ces deux branches auraient donné six autres solutions au problème.

La différence entre des problèmes sur-contraints consistants et inconsistants apparaît alors dans l'existence ou non de solutions. En effet, un problème sur-contraint consistant contient trop de spécifications mais, contrairement aux problèmes inconsistants, cet ensemble reste cohérent. Par exemple, un triangle spécifié par trois angles et une longueur est un problème sur-contraint. Or, si la somme des trois angles est égale à π , il n'y a qu'une redondance d'information et il existe une solution. Par contre, si cette somme est égale à $(\pi + \epsilon)$, avec $\epsilon \neq 0$, le problème devient incohérent. Il n'existe alors aucune solution.

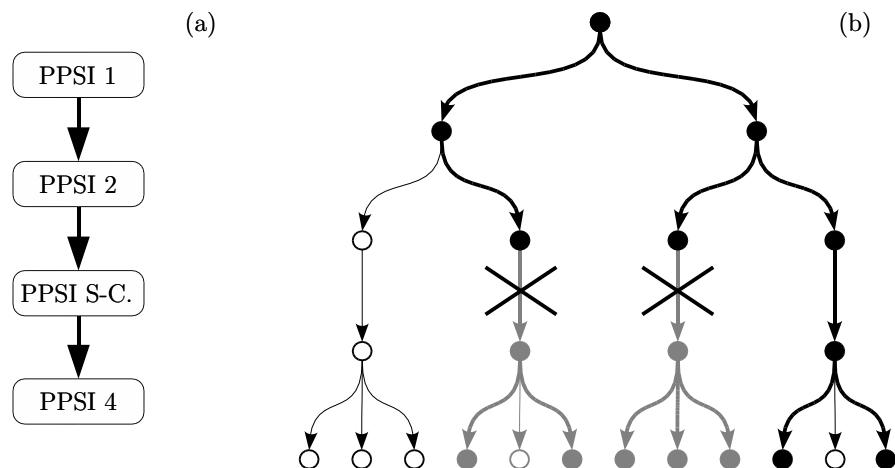


FIG. 10.3: (a) un plan de résolution d'un problème sur-contraint, (b) l'arbre de solutions associé.

Les inéquations

Les spécifications généralement utilisées s'expriment par des égalités entre paramètres. Or, l'utilisateur peut souhaiter restreindre les valeurs admissibles de certains paramètres. Sur cette idée, Serré [Serré 2000] a proposé un ensemble de contraintes géométriques, appelées contraintes de chiralité et contraintes d'appartenance à une zone, répondant à ce besoin. Ces contraintes s'expriment facilement par des inéquations et les inéquations permettent d'élargir le champ d'action de ce nouveau type de contrainte.

Une inégalité peut être vue comme une relation qui sur-contraint un problème. En effet, par définition, elle limite l'espace de définition des **variables**. Tout comme une relation sur-contrainante, elle doit donc être vérifiée par l'état de résolution. De plus, leurs gestions dans le plan de résolution sont identiques. Pour cela, dans la suite du document, plus aucune différence ne sera faite entre ces deux familles d'équations. En contre partie, afin de pouvoir les utiliser, il est nécessaire de les distinguer, au niveau du graphe biparti, des équations classiques. Cette distinction est réalisée par la pondération des nœuds.

Les équations supplémentaires

Une nouvelle famille de spécifications peut aussi être proposée. En effet, il y a les spécifications strictes traduites par des équations dont l'égalité doit être vérifiée et les contraintes de restriction réduisant le domaine de solutions qui sont exprimées sous la forme d'inéquations. On peut aussi proposer d'ajouter des relations qu'il serait souhaitable de vérifier. Ces relations, lorsqu'elles sont exprimées par des équations, peuvent alors aider à la résolution d'un problème localement sous contraint. En revanche, elles doivent faire partie en priorité des

équations sur-contraignantes. De plus, afin d'assurer une solution au problème, ces relations peuvent ne pas être vérifiées.

Il est en plus imaginable de donner la possibilité à l'utilisateur de hiérarchiser ses contraintes supplémentaires.

Pondération des nœuds *équations*

Afin d'autoriser l'utilisation d'inéquations et d'équations supplémentaires, celles-ci doivent être distinguées des équations prescrites à travers les spécifications de base de l'utilisateur. De plus, nous verrons que l'analyse qualitative peut identifier certaines équations comme sur-contraignantes par rapport à d'autres. Comme les traitements sont dynamiques, ces équations redondantes doivent être connues tout au long du processus.

Pour ces premières raisons, les nœuds *équations* sont pondérés afin de les différencier pendant les différents traitements, comme particulièrement pendant la recherche d'un couplage maximal. Une hiérarchie est donc proposée. Son premier rôle est de préférer la saturation de certains nœuds. Dans l'ordre décroissant des poids, nous avons donc :

- les nœuds *équations* de base ;
- les nœuds *équations* sur-contraignants, c'est-à-dire ceux qui apparaissent à un moment donné comme redondants pour le système. En effet, l'évolution du problème peut faire disparaître une sur-contraainte locale. Le graphe, pour redevenir localement bien contraint, doit alors saturer ces nœuds ;
- les nœuds *équations* supplémentaires avec les différents niveaux de hiérarchie proposés par l'utilisateur ;
- les nœuds *inéquations* qui ne doivent en aucun cas être saturés. En effet, une inéquation ne peut pas faire partie d'un PPSI puisque celui-ci serait alors structurellement bien contraint mais qualitativement sous-contraint.

10.2.3 Les PPSI sous-contraignants

Une composante semi-irréductible sous-contrainte est toujours localisée à l'extrémité d'une branche du plan de résolution (Figure 10.2). La raison est structurelle et est présentée en annexe C.6.2. Ceci facilite la résolution d'un tel problème. Quoi qu'il en soit, le problème peut être globalement sous-contraint et différentes stratégies de résolutions doivent pouvoir être mises en place.

Pour résoudre un problème sous-contraint, la première idée qui vient à l'esprit est d'utiliser un algorithme minimisant une fonctionnelle. Classiquement, c'est la variation des paramètres qui est minimisée. En pratique, cette approche est la moins appréciée des utilisateurs car la solution obtenue avec un tel critère est généralement modifiée plus globalement, alors que l'utilisateur préfère des

modifications plutôt locales et beaucoup plus intuitives. L'idéal est donc de ne modifier qu'un minimum de paramètres, les autres gardant leur valeur initiale.

En pratique, les paramètres inchangés sont les variables dont les nœuds du graphe n'ont pas été saturés. Le problème est alors arbitrairement rendu bien-contraint, mais cette approche n'est pas très satisfaisante pour différentes raisons :

- le choix des variables imposées est aléatoire ;
- les choix peuvent ne pas convenir à l'utilisateur ;
- le problème peut ne plus accepter de solution. En effet, la valeur des paramètres inchangés peut être la cause d'un problème d'existence de solutions (Cf. sous-section 11.3.2).

Pondération des nœuds *variables*

Afin de réaliser une sélection pertinente de *variables* à saturer, les nœuds sont pondérés afin de favoriser certains choix. Plusieurs stratégies de pondérations peuvent alors être proposées.

La première, déjà partiellement introduite, est dirigée par l'utilisateur à l'aide des contraintes de liberté (Cf. section 7.6). Ces contraintes lui permettent d'informer le système sur ses préférences. En pratique, les nœuds des **variables** associées à ces contraintes vont recevoir un poids favorisant leur saturation. La hiérarchie des nœuds *variables* est alors dirigée par les nœuds des **variables sources** :

- une **variable source** est un paramètre du problème, sa valeur est imposée. Son nœud doit donc être ignoré pendant l'analyse. Le poids de son nœud est nul ;
- une **variable libre** est une **variable** qui a été choisie par l'utilisateur pour voir sa valeur modifiée. Le poids de son nœud est important car il doit en priorité être saturé ;
- une **variable puits** est un paramètre dont la valeur n'est pas imposée mais généralement implicitement proposée par l'utilisateur à l'aide de l'esquisse initiale. Le poids de son nœud est donc moyen.

Cette première hiérarchie peut être précisée afin d'offrir des comportements particuliers. Par exemple, les **variables** géométriques sont essentiellement de deux natures différentes : les **variables** angulaires et les **variables** dimensionnelles. Privilégier la saturation d'une famille provoque des comportements particuliers. Effectivement, privilégier la saturation des dimensions favorise la conservation de l'aspect local de la géométrie. La géométrie finale subit fréquemment une dilatation locale suivant un axe. La figure 10.4-a présente une géométrie sous-contrainte et la figure 10.4-b présente un résultat pour ce comportement. Favoriser la saturation des variables angulaires permet d'obtenir une conservation locale des dimensions (Figure 10.4-c). Ce comportement est considéré comme

plus intuitif par les utilisateurs.

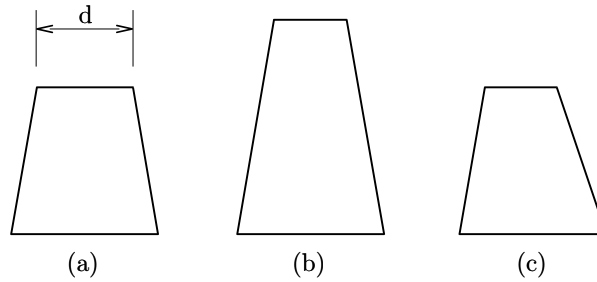


FIG. 10.4: Différents comportements de résolution d'un problème sous-contraint

Analyse d'un problème sous-contraint

Cette stratégie de résolution n'assure malheureusement pas l'obtention d'une solution. En effet, le problème est artificiellement bien contraint en imposant la valeur des variables puits non saturées. La figure 10.5-a illustre le plan de résolution d'un tel problème. La composante sous-contrainte est décomposée en deux PPSI bien-contraints et une variable non saturée dont la valeur appartient en théorie aux réels. L'image de cette variable par le premier PPSI lorsqu'elle prend successivement toutes les valeurs de \mathbb{R} est alors un sous ensemble de \mathbb{R} . La figure 10.5-b illustre l'arbre de solutions de ce problème. Les résultats du premier PPSI sont représentés par les deux ensembles. Or, seule une partie de ces ensembles peut admettre des solutions pour le second PPSI. Celle-ci est représentée en noir sur la figure et son image par le second PPSI est alors définie par plusieurs ensembles.

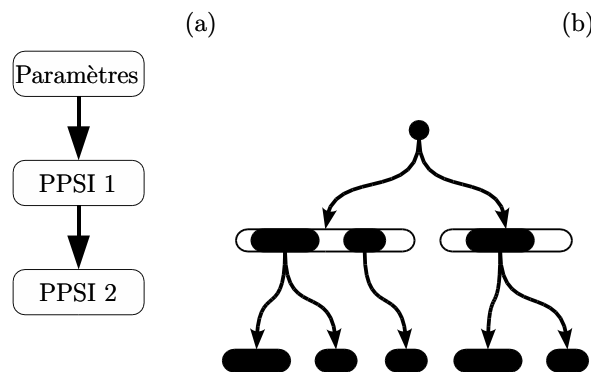


FIG. 10.5: Plan de résolution d'un système sous-contraint et arbre de solutions

Pour ces raisons, trouver une solution à une composante sous-contrainte tout en offrant un certain comportement est aléatoire. Malgré tout, cette stratégie est préférée pour plusieurs raisons. D'une part, elle est rapide car elle est basée sur une décomposition en PPSI de la composante sous-contrainte. De plus, les modifications demandées par l'utilisateur sont généralement de faible amplitude

et les solutions “proches” de l’esquisse initiale. Cependant, le risque d’échec de la résolution reste probable. Dans ce cas, un algorithme d’optimisation permettra de trouver une solution. Bien entendu, cet algorithme ne sera lancé que sur la composante sous-contrainte.

10.2.4 Conclusion

L’analyse structurelle a permis d’obtenir dans un premier temps la décomposition du problème en composantes sur-contraintes, bien-contraintes, et sous-contraintes. Pour obtenir un découpage plus précis, elle est ensuite transformée temporairement en une forme bien-contrainte. Malgré tout, l’analyse réalisée n’est que structurelle. Rien n’assure aux équations des différents PPSI d’être fonctionnellement indépendantes. Pour cela, la seconde étape de la préparation du plan de résolution est une analyse qualitative complétée par la correction du plan de résolution.

10.3 Analyse qualitative

10.3.1 Introduction

L’analyse qualitative complète l’analyse structurelle afin de vérifier et éventuellement, de corriger le plan résolution. En effet, l’analyse structurelle a proposé un découpage du système global en sous-systèmes structurellement biens contraints, c’est-à-dire que chaque PPSI est composé de n équations dont l’espace de définition minimum est de dimension n . Il est donc primordial de vérifier que ce découpage définit réellement des sous-systèmes n’admettant qu’un ensemble dénombrable de solutions.

L’analyse est réalisée individuellement pour chaque PPSI. Le problème est qu’elle n’est, en théorie, pas réalisable tant que le PPSI n’est pas parfaitement défini. En effet, les équations de chaque PPSI ne sont connues qu’à des paramètres près, qui sont soit des variables sources dont la valeur est choisie par l’utilisateur, soit des variables puits dont la valeur est solution des précédents PPSI. Par exemple, un PPSI peut être composé du système d’équations linéaires suivant :

$$\begin{cases} a_1x + b_1y + c_1z = d_1 \\ a_2x + b_2y + c_2z = d_2 \\ a_3x + b_3y + c_3z = d_3 \end{cases}, \quad (10.1)$$

avec a_i , b_i , c_i et d_i les paramètres et x , y , z les inconnues du problème. Pour ce PPSI, tant que les paramètres ne sont pas connus, il n’est pas possible de calculer le déterminant du jacobien afin de vérifier que les équations sont bien linéairement indépendantes.

L'analyse qualitative est tout de même réalisée avant la résolution, c'est-à-dire avant de connaître réellement chaque PPSI. Ce choix est motivé par un ensemble d'hypothèses qui est présenté à la sous-section suivante. Ensuite, les différents points permettant de qualifier l'état de contrainte d'un PPSI sont présentés et justifiés. Pour finir, les actions entreprises afin de corriger le plan de résolution sont présentées.

10.3.2 Hypothèses autorisant l'analyse

L'analyse qualitative est basée sur un ensemble d'hypothèses réduisant l'ensemble des systèmes résolubles par cette approche, mais elles sont peu restrictives pour que cette analyse puisse traiter une grande partie des problèmes de conception mécanique.

Les relations algébriques sont de classe C^1 . La première hypothèse est une hypothèse pratique. Le système d'équations est supposé être de classe C^1 (voir C^2) sur l'ensemble du domaine des variables intervenant dans le processus de résolution. La raison est simple, les algorithmes de résolution utilisés sont généralement basés sur une linéarisation du système d'équations, comme par exemple l'algorithme de Newton - Raphson. Si les fonctions ne sont pas de classe C^1 , ces algorithmes sont inutilisables.

Le problème n'est pas dégénéré. En effet, nous avons vu qu'un système de n équations fonctionnellement indépendantes dans un espace de dimension n admet un ensemble dénombrable de solutions, ce qui n'empêche pas à des systèmes de m équations dans un espace de dimension n avec $m < n$ d'avoir aussi un nombre dénombrable de solutions. Par exemple,

$$x^2 + y^2 = 0,$$

admet une solution unique qui est $x = 0, y = 0$. Or, ce système est dégénéré. En effet, si l'on perturbe la relation, l'ensemble des solutions peut devenir continu. Par exemple quelque soit $\epsilon \in \mathbb{R}^{+*}$,

$$x^2 + y^2 = 0 + \epsilon,$$

admet un ensemble continu de solutions. Ainsi, la recherche de solutions est difficile avec les algorithmes classiques. De plus, rien n'assure, qu'en raison des problèmes dus aux erreurs de troncature des ordinateurs, que la relation initiale ne devienne pas une relation perturbée.

Les valeurs spécifiées n'entraînent pas la dégénérescence du le problème. En effet, la spécification de valeurs critiques pour certaines contraintes peut dégénérer le problème, comme par exemple, une contrainte de distance nulle. En effet, plaçons nous dans un repère cartésien. Prenons un point \mathbf{O} connu de coordonnées (x_O, y_O) . Plaçons dans l'espace un point \mathbf{A} distant de 0 unité du point \mathbf{O} . En théorie, ce problème possède une solution puisque les deux points sont coïncidents alors qu'en pratique une seule équation a pu être écrite pour deux inconnues :

$$(x_O - x_A)^2 + (y_O - y_A)^2 = 0.$$

Le problème est alors structurellement sous-contraint bien qu'il existe une seule solution. Si l'on ajoute une autre contrainte, il deviendra structurellement et qualitativement bien-contraint alors qu'il est effectivement sur-contraint.

L'état initial n'est pas dégénéré. Les raisons sont de deux ordres. La première est que les valeurs initiales sont basées sur l'état initial. Si celui-ci est dégénéré, le risque existe que certains paramètres ne puissent pas être initialisés. Par exemple, un bipoint de mesure nulle empêche l'initialisation des angles liés à son vecteur support. La seconde raison est que l'état initial est utilisé pour réaliser un diagnostic pré-résolution sur l'état de contrainte du système. Si celui-ci est dégénéré, le jacobien du système peut devenir nul et provoquer des corrections erronées alors qu'à la solution, le problème ne l'est pas.

L'utilisateur peut demander la résolution d'un problème localement sous-contraint et/ou sur-contraint. Pour les zones sur-contraintes, des sous-systèmes composés de relations redondantes peuvent apparaître puisqu'elles sont structurellement bien contraintes. Les différentes hypothèses exposées ont permis de définir différentes analyses réalisées avant la résolution globale pour repérer et corriger de tels PPSI.

10.3.3 Analyse qualitative

Le système d'équations est supposé respecter les différentes hypothèses énoncées ci-dessus. Une analyse préalable à la résolution est réalisée sur chaque PPSI. L'analyse structurelle a défini un ensemble de PPSI tels que chaque PPSI soit un système d'équations composé de m fonctions ϕ_1, \dots, ϕ_m dans un espace vectoriel E de dimension finie $\dim(E) = m$. Cette analyse contrôle que chaque PPSI forme un système non-dégénéré d'équations de V défini dans une partie de E . La codimension d'un PPSI est au plus égale à m . Or, il faut qu'elle soit égale à m . Pour contrôler cela, il convient de vérifier (Cf. annexe E) :

- qu'il existe un ouvert U de E dans lequel les ϕ_i sont définies, qu'elles forment une sous variété V de E telle que $V \cap U$ soit l'ensemble des points $x \in U$

avec $\phi_1(x) = 0, \dots, \phi_m(x) = 0$;

- pour tout $x \in U$ les formes linéaires $d\phi_i(x)$ sont linéairement indépendantes, sauf dans un ensemble de mesure nul et pour tout $x \in V$.

Le premier point est direct. En effet, il suffit de choisir U comme le domaine de définition des ϕ_i . Comme V est défini par le système d'équations ϕ_i , alors V est l'ensemble des points $x \in U$ tel que $\phi_1(x) = 0, \dots, \phi_m(x) = 0$.

Le second point est plus délicat. En effet, il faut vérifier que la matrice jacobienne de ϕ_1, \dots, ϕ_m soit de rang maximal pour tout $x \in V$ et presque partout ailleurs. Or, comme E est de dimension m et que l'on veut vérifier que les ϕ sont de codimension m , ceci est équivalent à vérifier que le déterminant du jacobien des ϕ_i est non nul presque partout. En d'autres termes, la condition précédente exprime que le déterminant ne doit pas être identiquement nul sur U . D'un point de vue pratique, le calcul symbolique du déterminant jacobien est hors de portée puisqu'il a un très grand nombre de termes.

La solution choisie pour vérifier que le rang de la matrice jacobienne de chaque PPSI est maximal est donc basée sur l'approche probabiliste proposée par Lamure [Lamure *et al.* 1998]. Le déterminant du jacobien est calculé sur un échantillon de points de U obtenu aléatoirement. Si pour chaque point, le déterminant est nul, il y a une probabilité très proche de 1 que le déterminant soit identiquement nul. Dans les autres cas, les formes linéaires $d\phi_i(x)$ sont probablement linéairement indépendantes presque partout dans U . Il reste maintenant à vérifier qu'elles le sont sur V . Or, pour le problème initial, seul un point de V est connu. En raison de l'hypothèse de non dégénérescence de l'état initial, si en ce point le déterminant du jacobien est non nul, nous pouvons supposer qu'il en sera de même partout ailleurs sur V . Ceci permet de dire, en raison de l'hypothèse que les valeurs spécifiées ne dégèrent pas le problème, qu'il ne le sera pas à l'état final. Une vérification de cette configuration devra être effectuée à l'état final avant d'accepter les solutions.

10.3.4 Les modifications du plan de résolution

L'analyse qualitative a permis d'identifier les PPSI dont le système d'équations est dégénéré. Pour de tels PPSI, un ensemble de relations entraînant la dégénérescence du le problème est identifié pendant la triangularisation de la matrice du jacobien. Ces relations sont supprimées temporairement du graphe et leurs nœuds deviennent des nœuds "équations" sur-contrainants. Or supprimer une sur-contrainte permet de rendre le problème mieux contraint, c'est-à-dire que localement le problème redevient bien contraint. Dans le graphe biparti associé, il doit donc exister des nœuds "équations" dont la saturation permettrait d'obtenir localement des PPSI non-dégénérés.

Pour des raisons de performances, il est ici aussi souhaitable de ne pas refaire

l'analyse structurelle sur le graphe complet mais uniquement sur le plus petit sous-graphe possible. La section suivante présente donc l'ensemble des algorithmes permettant une gestion dynamique du plan de résolution.

10.4 Gestion dynamique du plan de résolution

Le plan de résolution qui représente une description complète des spécifications de l'utilisateur est amené à évoluer tout au long de la définition d'un objet. En effet, toute modification apportée au modèle de spécifications se répercute généralement² sur le plan de résolution. Or, la mise à jour de ce plan est dépendante de celle du graphe biparti associé. La problématique est dans un premier temps de savoir comment analyser un graphe après qu'il ait subi des modifications sans avoir à le traiter globalement. En d'autres termes, le problème est d'identifier le plus petit sous-graphe contenant les modifications. Dans un deuxième temps, il faut être en mesure de mettre à jour localement le plan de résolution.

10.4.1 Gestion dynamique d'un graphe biparti

Soit un graphe biparti $G = [S \cup T, U]$. Soit la décomposition canonique de ce graphe définie par les couples d'ensembles de sommets $\{X_i \cup W_i\}$, $\{S_i \cup T_i\}$, $\{U_i \cup V_i\}$. Le graphe est décomposé en sous graphes H_i , G_i , L_i .

Le problème est d'identifier le plus petit sous-graphe contenant les modifications apportées dans le graphe biparti. Ces modifications se résument à la suppression ou l'ajout d'un nœud et de son cocycle. Pour cela, les travaux de Dulmage et Mendelsohn sur la stabilité d'un graphe biparti [Dulmage *et al.* 1959] permettent de répondre en partie à cette problématique. Leurs travaux portent sur l'évolution de sa structure suite à la suppression et à l'ajout d'un ensemble d'arêtes. Leurs résultats sont résumés par les neuf propriétés exposées à la sous section C.6.3 de l'annexe C.

À partir de ces propriétés, les algorithmes permettant de gérer dynamiquement la structure d'un graphe biparti ont pu être définis. En effet, dans l'approche présentée dans ce document, le graphe évolue par la suppression ou l'ajout d'un ensemble de nœuds accompagné de son cocycle.

Ces algorithmes sont basés sur les propriétés supplémentaires propres à la suppression ou l'ajout d'un nœud. Elles sont proposées ci-dessous :

Addition d'un nœud Soit un nœud t et son cocycle $\omega(t) = \{u_i^t = (s_i, t), s_i \in S\}$ ajouté à $G = [S \cup T, U]$. Soit H_i , G_i , L_i la

²La modification de la valeur d'un paramètre d'une spécification est un des rares cas qui n'entraîne pas la modification du plan de résolution.

décomposition canonique de G . Soit G° le graphe induit de G par la décomposition canonique :

- si $s_0 \in U_k$, alors t et u_0^t sont ajoutés respectivement à V_k et H_k . La décomposition canonique de G ne change pas ;
- si $s_0 \in S_k$, alors t et u_0^t sont ajoutés respectivement à T_k et G_k . G_k devient une composante semi-irréductible de type H_k . Soit A° l'arbre inférieur de racine S_k dans G° . Soit $\Upsilon = \{\cup(P \times Q), \forall (P, Q) \in A^\circ\}$. Alors le sous-graphe fondamental composé des arêtes de $U \cap [\Upsilon \cup u_0^t]$ est une nouvelle composante semi-irréductible minimale de type H_k de G remplaçant les composantes contenues dans $(P \times Q)$ et G_k ;
- si $s_0 \in X_k$, alors la dimension extérieure de X_k augmente d'une unité. Il suffit de décomposer X_k en composantes de type G_i et L_i . Elles remplaceront X_k dans la décomposition canonique de G . Le reste de la structure ne sera pas affecté.

Ensuite, il suffit d'ajouter les autres arêtes u_i^t en utilisant les propriétés de Dulmage et Mendelsohn.

Pour ajouter un nœud s et son cocycle il suffit d'inverser le rôle des ensembles S et T .

Suppression d'un nœud Soit un nœud x et son cocycle $\omega(x)$ à supprimer de G . Il suffit de supprimer le cocycle $\omega(x)$ de G en utilisant les propriétés de Dulmage et Mendelsohn. Le nœud x devient alors une composante connexe de G qu'il suffit de supprimer.

10.4.2 Gestion du plan de résolution

Le plan de résolution est défini sous la forme d'un graphe orienté. Ce graphe ordonne les PPSI en fonction de leurs dépendances de résolution.

Rappelons qu'un PPSI est un sous-système d'équations indépendantes, c'est-à-dire qu'il peut être résolu seul à partir du moment où toutes ses équations sont parfaitement définies. En d'autres termes, une équation appartenant à un PPSI relie un ensemble de variables. Dans cet ensemble, seul un sous-ensemble est déterminé par ce PPSI. Les autres sont déterminés par les PPSI précédents dans le plan de résolution. Ils sont appelés paramètres.

Les équations d'un PPSI sont donc parfaitement définies lorsque la valeur des différents paramètres est connue. La modification du modèle de spécifications entraîne donc généralement une modification locale du plan de résolution. Un ensemble de PPSI est supprimé, alors qu'un autre ensemble est ajouté. Or, la modification des éléments du plan est restreinte au sous graphe composé des PPSI supprimés. En effet, la suppression d'un ensemble de PPSI entraîne la suppression des modules de résolution des paramètres de certains PPSI. Les PPSI ne sont pas concernés par les modifications :

- s'ils se trouvent avant les PPSI supprimés car ils restent résolubles ;
- s'ils se trouvent après, ils ne peuvent pas être résolus tant que leurs paramètres ne sont pas connus. Or, l'ensemble des nouveaux PPSI compose obligatoirement les nouveaux modules nécessaires à la détermination de ces paramètres.

Finalement, la gestion dynamique du plan de résolution consiste donc simplement à supprimer les PPSI obsolètes et à les remplacer par les nouveaux PPSI.

10.5 Conclusion

Ce chapitre a permis de présenter le concept du plan de résolution. Cet ensemble d'informations est propre au module de résolution. En effet, un module par principe est remplaçable. Or, le plan de résolution ne regroupe que les informations nécessaires à la stratégie de résolution de ce module. Il ne fait donc pas partie du *modèle dynamique de spécifications d'ingénierie*.

Ce plan de résolution n'est autre que le résultat d'une analyse structurelle. Les différents algorithmes utilisés ont été présentés suivis de propositions sur les résultats obtenus qui ont abouties à la définition de hiérarchies des paramètres et des relations. Ces différentes hiérarchies permettent d'offrir à l'utilisateur la possibilité d'enrichir son modèle dynamique par des contraintes facultatives ou des paramètres portant sur le comportement de la résolution.

Cette première analyse possède des limites. Pour combler cela, une deuxième analyse est utilisée. Elle a été présentée dans la troisième section de ce chapitre.

Finalement, la présentation des différents algorithmes nécessaires a une gestion dynamique du plan de résolution a clôturé ce chapitre. Maintenant que le plan de résolution est défini, la résolution proprement dite peut être menée. La démarche utilisée pour cette étape va donc être proposée au chapitre suivant.

Chapitre 11

La résolution

L'organisation de la résolution est définie. Il faut maintenant résoudre le problème. Cette tâche est prise en charge par un module. Ce chapitre montrera que le rôle de ce module n'est finalement que de diriger un ensemble d'agents de résolution.

11.1 Algorithme de résolution

L'algorithme de résolution a deux niveaux de granularité. En effet, nous avons vu que l'analyse structurelle du problème décompose le système initial en trois groupes de composantes : les composantes sur-contraintes, la composante bien-contrainte et les composantes sous-contraintes. Ces composantes sont elles-mêmes divisées en PPSI. Ainsi, la résolution comporte trois étapes majeures, toutes gérées par un meta-agent. Le premier traite les composantes sur-contraintes en parallèle, le second résout la composante bien-contrainte et le dernier termine la résolution. Ils sont qualifiés de meta-agents en raison de leur tâche s'apparentant essentiellement à un rôle de manager. Ces agents sont basés sur le même algorithme principal, présenté ci-dessous.

L'algorithme principal de résolution est relativement simple (Figure 11.1) et peut se résumer en une phrase : “tant qu'il existe des PPSI non-résolus, les résoudre”. Le rôle de ces meta-agents consiste alors principalement à diriger un ensemble d'agents de résolution afin d'obtenir un résultat.

En effet, comme il n'existe pas de solveurs génériques “robustes” et que chaque famille de PPSI a des particularités, différents agents sont proposés afin de mener au mieux la résolution.

Malgré tout, le meilleur ensemble d'agents n'est pas suffisant pour trouver une solution. En effet, chaque agent sait résoudre une certaine gamme de problèmes mais peut bloquer sur certains cas. Il faut donc être en mesure d'analyser les différents échecs. Cette analyse permettra de poursuivre au mieux la recherche

d'une solution ou d'annoncer un diagnostic. Pour cela, elle doit permettre de connaître le nombre de solutions d'un PPSI. Ceci permettrait de différencier, entre autres, l'échec numérique de la non-existence de solutions. Une première question est alors posée : "comment connaître le nombre de solutions à un PPSI donné ?".

Dans les cas où l'échec de résolution n'est pas dû à un problème d'existence de solution, il faut tout mettre en œuvre pour la trouver. Mais là, une nouvelle problématique apparaît : "comment assurer la convergence de la résolution ?".

S'il n'y a pas de solution, il suffit de changer de branche dans l'arbre de solutions (Cf. section 9.4). Pour cela, il faut retourner à un PPSI admettant plusieurs solutions, en trouver une autre et reprendre la procédure principale. Le dernier point délicat de la résolution vient d'être posé : "comment trouver différentes solutions à un PPSI ?".

Afin de répondre à toutes ces questions, ce chapitre est découpé en cinq sections :

- la seconde section introduit les différents agents de résolution. Nous verrons que les premiers sont élémentaires alors que d'autres peuvent faire plus que simplement résoudre ;
- la troisième section contient la présentation détaillée des points clés de l'algorithme avec, entre autres, le détail des différents critères de choix des agents de résolution ;
- la quatrième section expose les particularités des meta-agents de résolution des composantes sous- et sur-contraintes ;
- le cinquième section devra conclure sur ce module avant d'introduire le dernier chapitre de cette partie.

11.2 Les agents de résolution

Le module de résolution est destiné à une application ne nécessitant qu'une certaine précision. En revanche, afin qu'elle reste interactive, l'optimisation du temps de résolution est primordial. Il est donc important de mettre en place l'approche de résolution la plus rapide assurant la précision souhaitée. Cette raison a contribué au choix de la stratégie choisie pour la résolution.

Le problème global est décomposé en sous-problèmes indépendants (PPSI) plus facilement et rapidement résolubles. L'optimisation du temps de résolution passe donc par un choix judicieux de l'algorithme de résolution pour chaque PPSI. Ainsi, le module est basé sur une approche multi-agents où chaque agent est spécialisé dans la résolution d'une famille de PPSI.

Pour certains groupes, plusieurs agents peuvent être disponibles. Ils sont de vitesse et de robustesse différentes. L'intérêt est de disposer de différents algo-

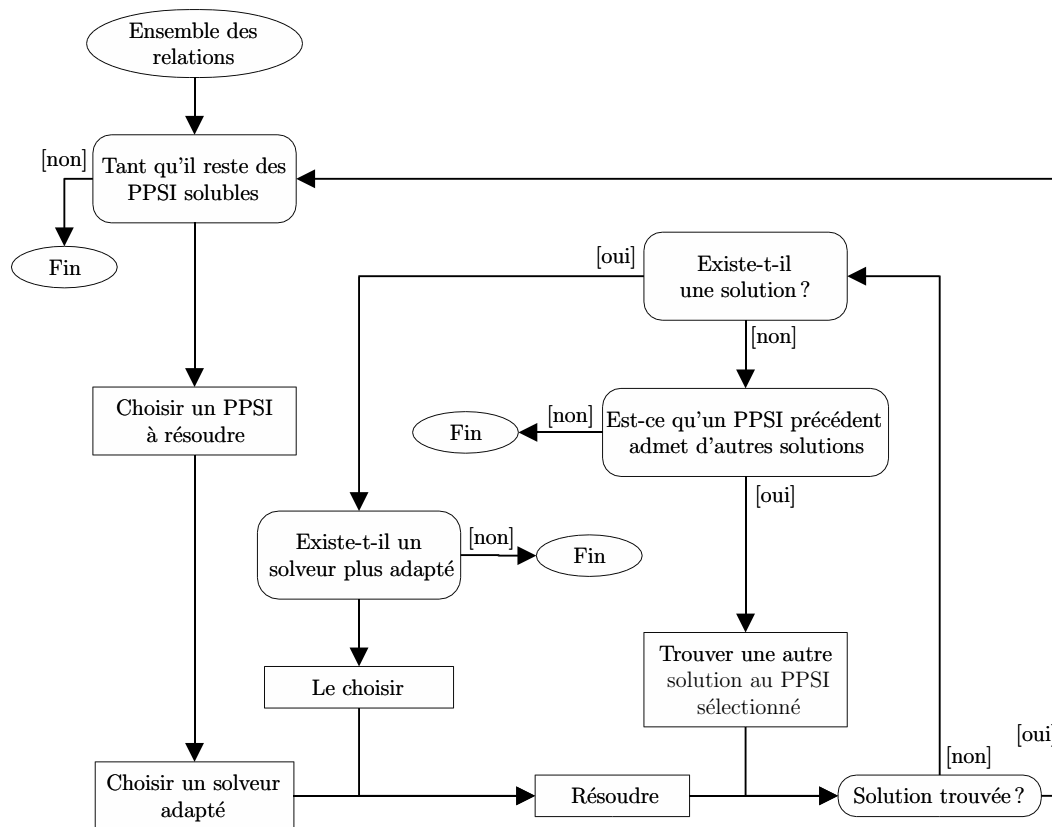


FIG. 11.1: Algorithme de résolution

rithmes pour assurer la convergence de la résolution. Le principe est alors de choisir dans la liste des agents celui qui a le meilleur rapport “vitesse de résolution / robustesse”. En cas d’échec de résolution, d’autres agents plus fiables doivent pouvoir prendre le relais.

Cette section propose donc une première série d’agents qui ne demande qu’à être complétée.

11.2.1 Agents linéaires

Les agents linéaires sont des solveurs destinés à résoudre les PPSI linéaires. La résolution d’un système linéaire pose en théorie peu de difficultés, alors qu’en pratique la résolution numérique de certains cas peut apparaître très instable. Heureusement, ce thème a été très largement étudié et l’on trouve dans la littérature de nombreux algorithmes.

Le choix du meilleur algorithme n’a pas été une priorité. Dans un premier temps, l’agent linéaire est donc simplement basé sur la méthode de Gauss avec une stratégie de pivot partiel afin d’avoir un bon compromis entre robustesse et temps de résolution.

11.2.2 Newton-Raphson

Quelque soit le problème géométrique posé, les PPSI non-linéaires sont présents dans les PPSI à résoudre. L’algorithme de Newton-Raphson a donc été choisi pour devenir le premier agent non linéaire. Malheureusement, cet algorithme diverge trop souvent. D’autre part, il ne propose qu’une solution dont la stabilité est loin d’être acquise (Cf. Figure 3.4).

Malgré son comportement chaotique, sa convergence quadratique en fait un algorithme très performant. De plus, afin de ne pas perdre de temps dans la résolution, un contrôle de la convergence permet de rapidement l’arrêter en cas de divergence et de faire appel à un autre agent plus stable. Ainsi, cet algorithme est très utile pour obtenir rapidement une première solution.

11.2.3 Homotopie

L’homotopie (aussi appelée méthode de continuation ou de cheminement) est une bonne alternative à la méthode de Newton-Raphson. Elle est à la base du second agent non-linéaire. Son principe est de déformer un système d’équations initiales $G(X)$ dont on connaît une solution jusqu’au système cible $F(X)$ et de suivre l’évolution de la solution. La courbe ainsi définie est appelée courbe ou chemin homotopique. Classiquement, la déformation du système est obtenue par une interpolation linéaire, bien que d’autres interpolations soient possibles. La

fonction homotopique H devient alors :

$$H(X, t) = tF(X) + (1 - t)G(X), \quad t \in [0, 1].$$

Dans notre cas, le système d'équations $F(X, P) = 0$ représente un problème géométrique comportant un certain nombre d'inconnues X que l'on cherche à résoudre par un jeu de paramètres P_1 donnés. Or, pour les valeurs initiales des paramètres P_0 , l'esquisse est une solution particulière du système. Le système initial G est donc défini par le système F avec les paramètres initiaux P_0 . Une solution à ce système est donnée par les valeurs X_0 de l'esquisse. Soit $G(X) = F(X, P_0)$ ce système. Le système cible pour lequel on souhaite obtenir une solution n'est autre que $F(X, P_1)$ avec P_1 les valeurs des paramètres imposées par l'utilisateur.

Une autre fonction homotopique, connue sous le nom d'homotopie restreinte, peut, sur le même principe, être construite. Soient $F(X, P_0) = 0$ le système défini par la valeur initiale des paramètres et vérifié par l'esquisse initiale, et $F(X, P_1) = 0$ le système défini par la valeur souhaitée des paramètres, la fonction d'homotopie restreinte est alors :

$$H(X, t) = F(X, tP_1 + (1 - t)P_0), \quad t \in [0, 1].$$

Les méthodes homotopiques sont reconnues pour avoir un comportement *naturel*. En effet, elles convergent généralement vers la solution la plus proche de l'approximation initiale comme l'illustre la figure 3.4 [Lamure *et al.* 1995].

Malgré tout, ces méthodes ne sont pas parfaites et peuvent échouer dans des cas très simples. Il se peut qu'il n'existe pas de chemin homotopique dans le corps des réels entre le point initial et le point final. Un exemple très simple peut être construit à partir du triangle de la figure 11.2. À l'état initial, les paramètres L_1 , L_2 , α ont respectivement pour valeur 160, 70, $\pi/9$. Les valeurs spécifiées pour ces paramètres sont respectivement 70, 110, $\pi/2$. L'homotopie restreinte demande de résoudre progressivement pour t variant de 0 à 1 les triangles dont la valeur des paramètres L_1 , L_2 , α évolue respectivement suivant les lois :

$$160t + 70(1 - t) ;$$

$$70t + 110(1 - t) ;$$

$$\frac{\pi}{9}t + \frac{\pi}{2}(1 - t).$$

Or ces triangles doivent, pour exister, vérifier la relation :

$$L_1 \sin \alpha < L_2.$$

On s'aperçoit qu'elle ne l'est pas en $t = 0,5$. Pour cet exemple, il n'existe donc pas de chemin homotopique entre l'état initial et l'état final. Il n'y a pas non plus

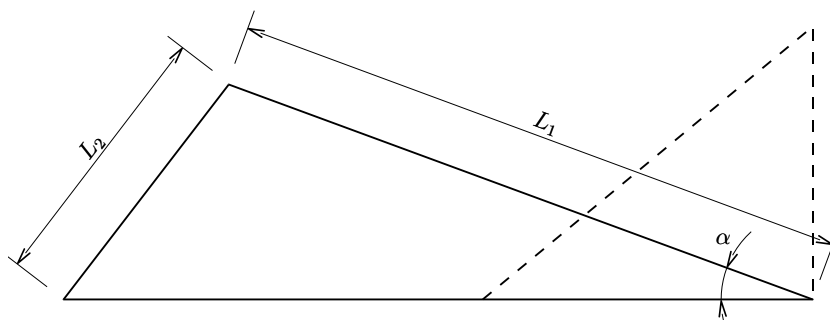


FIG. 11.2: État initial et état final d'un triangle bien contraint ne pouvant pas être résolu par une approche homotopique.

de chemin continu avec la fonction homotopique classique. Par contre, ce même exemple résolu dans le corps des complexes permet de trouver une solution.

La figure 11.3 présente les courbes d'une homotopie linéaire entre les fonctions :

$$f(x) = (x + 0.5) \sin(x - 0.0015) ;$$

$$g(x) = (x - 0.5) \sin(x + 0.0015) .$$

Cet exemple permet d'illustrer que le comportement *naturel* d'une homotopie est parfois contestable. En effet, dans cet exemple, la solution ne converge pas toujours vers la solution la plus proche de l'approximation initiale.

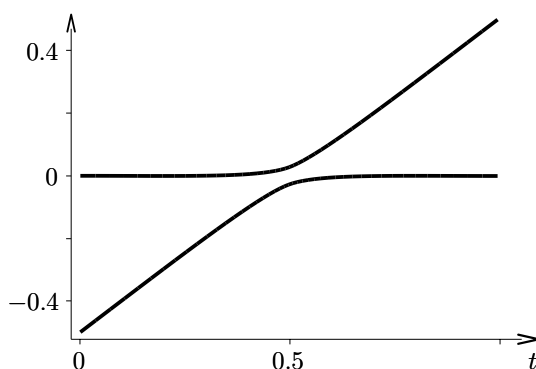


FIG. 11.3: Une courbe homotopique

Malgré tout, cette méthode est très stable contrairement à Newton-Raphson. De plus, le temps de résolution, qui est censé être plus important car elle utilise à plusieurs reprises l'algorithme de Newton-Raphson pour suivre le chemin homotopique, ne l'est pas spécialement. En effet, la convergence de Newton-Raphson peut suivre un parcours chaotique, alors que la méthode de continuation ne fait que recalculer la solution sur les quelques étapes de parcours de la courbe homotopique. Le nombre d'itérations nécessaires est alors généralement restreint. De

plus, en cas d'échec dans le corps des réels, une tentative dans le corps des complexes peut permettre de trouver la solution.

11.2.4 Agent de résolution multi-solutions

Le troisième agent non-linéaire est basé sur le suivi de plusieurs courbes d'une homotopie dans le corps des complexes. L'intérêt de cette approche est que dans certaines conditions, elle permet de trouver toutes les solutions à un problème non-linéaire. En effet, Morgan [Morgan 1987] a montré que pour trouver toutes les solutions d'un système d'équations polynomiales non dégénéré $F(x)$ d'au plus n solutions, il suffit de suivre toutes les courbes homotopiques $H(x, t) = 0$ entre une fonction initiale $G(x)$ et la fonction cible $F(x)$, avec :

- $H(X, t) = t\gamma F(X) + (1 - t)G(X)$, $\gamma \in \mathcal{C}$
- $G(X)$ un système polynômial de n solutions distinctes connues.

En effet, d'après son théorème [Morgan 1987], pour quasiment tous les systèmes G et la quasi-totalité des valeurs de $\gamma \in \mathcal{C}$, l'homotopie H génère une collection de n courbes homotopiques qui ont la particularité d'être continues C^1 et de ne pas se croiser. De plus, il démontre que si une solution de F est de multiplicité m , alors, exactement m courbes homotopiques convergeront sur elle. Dans le cas où le système F a moins de solutions que le système G , alors les courbes qui ne convergent pas sur une racine de F divergent à l'infini.

Le théorème de Morgan est basé sur des systèmes polynômiaux pour la principale raison qu'un majorant des racines est obtenu facilement. Or, ce nombre est nécessaire pour la démonstration du théorème. Ainsi, on peut supposer qu'il est possible de trouver toutes les racines d'un système non-dégénéré quelconque F , à partir du moment où l'on connaît un majorant m du nombre des racines. Dans la pratique, cette approche permet d'obtenir facilement plusieurs solutions pour un système quelconque non-dégénéré.

11.2.5 Solveur algébrique 2D

Les quatre premiers agents sont basés sur des méthodes de résolution numériques mais surtout générales. Or, pour les problèmes de dimension deux, une famille de PPSI non-linéaires est très courante. Cette famille est évidemment associée aux différentes étapes élémentaires d'assemblage. Un problème bien contraint doit effectivement se décomposer en étapes d'assemblage élémentaires. Ces étapes doivent donc apparaître dans les différents PPSI. Les PPSI classiques sont alors de la forme :

$$\begin{cases} a \cos \alpha + b \cos \beta + c_1 = 0 \\ a \sin \alpha + b \sin \beta + c_2 = 0 \end{cases} \quad (11.1)$$

dont les deux inconnues sont à choisir parmi a , α , b et β . c_1 et c_2 sont des constantes. Cette famille contient donc quatre cas différents. Pour chacun, une

méthode de résolution est proposée et va permettre de calculer toutes les solutions possibles. De plus, ces cas ont des conditions d'existence qui vont permettre de dire si la solution existe ou non.

Cas 1 : a et b sont inconnus.

– si $(\alpha - \beta) \neq 0 [\pi]$, la solution unique est :

$$\begin{cases} a = \frac{c_1 \sin \beta - c_2 \cos \beta}{\sin(\beta - \alpha)} \\ b = \frac{c_1 \sin \alpha - c_2 \cos \alpha}{\sin(\beta - \alpha)} \end{cases} ; \quad (11.2)$$

- si $\alpha = \beta [\pi] = 0 [\pi]$ et $c_2 = 0$, le problème est dégénéré et admet une infinité de solutions vérifiant $a \pm b \pm c_1 = 0$;
- si $\alpha = \beta [\pi] = \pi/2 [\pi]$ et $c_1 = 0$, le problème est dégénéré et admet une infinité de solutions vérifiant $a \pm b \pm c_2 = 0$;
- si $\alpha = \beta [\pi] = 0 [\pi/2]$, $c_1 \neq 0$ et $c_2 \neq 0$, alors il n'y a pas de solution.

Cas 2 : a et α sont inconnus. α est obtenu facilement par la relation suivante trouvée en faisant le rapport des relations du système (11.7) :

$$\alpha = \arctan \left(\frac{c_2 + b \sin \beta}{c_1 + b \cos \beta} \right). \quad (11.3)$$

Or, la fonction arctan définit un angle modulo π . Deux valeurs sont donc connues pour α . À partir de ces deux valeurs admissibles, on en déduit les valeurs possibles de a par une des deux relations du système (11.7). Par exemple :

$$a = -\frac{b \cos \beta + c_1}{\cos \alpha}. \quad (11.4)$$

Or, pour $\alpha_2 = \alpha + \pi$, on obtient $a_2 = -a$. Ces deux couples de solutions ne définissent donc qu'une seule configuration. Le couple choisi sera de préférence celui dont la valeur de a est positive.

Cas 3 : a et β sont inconnus. À partir des deux relations du système (11.7), l'équation du second degré ci-dessous est obtenue, permettant d'obtenir les valeurs admissibles de a :

$$a^2 + 2a(c_1 \cos \alpha + c_2 \sin \alpha) + c_1^2 + c_2^2 - b^2 = 0 \quad (11.5)$$

Cette équation a des solutions réelles si et seulement si le discriminant δ est positif. Celui-ci est donné par la relation :

$$\delta = b^2 - (c_1 \sin \alpha - c_2 \cos \alpha)^2. \quad (11.6)$$

Les valeurs de a admissibles sont donc :

- si $\delta < 0$: il n'y a pas de solution ;
- si $\delta = 0$: a est donné par la relation :

$$a = -c_1 \cos \alpha - c_2 \sin \alpha.$$

- si $\delta > 0$: les deux valeurs a_1 et a_2 possibles de a sont :

$$a_1 = -c_1 \cos \alpha - c_2 \sin \alpha - \sqrt{\delta},$$

$$a_2 = -c_1 \cos \alpha - c_2 \sin \alpha + \sqrt{\delta}.$$

Pour chaque valeur admissible de a , la valeur associée de β est obtenue modulo 2π à partir des deux relations du système (11.7).

Cas 4 : α et β sont inconnus. Pour résoudre ce cas, on définit c et γ tels que :

$$c = \sqrt{c_1^2 + c_2^2},$$

et γ soit la solution modulo 2π du système :

$$\begin{cases} \cos \gamma = c_1/c \\ \sin \gamma = c_2/c \end{cases} . \quad (11.7)$$

À partir des deux relations du système (11.7), les relations ci-dessous sont obtenues et permettent d'identifier les différentes valeurs de α et de β :

$$\alpha - \gamma = \pm \arccos \left(\frac{b^2 - a^2 - c^2}{2ac} \right) ;$$

$$\beta - \gamma = \pm \arccos \left(\frac{a^2 - b^2 - c^2}{2bc} \right).$$

Les couples de solutions sont les couples vérifiant la relation :

$$a \sin \alpha + b \sin \beta + c \sin \gamma = 0.$$

Différentes conditions d'existence d'une solution peuvent aussi être identifiées, comme par exemple :

$$\frac{b^2 - a^2 - c^2}{2ac} \in [-1, 1] ;$$

$$\frac{a^2 - b^2 - c^2}{2bc} \in [-1, 1].$$

11.2.6 Solveur algébrique 3D

Comme pour les problèmes 2D, les problèmes 3D ont une famille de PPSI très courante. Cette famille est associée à l'étape élémentaire de l'assemblage d'un ensemble de plans vectoriels. Or, nous avons vu que ces étapes sont duales de la résolution d'un polygone sphérique (Cf. section 8.5). Pour ces parties de résolution, les relations associées ne sont pas encore constituées.

Nous avons dit que pour conclure sur l'existence d'un polygone sphérique, connaissant la mesure de chacune de ces entités, il suffisait de le décomposer en un ensemble de triangles sphériques et de vérifier l'existence de chacun. La résolution d'un tel polygone est basée sur les mêmes principes. La suite de cette sous-section va donc commencer par la présentation de la résolution de ces polygones élémentaires. Dans un deuxième temps, la stratégie de résolution des polygones sphériques sera alors proposée. Finalement, les cas plus complexes seront introduits.

Résolution d'un triangle sphérique

Un triangle est constitué de six éléments : 3 angles et 3 côtés (Figure 11.4). Il est parfaitement défini lorsque les six éléments sont connus. Or, un triplet d'éléments suffit à construire un unique triangle sphérique, les trois autres éléments étant déductibles. Il existe six cas différents dont la résolution (c'est-à-dire l'identification des trois autres éléments) va être détaillée. Pour chacun d'eux, les relations permettant d'identifier les trois inconnues sont rappelées. Les équations (8.15) à (8.20) ont été présentées à la sous-section 8.5.1. De plus, dans certains cas, des relations d'existence [Korn *et al.* 1961] seront énoncées :

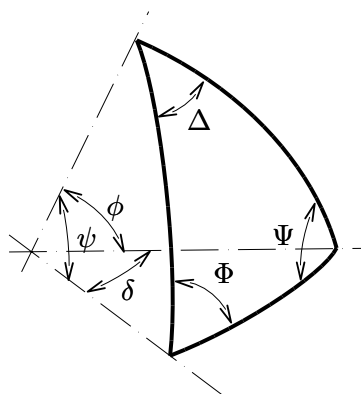


FIG. 11.4: Un triangle sphérique

Trois côtés connus. Les côtés δ , ϕ , ψ sont connus, les angles sphériques Δ , Φ , Ψ se déduisent donc de ces côtés par les relations (8.20). La condition d'existence d'une solution est que la somme de deux côtés soit supérieure à la longueur du troisième.

Trois angles sphériques connus. Les angles sphériques Δ , Φ , Ψ sont connus, les côtés δ , ϕ , ψ se déduisent donc de ces angles par les relations (8.20).

Deux conditions d'existence doivent alors être vérifiées :

- $3\pi > \Delta + \Phi + \Psi > \pi$;
- la somme de deux angles sphériques (par exemple $\Delta + \Phi$) doit être inférieure à la mesure du troisième angle additionnée de π (soit : $(\pi + \Psi)$).

Deux côtés et l'angle sphérique inclus. Les deux côtés ϕ et ψ et l'angle sphérique Δ sont connus. Des relations (8.18), $\frac{\Phi+\Psi}{2}$ et $\frac{\Phi-\Psi}{2}$ sont identifiées. Le côté δ se déduit alors des relations (8.16) ou (8.19) ou (8.20).

Deux angles sphériques et le côté inclus. Les deux angles sphériques Φ et Ψ et le côté δ sont connus. $\frac{\phi+\psi}{2}$ et $\frac{\phi-\psi}{2}$ et l'angle sphérique Δ sont alors obtenus respectivement par les relations (8.18) et par une des relations (8.17) ou (8.19) ou (8.20).

Deux côtés et un angle sphérique opposé. Si les deux côtés ϕ et ψ et l'angle sphérique Φ sont connus, ce problème a une ou deux solutions si $(\sin \psi \sin \Phi) \leq \sin b$. Ψ est alors obtenu à partir de l'équation (8.15). Δ et δ se déduisent alors des relations (8.18). Les valeurs admissibles de Ψ sont alors celles qui vérifient que $(\Delta - \Phi)(\delta - \phi) \geq 0$ et $(\Delta + \Phi - \pi)(\delta + \phi - \pi) \geq 0$.

Deux angles sphériques et un côté opposé. Les deux angles sphériques Φ et Ψ et le côté b sont connus. Le problème admet jusqu'à deux solutions dès que $(\sin \phi \sin \Psi) \leq \sin \Phi$. Comme pour le cas précédent, Ψ est obtenu à partir de l'équation (8.15), Δ et δ se déduisent des relations (8.18). Les conditions d'existence ne changent donc pas : $(\Delta - \Phi)(\delta - \phi) \geq 0$ et $(\Delta + \Phi - \pi)(\delta + \phi - \pi) \geq 0$.

Résolution des polygones sphériques

Par analogie avec les triangles sphériques, un polygone sphérique est constitué de $2n$ éléments : n angles et n côtés. Il est parfaitement défini lorsque les $2n$ éléments sont connus. Or, il suffit de connaître $(2n - 3)$ éléments pour construire un polygone sphérique, les trois derniers éléments étant déductibles. Il n'est pas possible de définir pour chaque famille de polygones les méthodes de résolution puisque le nombre de ces familles est infini. La stratégie de résolution d'un polygone retenue est basée sur une approche procédurale. Les paramètres inconnus d'un polygone sont déduits suite à sa décomposition en triangles sphériques bien contraints.

Connaissant un polygone sphérique, le problème est donc de le décomposer en triangles sphériques résolubles indépendamment les uns des autres. Comme trois paramètres définissent complètement un triangle, il faut donc rechercher dans le polygone les séries de trois paramètres successifs spécifiés. Deux cas sont

possibles : soit deux côtés et l'angle sphérique inclus spécifiés, soit deux angles sphériques et le côté inclus spécifiés. Seul le premier ensemble est intéressant. En effet, il permet de réduire le polygone initial de n côtés en un polygone de $(n - 1)$ côtés (Figure 11.5). Le second (deux angles et le côté inclus) ne possède pas cette propriété. En effet, il ne définit pas un triangle inscrit dans le polygone (Figure 11.5). Le principe est donc de rendre un polygone irréductible par cette méthode. Quant au second, il sera utilisé pour résoudre des cas particuliers.

L'approche de la décomposition d'un polygone en un ensemble de triangles est simple : *tant que le polygone sphérique n'est pas irréductible, rechercher sur le polygone un ensemble de deux côtés consécutifs et de l'angle inclus spécifiés et le remplacer par un nouveau côté.* Un polygone est irréductible dès qu'il ne peut plus être simplifié. Les polygones irréductibles élémentaires sont les triangles. La trigonométrie sphérique permet de résoudre tous les triangles. La question est : *quels sont les autres polygones sphériques irréductibles et comment les résoudre ?* Cette première question pose une nouvelle interrogation : *comment être sûr de connaître tous les polygones irréductibles ?*

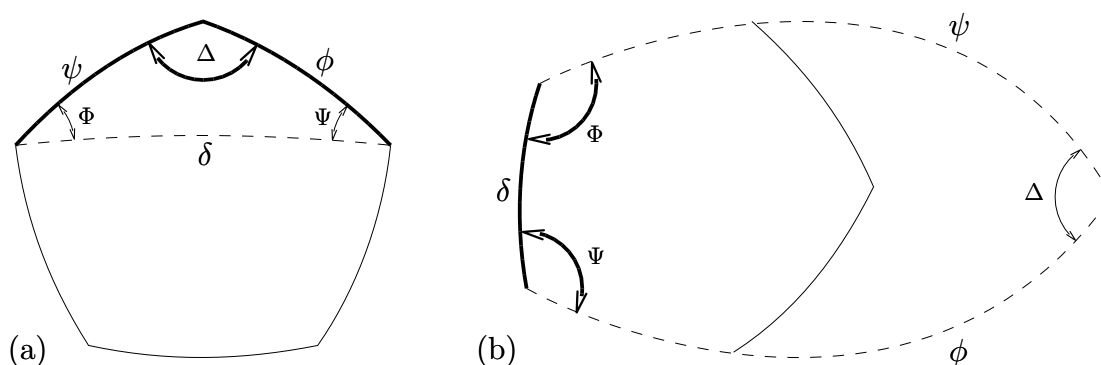


FIG. 11.5: (a) réduction d'un polygone connaissant 2 longueurs successives et l'angle inclus, (b) un triangle sphérique obtenu connaissant 2 angles successifs et le côté inclus ne permet pas de réduire un polygone.

Les polygones sphériques irréductibles

Les polygones sphériques irréductibles sont, d'après l'algorithme de réduction énoncé précédemment, des polygones ne comportant pas deux côtés et l'angle inclus de mesures connues. Ainsi, on peut montrer que le plus grand polygone sphérique irréductible est un hexagone dont trois côtés non adjacents sont les inconnues (Figure 11.6-g). La figure 11.6 regroupe les sept polygones irréductibles autre que les triangles. Ces polygones ont été obtenus en recherchant toutes les combinaisons ne comportant pas deux côtés successifs et l'angle sphérique inclus de mesures connues et tel que le nombre d'inconnues soit trois.

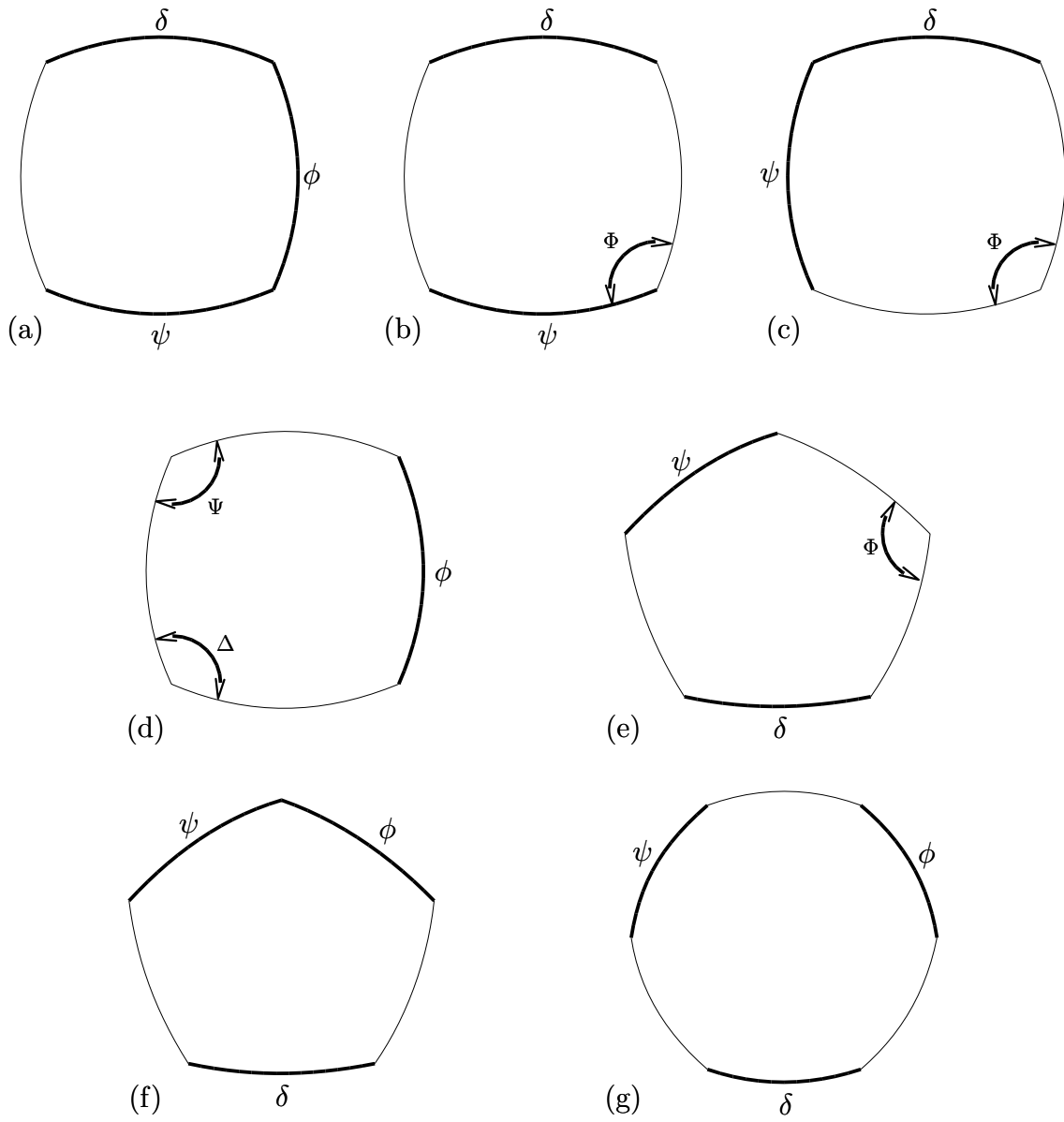


FIG. 11.6: Les différents polygones sphériques irréductibles

Résolution des polygones irréductibles seuls

Pour les problèmes simples, la résolution d'un polygone sphérique apparaîtra seule dans une étape de résolution. L'idée de cette résolution est de proposer une procédure simple pour réduire un polygone à un cas plus simple pour lequel une procédure peut être définie. Sept cas ont été identifiés. Pour chacun d'eux, une procédure de résolution peut être proposée. Elles ne seront pas toutes détaillées dans ce manuscrit, mais les idées principales en seront exposées.

La décomposition des polygones a été réalisée en recherchant des triplets de deux côtés consécutifs et un angle inscrit dont les mesures sont connues. Pour résoudre les configurations irréductibles, les triplets, composés de deux angles consécutifs et du côté inscrit aux valeurs connues, vont être recherchés. Cette combinaison permet de construire un triangle circonscrit au polygone afin d'identifier certains paramètres. Par exemple, prenons le cas de la figure 11.6-a. Pour le résoudre, il faut remarquer que le triangle sphérique a, e, d (Figure 11.7) est complètement défini (deux angles et une longueur connue). Ainsi, les côtés ae, de et l'angle $(\widehat{ea, ed})$ peuvent être calculés. L'angle $(\widehat{be, bc})$ (respectivement l'angle $(\widehat{ce, cb})$) est le complémentaire de l'angle $(\widehat{ba, bc})$ (respectivement l'angle $(\widehat{cd, cb})$). Les angles $(\widehat{ce, cb}), (\widehat{be, bc}), (\widehat{eb, ec})$ permettent de calculer les côtés du triangle $\{bec\}$. Finalement, la longueur des segments ab et dc se déduit respectivement des couples de segments ae, eb et de et ec .

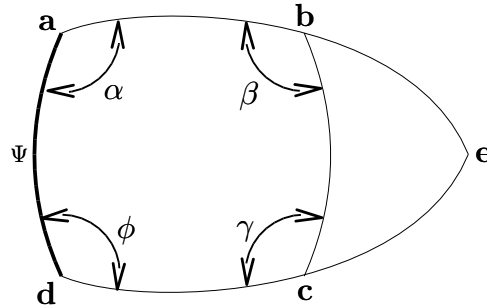


FIG. 11.7: Illustration de la procédure de réduction d'un polygone. Le côté Ψ et les angles sphériques $\alpha, \beta, \gamma, \phi$ sont de mesures connues.

Finalement, deux polygones ne peuvent pas être résolus exactement : les polygones c et d de la figure 11.6. Pour ceux-là, une décomposition arbitraire en triangles sphériques est proposée. Ensuite, un système d'équations nécessaire et suffisant devra être écrit à partir des relations de la trigonométrie sphérique liant les différents paramètres de ces triangles. La résolution sera alors réalisée par une procédure numérique.

Résolution des PPSI comportant plusieurs Polygones

Dans certains cas, il est imaginable de trouver un ensemble de polygones imbriqués (c'est-à-dire qu'ils doivent être résolus ensembles) et couplé à d'autres relations.

Dans ces cas, il est nécessaire de proposer une procédure permettant d'obtenir un système nécessaire et suffisant d'équations permettant une résolution numérique.

L'idée de cette procédure est, dans un premier temps, de ramener les polygones à des polygones irréductibles pour minimiser la taille du problème. Dans un second temps il s'agit de proposer une décomposition arbitraire des polygones irréductibles en triangles et pour chacun d'eux, de choisir un ensemble de relations.

11.2.7 Algorithmes d'optimisation

Pour résoudre les composantes sous-contraintes, nous avons choisi de les rendre bien-contraintes de façon plus ou moins aléatoire. Or, nous avons vu que cette approche ne conduit pas nécessairement à la solution. Dans certain cas, il va falloir résoudre globalement la composante. L'idée est d'utiliser des algorithmes d'optimisation afin de trouver une solution minimisant une fonctionnelle. Les algorithmes de minimisation sont courants dans la littérature : les algorithmes génétiques, les réseaux de neurones, BFGS (des noms de ses auteurs), le gradient conjugué en sont quelques exemples. Plusieurs fonctionnelles sont alors proposées à l'utilisateur afin de lui offrir le choix entre différents comportements.

11.2.8 Agent de résolution de zones comportant des courbes planes

Les surfaces et courbes paramétriques permettent la modélisation d'objets comportant des surfaces gauches avec rigueur. Or, la définition de ces modèles requiert de nombreuses heures de travail. La modification de tel modèle devient alors fastidieuse voir impossible au cours d'un processus de conception. Pour faciliter cette tâche, de nombreux travaux ont cherché à apporter des méthodes et des outils de modifications de ces surfaces et courbes plus intuitive pour l'utilisateur. Parmi ces méthodes, la méthode des densités de forces – Fdm¹ – se distingue pour sa propriété à offrir un comportement naturel à l'objet durant la modification tout en assurant des calculs instantanés en raison de la linéarité des relations entre les efforts extérieurs et les positions des nœuds libres du réseau.

Pour les mêmes raisons, la spécification par un ensemble de contraintes géométriques de ces familles d'objets n'a reçu que très peu de solutions. Le prin-

¹Force density method

cipe choisi dans la solution proposée est d'adapter les résultats obtenus à partir de la Fdm [Léon *et al.* 1995], [Guillet 1999].

Méthode des densités de forces

La méthode des densités de forces a été développée par Schek [Léon *et al.* 1995] pour calculer la position d'équilibre statique d'un réseau de barres quelconque.

Un réseau de barres est formé d'un ensemble de barres, sollicitées en traction, rotulées sans frottement en chaque nœud \mathbf{S}_i (Figure 11.8). Ces nœuds pouvant être soit libres, soit fixes. Les liaisons rotules étant parfaites, les équations de moments en chaque nœud du réseau n'interviennent pas. Une barre \mathbf{a}_j possède un paramètre mécanique q_j strictement positif nommé *densité de force*. À l'extrémité \mathbf{S}_i d'une barre \mathbf{a}_j de longueur l_j s'exerce une force \mathbf{f}_j d'intensité $f_j = l_j q_j$ portée dans le même sens que la barre. Avec ce modèle mécanique, le réseau de barres d'une surface paramétrique a un comportement de type membrane sous pression et celui d'une courbe, un comportement de câble pesant sous tension.

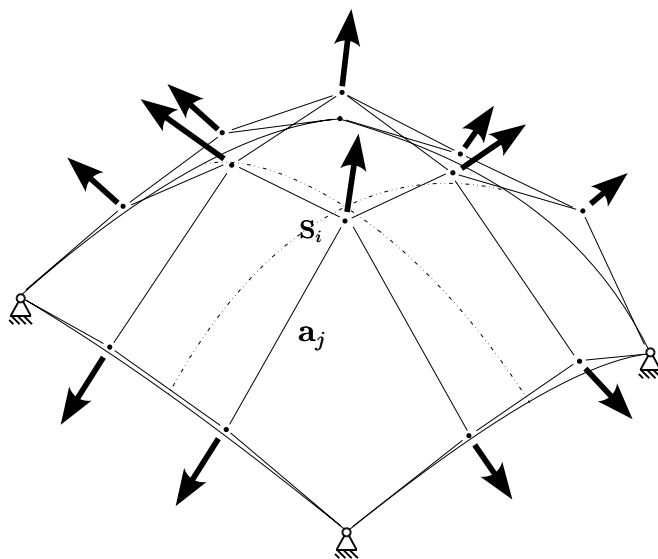


FIG. 11.8: Modèle mécanique proposé par Schek

Le principe de cette méthode appliquée aux courbes ou surfaces paramétriques est d'associer à leurs pôles, les nœuds du réseau de barres. La modification de la surface consiste alors à rechercher les modifications à réaliser sur la structure mécanique pour que l'objet associé corresponde aux attentes de l'utilisateur. Suite aux remarques de Guillet [Guillet 1999], la densité de force de chaque barre est un paramètre connu, mais la possibilité de définir des réseaux de barres de densité de force non uniforme est conservé. En conséquence, seule la modification de l'intensité et de la direction des forces est alors possible pour modifier une courbe.

En raison de son comportement mécanique, le réseau d'une surface, suite à la modification de l'une des forces appliquées aux différents pôles, subit une déformation particulièrement bien répartie, offrant ainsi à l'utilisateur l'impression d'avoir modelé son objet.

Génération des relations de la Fdm

La structure proposée pour une courbe paramétrique est identique à la représentation classique de son réseau caractéristique (Cf. sous-section 6.3.1). De la même façon, le réseau de barres associé à ces courbes est identique à leur structure.

L'approche initiale pose le problème dans un repère particulier. Or, celle exposée voit son originalité dans l'absence de repères, ce qui nécessite quelques adaptations. La principale consiste à définir les relations d'équilibre en chaque nœud du réseau de barres tout en gardant le formalisme général de l'approche. La figure 11.9-a illustre les éléments de la structure du réseau caractéristique d'une courbe paramétrique au voisinage du nœud S_i . Pour que le système puisse être en équilibre, il faut appliquer en chaque nœud S_i du réseau, une force \mathbf{f}_i (Figure 11.9-b) telle que :

$$\mathbf{f}_i + q_i \mathbf{a}_i + q_{i+1} \mathbf{a}_{i+1} = 0.$$

Pour cela, il suffit d'ajouter au système d'équations, pour chaque nœud S_i , les deux relations suivantes :

$$\mathbf{f}_i \cos \alpha_i - q_i \mathbf{a}_i + q_{i+1} \mathbf{a}_{i+1} \cos(\widehat{\mathbf{e}_i, \mathbf{e}_{i+1}}) = 0 \quad (11.8)$$

$$\mathbf{f}_i \sin \alpha_i + q_{i+1} \mathbf{a}_{i+1} \sin(\widehat{\mathbf{e}_i, \mathbf{e}_{i+1}}) = 0 \quad (11.9)$$

Ces relations génèrent, pour chaque nœud S_i , les variables f_i , et α_i .

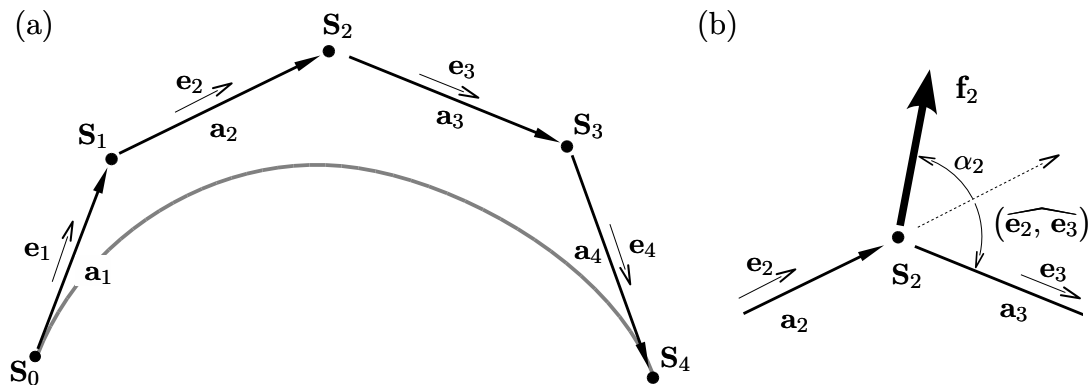


FIG. 11.9: Équilibre d'un nœud du réseau de barre

Résolution

Un problème comportant des courbes paramétriques associées à un comportement de Fdm est alors défini par un système d'équations comportant les équations et variables géométriques (Cf. section 8.3), des relations et variables d'ingénierie mais aussi les relations et variables de la Fdm. L'analyse qualitative du système va identifier des composantes sous-contraintes composées de relations de la Fdm. Contrairement aux autres composantes sous-contraintes, celles-ci deviennent directement des PPSI. Lors de la résolution, ces PPSI sont alors résolus par cet agent spécialisé. Pour cela, celui-ci fera simplement appel à un algorithme d'optimisation (Cf. sous-section 11.2.7) pour trouver une solution par rapport au comportement de la Fdm choisi par l'utilisateur, c'est-à-dire pour minimiser la fonction coût associée à ce comportement.

11.3 Les points clés de l'algorithme

L'algorithme global a été introduit dans la première section. Il fait appel à un ensemble d'agents de résolution qui ont été présentés à la section précédente. Le choix de l'agent est relativement simple, il sera exposé à la sous-section suivante. En revanche, l'algorithme est basé sur deux questions délicates : *existe-t-il une solution ?* et *existe-t-il d'autres solutions ?* Des réponses seront ébauchées dans la deuxième partie de cette section.

11.3.1 Choix du solveur

Un PPSI peut être de différentes natures. De plus, nous avons vu qu'il peut aussi correspondre à la résolution d'un problème géométrique connu. Le premier choix de l'agent est donc fonction de la nature du PPSI à résoudre :

- un PPSI linéaire est résolu par un agent linéaire. Il admet au plus une solution. Cette solution existe dès que le déterminant du jacobien est non nul. En raison des erreurs numériques qui peuvent être importantes dans certains cas (un pivot de Gauss trop petit par exemple), la résolution peut être confiée à un autre agent linéaire qui sera plus fiable mais plus coûteux ;
- un PPSI dual d'une résolution d'un cycle vectoriel est résolu par l'agent algébrique 2D. Il admet jusqu'à deux solutions. Celles-ci sont toujours connues puisqu'elles sont résolues de façon algébrique. L'inexistence de solutions pourra aussi être diagnostiquée ;
- pour la résolution des problèmes vectoriels 3D, l'utilisation de l'agent algébrique 3D permet d'obtenir dans les cas simples une résolution algébrique permettant de connaître toutes les solutions. Dans des cas plus complexes, celui-ci préparera le système d'équations et sous-traitera la

- résolution à un agent non-linéaire ;
- tous les autres PPSI sont résolus par un agent non-linéaire. En raison de la difficulté de cette tâche, différents agents sont proposés et se succèdent jusqu'à obtenir la solution.

Comme nous l'avons vu, plusieurs agents sont disponibles pour certaines familles de PPSI. Or pour ce type d'application, la réduction des temps de calcul est primordiale. Le choix de l'agent est donc basé sur des critères de performance. Malheureusement, les approches les plus rapides sont aussi les moins stables. Ainsi, en cas d'échec numérique, la résolution est proposée à un agent plus robuste mais plus lent. Par exemple, pour les problèmes non-linéaires, une première tentative est réalisée à partir de l'algorithme de Newton-Raphson. En cas d'échec, une méthode homotopique dans le corps des réels prend le relais. S'il n'existe pas de chemin, une troisième tentative est réalisée, toujours avec une méthode homotopique mais dans le corps des complexes, etc.

11.3.2 Existence d'une solution

Ce point est délicat. En effet, la stratégie de résolution utilisée met tout en œuvre pour trouver une solution. Or rien ne nous dit si, pour un système donné, il existe ou non de solution. Dans les cas où il n'existe pas de solution, le système risque de chercher en vain. Avec une autre approche, le système pourrait conclure à l'inexistence de solution dès le premier échec de résolution. Ainsi, il est donc important de répondre à cette question.

Les travaux de P. Serré [Serré 2000] ont apporté des premiers éléments de réponse à cette question. En effet, il propose des contraintes "d'existence" qui *"représentent le fait que l'objet à construire doit exister dans l'espace maximal auquel il serait susceptible d'appartenir"* [Serré 2000]. La formalisation de ces contraintes consiste à vérifier d'une part que tout ensemble de $(n + 1)$ points soit bien inclus dans un espace de dimension n et d'autre part que tout ensemble de n vecteurs soit défini dans un espace vectoriel de dimension n . Ainsi, à partir de ces contraintes, l'idée serait de vérifier qu'un objet, en fonction des spécifications, peut exister avant même d'être construit. Malheureusement, en pratique ceci n'est possible que pour un ensemble restreint de cas. En effet, ces vérifications nécessitent de connaître la mesure d'une grande partie des éléments. Par exemple, afin de vérifier qu'un ensemble de points est inclus dans un espace de dimension n , les relations proposées utilisent la distance entre chaque couple de points. Or le problème est en phase de résolution et donc cet ensemble de mesures n'est généralement pas encore connu. Quoiqu'il en soit, dans un problème géométrique classique, il est toujours possible de vérifier l'existence de certaines sous-figures avec une approche de cette nature.

Malheureusement, ces résultats sont à appliquer à un problème géométrique,

alors que dans l'approche présentée le problème est purement algébrique. Malgré tout, on s'aperçoit que les sous-parties qui pourraient être analysées avec le principe des contraintes d'existence sont des cas dont la résolution est connue. Ainsi, l'utilisation des solveurs algébriques permet de diagnostiquer, de façon exacte, l'inexistence de solution pendant la résolution.

11.3.3 Recherche de plusieurs solutions

Le problème de la recherche de plusieurs solutions est tout aussi difficile que le précédent. En effet, comment, à partir d'un système d'équations quelconques, peut-on calculer les différentes racines et surtout comment peut-on savoir si on les a toutes ?

En théorie, il n'y a pas de réponse. En pratique, on dispose d'un ensemble d'outils permettant dans les cas simples d'avoir des réponses exactes et dans les cas plus compliqués, d'avoir au moins plusieurs solutions lorsqu'elles existent. En effet, nous avons vu que dans les problèmes classiques certaines familles de PPSI apparaissaient souvent. De plus, pour ces familles de PPSI, les solutions algébriques sont connues. Ainsi, pour ces cas, toutes les solutions peuvent être trouvées.

Dans le cas général, l'agent de résolution multi-solutions est un outil qui permet de rechercher une grande partie des solutions. Ainsi, plusieurs, voire dans certains cas, toutes les solutions d'un système quelconque peuvent être trouvées. La limitation de cette approche est surtout due à la taille du système et au nombre de points de départ.

11.4 Les méta-agents

Le module de résolution dispose de trois méta-agents. Le premier, dont l'algorithme a été exposé à la première section, permet de résoudre la composante bien-contrainte d'un problème. Les deux autres sont destinés à résoudre les composantes sur et sous-contraintes. Or, nous avons vu que ces composantes étaient rendues artificiellement bien contraintes afin de définir une décomposition en PPSI. Leur résolution est donc similaire à celle de la composante bien contrainte. Malgré tout, ces composantes nécessitent quelques aménagements de l'algorithme afin de prendre en compte leurs différences.

11.4.1 Le méta-agent de résolution des problèmes sous-contraints

L'algorithme de ce méta-agent diffère très peu de l'algorithme classique. En effet, afin de rendre bien contrainte une composante sous-contrainte, la totalité des

variables puits dont les nœuds n'ont pas été saturés dans le graphe biparti a été supprimée de l'ensemble des inconnues du problème. En d'autres termes, ces **variables** sont devenues artificiellement des **variables sources**. Le plan de résolution envoyé à l'agent est donc un simple plan bien contraint.

Or, nous avons vu que le choix (arbitraire ou dirigé) des **variables** pouvait provoquer un problème d'existence, c'est-à-dire qu'en raison des valeurs initiales des **variables** sélectionnées, le problème sous-contraint n'admet pas de solution, alors qu'en théorie il en existe une infinité. Dans ce cas, une des solutions pourrait être de changer la sélection des variables "sous-contrainantes". Mais cette approche, pour des raisons d'une combinatoire trop importante, n'est pas viable. Ainsi, la solution retenue est d'offrir à l'utilisateur :

- soit la possibilité de diriger les modifications en lui présentant les choix réalisés ;
- soit la résolution globale de la composante par une approche de minimisation d'une fonction coût qu'il choisira ou proposera.

Pour cela, le méta-agent résout le problème bien contraint. En cas d'échec, il sollicite l'utilisateur. En fonction de ses choix, il doit :

- soit lancer une analyse de la composante avec les nouvelles informations afin d'obtenir un nouveau plan de résolution (pour relancer une nouvelle résolution) ;
- soit faire appel à un agent disposant d'un algorithme d'optimisation, afin de trouver une solution globale minimisant la fonctionnelle choisie.

11.4.2 Le méta-agent de résolution des problèmes sur-contraints

Le plan de résolution d'une composante sur-contrainte diffère par la présence de PPSI sur-contrainants, c'est-à-dire d'une équation dont le nœud n'a pas été saturé pendant l'analyse. Ces équations ou inéquations ne servent donc pas à la résolution mais doivent être vérifiées. Pour mettre en place cela, l'idée consiste simplement à ajouter un agent particulier dont la tâche n'est pas de résoudre comme le font les autres mais de regarder si la relation est ou non vérifiée. Dans le cas où elle ne l'est pas, l'agent se comporte comme un agent qui diagnostique l'inexistence de solution d'un PPSI.

11.5 Conclusion

Ce chapitre a présenté le module de résolution. Il est composé de trois méta-agents organisant la résolution des différents PPSI. Celle-ci est réalisée par des agents de résolution élémentaire. Pour chaque PPSI, l'agent le plus adapté est choisi afin de rendre la résolution plus rapide dans un premier temps, mais surtout

plus robuste dès que nécessaire. Ainsi, tout est réalisé afin de trouver une ou plusieurs solutions. Malheureusement, dans certains cas, aucune solution ne sera trouvée. La cause peut avoir de multiples origines. Quoiqu'il en soit, elle doit être présentée à l'utilisateur afin que celui-ci puisse réaliser les corrections nécessaires. Ce thème est l'objet du chapitre suivant.

Chapitre 12

La génération de diagnostics

Le module de résolution présenté n'est pas un simple solveur algébrique. En effet, en plus de la résolution, il propose différentes analyses nécessaires à l'utilisateur pour comprendre l'état de contrainte de son problème et les causes d'éventuels échecs. Le dernier outil nécessaire à ce module va donc être proposé. Son rôle est de préparer les différentes analyses et de présenter les diagnostics.

12.1 Introduction

La résolution est basée sur une stratégie de décomposition du problème pour deux raisons : la première, et principale, est le gain obtenu en vitesse et en robustesse ; la seconde est la capacité d'analyse obtenue. En effet, nous avons vu que la première étape de la construction du plan de résolution permettait d'identifier les composantes sous-, sur- et bien contraintes du système. De plus, une résolution progressive permet de décomposer le problème en cours de résolution, en quatre parties : tout d'abord le système à résoudre, puis ce qui a déjà été résolu, ensuite ce qui n'a pas encore été résolu et enfin les éléments qui ne sont pas impliqués par et dans la résolution du PPSI courant. Nous allons voir que ces deux descriptions du plan de résolution sont à la base de la génération de diagnostics.

L'identification des composantes du plan de résolution a déjà été présentée. La seconde section de ce chapitre qui aborde la préparation du diagnostic va donc décrire plus en détail l'isolement des différentes parties d'un système en cours de résolution. Le point clé de ce chapitre est la génération du diagnostic. En effet, celle-ci doit être basée sur le modèle géométrique pour avoir un point de vue graphique et sur le modèle d'information pour prendre en compte toutes les spécifications. Ce point sera présenté dans la troisième section.

12.2 Préparation d'un diagnostic en cas d'échec

La préparation d'un diagnostic, qu'il soit pré- ou post-résolution, nécessite de reconnaître dans le problème, différentes parties dont les caractéristiques permettront de proposer un diagnostic. Pour les diagnostics post-résolution, les caractéristiques des composantes recherchées décrivent l'état de contrainte local du problème. En cas d'échec de résolution, les composantes à trouver sont soit résolues, soit à résoudre, soit indépendantes.

12.2.1 Recherche des différentes composantes du problème

La recherche des différentes composantes du problème est réalisée par le module d'analyse. En effet, la seconde étape de l'analyse structurale consiste à extraire du système global les composantes semi-irréductibles, autrement dit, les composantes sur- et sous-contraintes (Cf. sous-section 10.2.1). Par définition, il n'y a qu'une composante bien contrainte.

12.2.2 Isolement des différentes parties d'un système en cours de résolution

Nous avons vu que le plan de résolution est un graphe orienté dont les arcs représentent les dépendances entre les différents PPSI (Figure 12.1). À partir

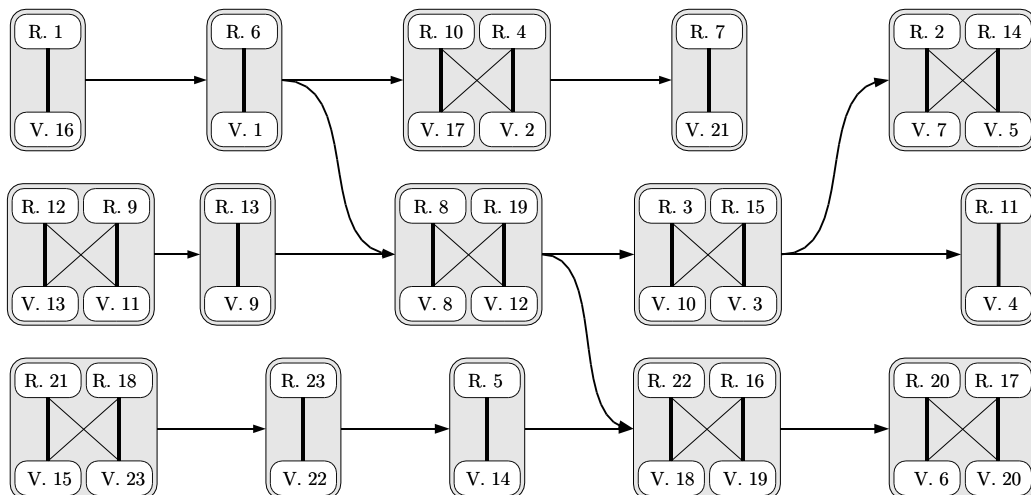


FIG. 12.1: Un plan de résolution.

d'un PPSI, le graphe peut être partitionné en trois ensembles :

- le premier regroupe l'ensemble des PPSI nécessaires à la résolution du PPSI courant. Il est composé d'un ensemble de sous-graphes connexes. Pour

l'identifier, il suffit de rechercher une forêt orientée parcourant le plan de résolution en sens inverse et dont la racine des arbres précède le PPSI courant (Figure 12.2). Les composantes connexes de cette forêt représentent les parties rigides (car déjà résolues) de la structure nécessaires à la résolution du PPSI courant ;

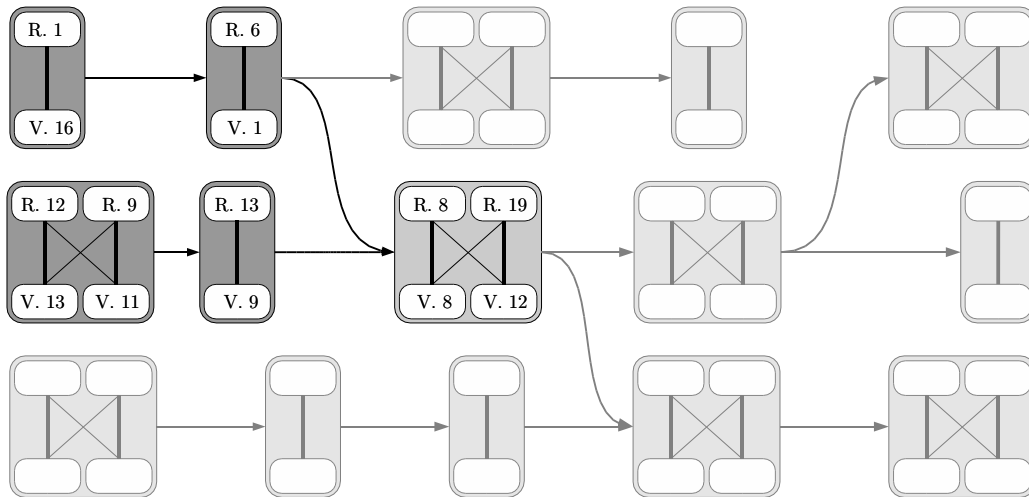


FIG. 12.2: Les PPSI nécessaires à la résolution du PPSI courant défini par les relations *R. 8 et R. 19* et les variables *V. 8 et V. 12*.

- le second contient l'ensemble des PPSI dépendant de la résolution du PPSI courant. Pour l'identifier, il suffit de rechercher l'arbre orienté parcourant le plan de résolution dont la racine est ce PPSI (Figure 12.3). Cet ensemble n'existe pas lorsque le PPSI courant est un PPSI sur-contraignant. Or ces PPSI ne sont définis que pour résoudre facilement le problème. L'échec de résolution d'un PPSI sur-contraignant est donc équivalent à l'échec de résolution d'une composante sur-contrainte. L'ensemble des PPSI dépendants est donc défini par l'arbre issu de cette composante (Figure 12.3).
- le troisième est composé des PPSI qui ne sont pas liés au PPSI courant.

12.3 Génération d'un diagnostic

À partir de ces ensembles de PPSI, les spécifications impliquées devront être retrouvées. En effet, pour que le diagnostic soit compris facilement par l'utilisateur, il faut qu'il soit énoncé à partir des spécifications. En pratique, cette tâche est relativement simple. En effet, chaque PPSI ou composante permet de résoudre un ensemble de **variables**. Or chaque **variable** est associée à un ensemble de spécifications. De plus, certaines relations sont, elles aussi, liées à des spécifications. Par exemple, les contraintes d'ingénierie sont directement for-

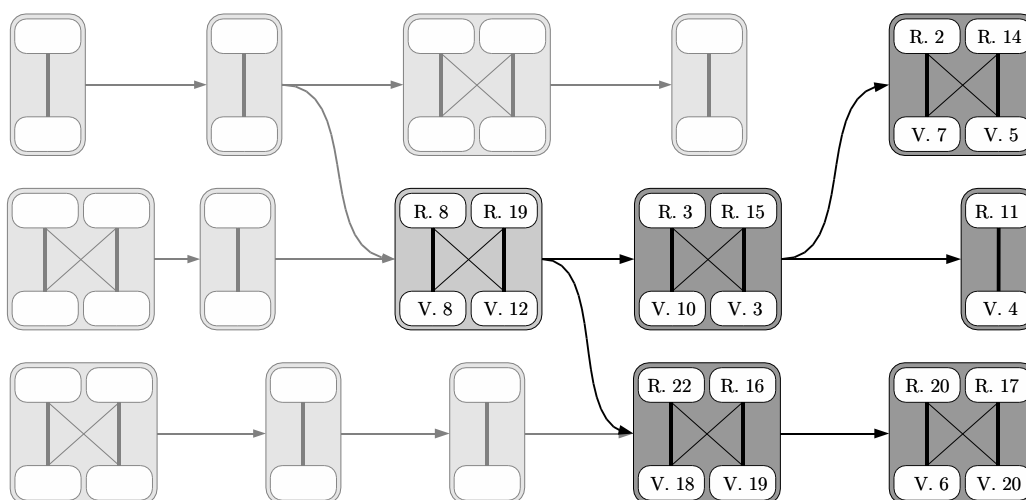


FIG. 12.3: Les PPSI dépendants de la résolution du PPSI courant.

mulées sous une forme algébrique. Ainsi, chaque ensemble de PPSI est lié à un ensemble de spécifications. La génération d'un bon diagnostic consiste alors à trouver une présentation des différents groupes de spécifications permettant une interprétation intuitive.

12.3.1 Diagnostic pré-résolution

Le diagnostic pré-résolution consiste à indiquer à l'utilisateur l'état de contrainte de son problème. Nous avons vu que l'analyse structurale a permis de décomposer le problème en composantes bien, sur- et sous-contraintes. Chaque composante est liée à un ensemble de spécifications qui a pu être identifié. Il suffit ensuite de les montrer à l'utilisateur. Par exemple, pour la figure 12.4-a représentant un objet géométrique contraint, les deux parties sur-contraintes de la pièce sont illustrées par les figures 12.4-b et 12.4-c. L'ensemble bien contraint de la géométrie et pour finir les zones sous-contraintes sont ensuite localisées (Figures 12.4-d et 12.4-e). En pratique, un code de couleur peut être utilisé afin de localiser les différents groupes sur l'objet géométrique.

Les spécifications sont maintenant couramment représentées dans un arbre de *features*. Dans une approche variationnelle, elles n'ont pas d'ordre. Elles peuvent alors être regroupées par composantes afin de permettre à l'utilisateur de retrouver les spécifications d'ingénierie contenues dans les différents groupes.

12.3.2 Diagnostic post-résolution

Le diagnostic post-résolution va décrire les causes d'un échec de résolution. Un échec est porté par un PPSI. Il suit un ensemble de PPSI dont la résolution a convergé, mais va stopper la résolution de certains PPSI, alors qu'une partie

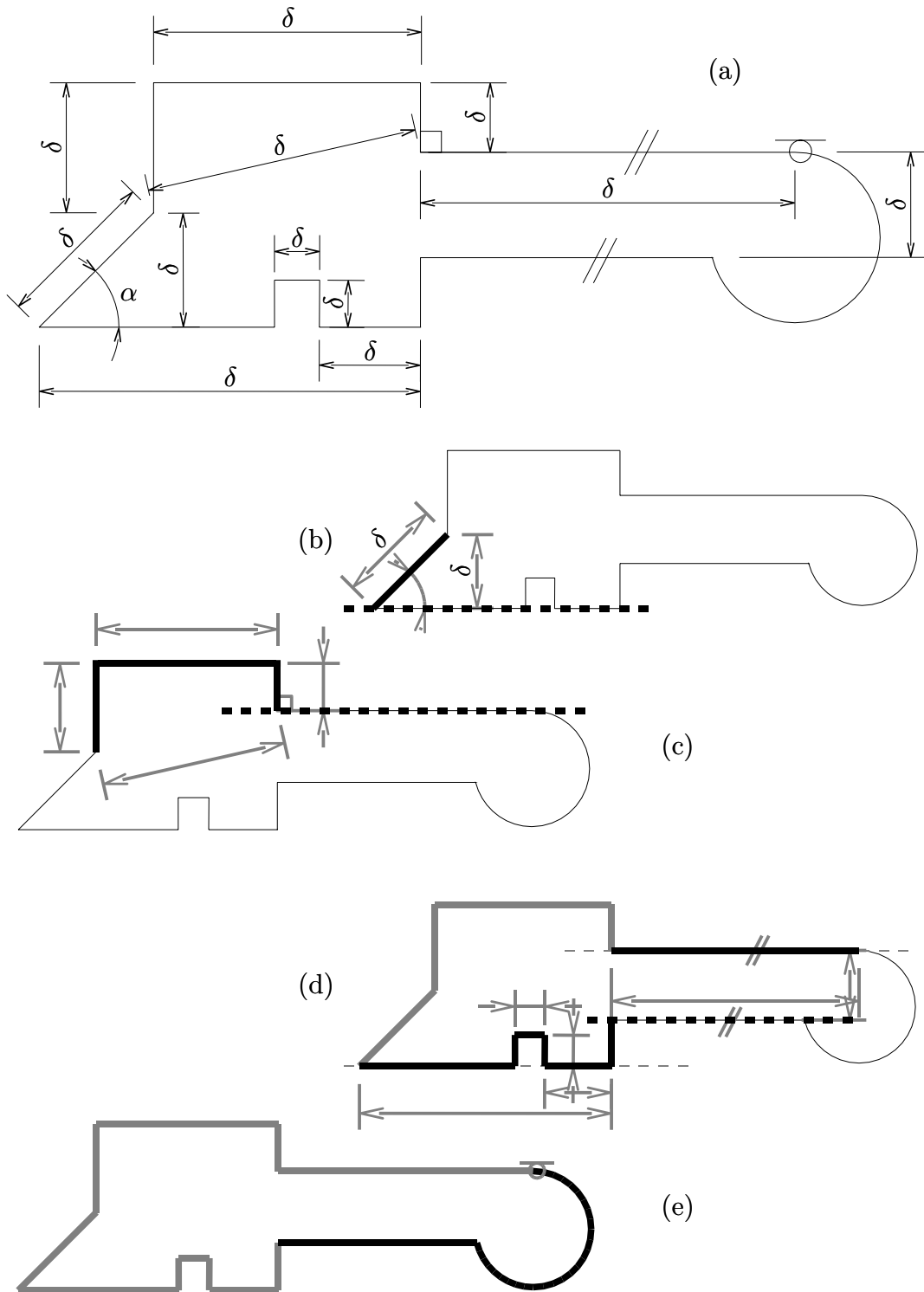


FIG. 12.4: Exemple de diagnostic sur l'état de contrainte.

indépendante doit pouvoir être résolue. Ces trois groupes doivent être indiqués à l'utilisateur afin qu'il puisse se rendre compte de l'influence de l'échec.

Le sous-graphe contenant l'ensemble des PPSI qui ont pu être résolus et qui sont nécessaires à la résolution du PPSI en échec peuvent avoir plusieurs composantes connexes. Chaque composante définit une partie rigide et indépendante de l'objet. Elles doivent donc être distinguées entre elles. La figure 12.5-a propose un objet géométrique qui servira d'exemple. Les figures 12.5-b et 12.5-c présentent les deux parties dont la résolution a convergé. En traits fins, on distingue les éléments définis par rapport à ces parties et pouvant être résolus. La résolution de ces deux parties est aussi nécessaire à leur assemblage avec les deux segments (Figure 12.5-d). En raison de la valeur des spécifications, cet assemblage est impossible. Il y a donc un problème d'existence. Sur cette dernière figure, on distingue en traits fins les éléments impliqués par l'échec de la résolution.

12.4 Conclusion

L'outil de diagnostic proposé est un outil nécessaire dès que la taille du problème à résoudre devient conséquente. En effet, même pour un problème de dimension deux sans contrainte d'ingénierie, l'œil averti du concepteur peut avoir des difficultés à déceler les erreurs de spécifications. Ce module sera d'autant plus nécessaire que la dimension des problèmes à traiter augmentera et que l'utilisation des contraintes d'ingénierie se généralisera.

Un diagnostic composé de paquets d'équations et de variables n'est pas acceptable. Au contraire, il doit être facilement interprété par l'utilisateur. La solution proposée va dans cette direction, mais pour cela, la connaissance des spécifications est nécessaire. Pour cette raison, l'outil de diagnostic est dépendant du modèle de spécifications.

La présentation du dernier outil termine l'exposé du module de résolution. Nous avons vu qu'il est globalement indépendant du modèle dynamique de spécifications puisqu'il ne traite qu'un système d'équations algébriques. Mais cet outil est dépendant d'une *bonne* mise en équations. Finalement, il permet de répondre, en collaboration avec le modèle dynamique de spécifications, à de nombreux points du cahier des charges (Cf. section 4.2).

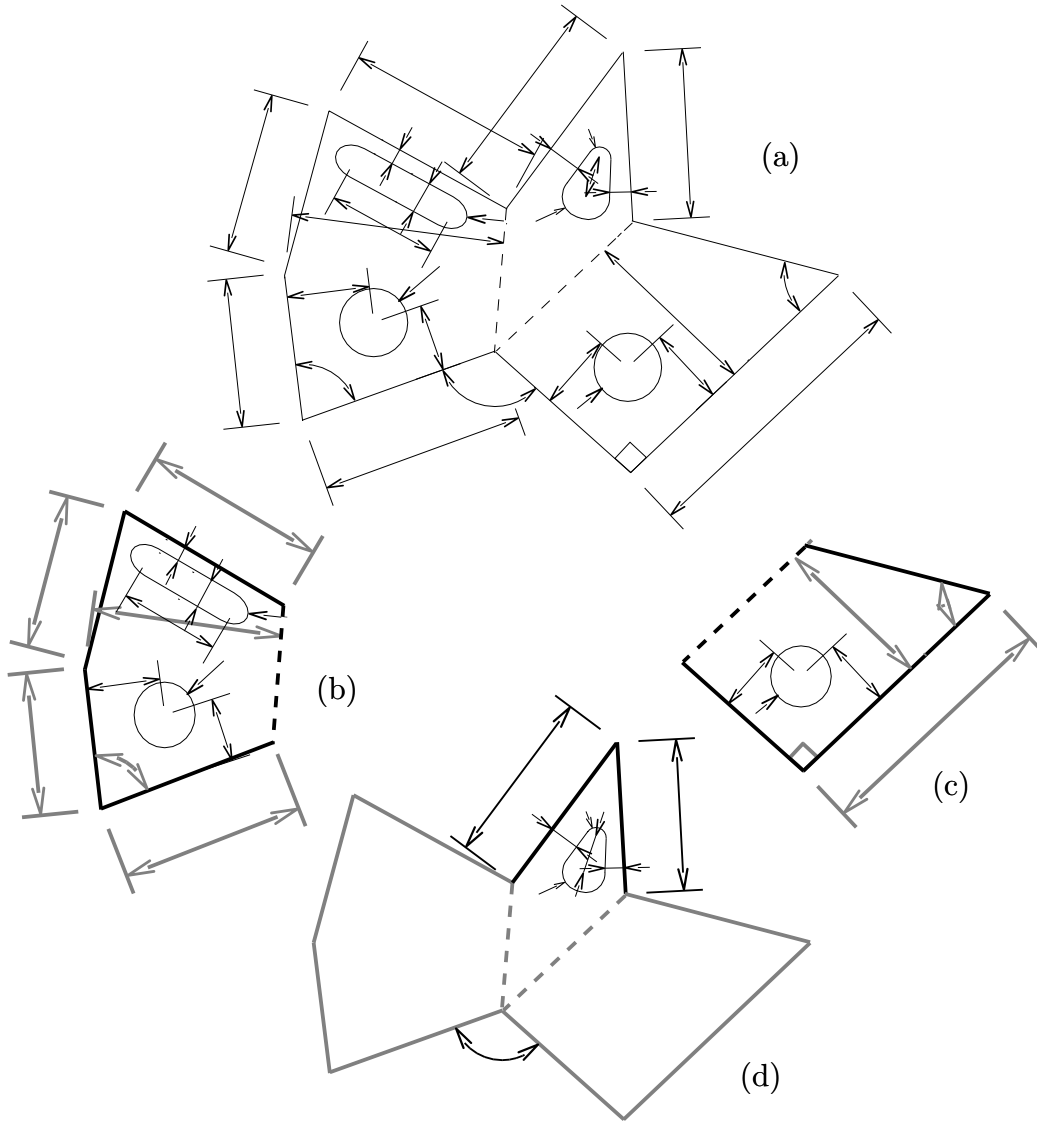


FIG. 12.5: Exemple de diagnostic pour l'échec de la résolution d'un PPSI.

Quatrième partie

Maquette, exemples, résultats et conclusion

Chapitre 13

Maquette informatique, exemples et résultats

Suite à une rapide présentation de la maquette informatique, ce dernier chapitre détaille la résolution de quelques cas. Ainsi, le lecteur pourra apprécier les apports obtenus grâce aux différents concepts proposés.

La seconde section commence donc par un cas simple afin de familiariser le lecteur à toutes les étapes de la résolution. Ce premier exemple est suivi d'un cas mettant en œuvre les spécifications originales proposées pour les coniques alors que les sous-sections 13.2.3 et 13.2.4 démontrent d'une part l'intérêt principal de l'approche en présentant un réel problème d'ingénierie et d'autre part, illustre un dernier apport qui n'est pas négligeable : la spécification de courbes NURBS. Avant de terminer ce chapitre, le déroulement de la correction d'un plan de résolution est présenté. Finalement, le plan de résolution d'un problème géométrique réel est proposé.

13.1 Maquette informatique

La maquette informatique réalisée, a été développée dans l'environnement de prototypage du service de recherche en nouvelles technologies de la société Dassault Systèmes. Cet environnement appelé MOKA (Mechanical Oriented Knowledge Application) est écrit entièrement en Java. La figure 13.1 illustre l'application. On distingue entre autres une esquisse et la boîte de dialogue proposant les différentes contraintes applicables aux éléments sélectionnés.

Cette application a permis de valider un ensemble de concepts présentés dans ce manuscrit. En raison du faible apport visuel de l'application, des exemples vont être exposés en détail afin de bien illustrer les différentes étapes de la résolution.

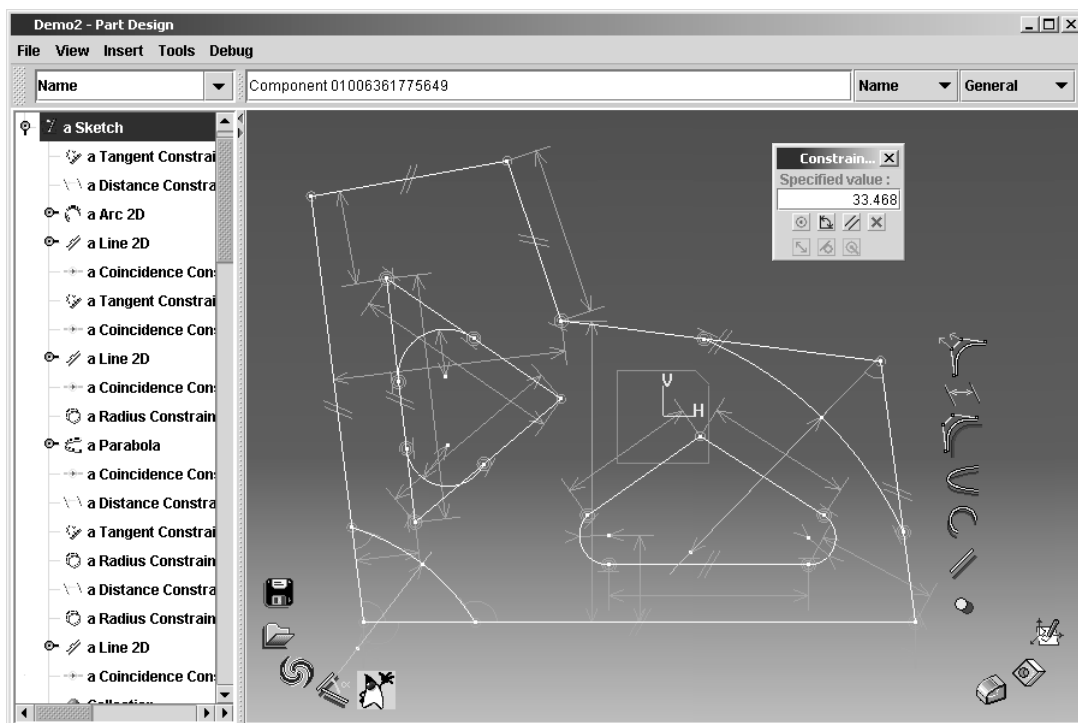


FIG. 13.1: Interface graphique de la maquette informatique

13.2 Exemples et résultats

13.2.1 Un exemple élémentaire de géométrie contrainte

Le premier exemple (Figure 13.2) représente un quadrilatère spécifié par

- une contrainte de distance “point – droite” de mesure d_1 ;
- deux distances “point – point” de mesures d_2 et d_3 ;
- un angle “droite – droite” de mesure α_1 ;
- une contrainte de parallélisme “droite – droite”.

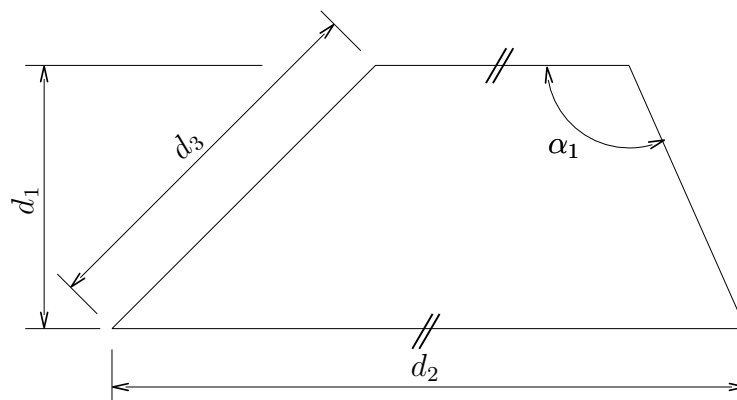


FIG. 13.2: Un premier exemple simple de géométrie contrainte

La structure associée à cet objet est composée d'un graphe affine et d'un graphe vectoriel, représentés à la figure 13.3. Sur le graphe affine (Figure 13.3-a), on reconnaît la structure de la contrainte de distance "point – droite" présentée à la sous-section 7.5.2. Les éléments spécifiés de ces graphes sont mis en évidence par des arcs épais.

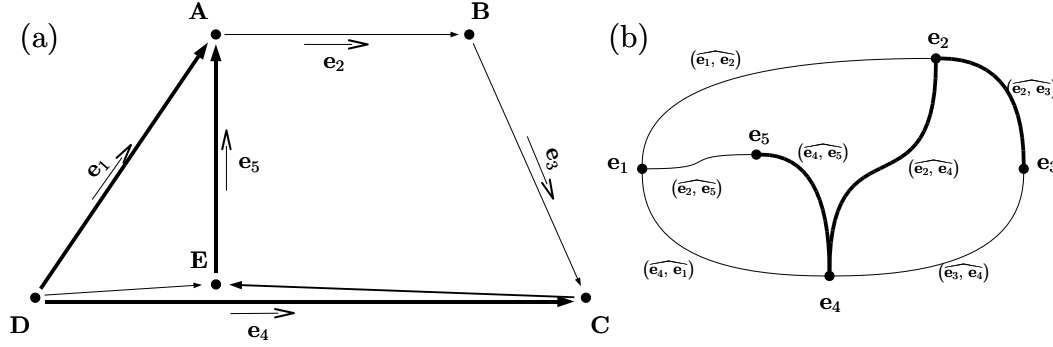


FIG. 13.3: La structure de l'objet de la figure 13.2 est composée d'un graphe affine (a) et d'un graphe vectoriel (b)

À partir du graphe affine, une base de cycles de poids minimum est identifiée (Cf. sous-section 8.4.3). Cette base est composée des cycles :

- A, D, E;
- A, B, C, D;
- C, D, E.

Ces trois cycles permettent d'écrire les relations affines nécessaires à la résolution du problème. Pour cela, il faut identifier, pour chaque cycle, le vecteur de projection à l'aide des différentes heuristiques proposées à la sous-section 8.4.3. Dans cet exemple, tous les cycles sont à projeter sur le vecteur \mathbf{e}_4 . En effet, celui-ci est commun à tous les cycles et est utilisé par deux angles spécifiés : $(\widehat{\mathbf{e}_4, \mathbf{e}_2})$, $(\widehat{\mathbf{e}_4, \mathbf{e}_5})$. La projection de ces cycles sur ce vecteur permet d'obtenir le système d'équations suivant :

$$\begin{cases} \mathbf{DE} + \mathbf{EA} \cos(\widehat{\mathbf{e}_5, \mathbf{e}_4}) - \mathbf{AD} \cos(\widehat{\mathbf{e}_1, \mathbf{e}_4}) = 0 \\ \mathbf{EA} \sin(\widehat{\mathbf{e}_5, \mathbf{e}_4}) - \mathbf{AD} \sin(\widehat{\mathbf{e}_1, \mathbf{e}_4}) = 0 \\ \mathbf{AB} \cos(\widehat{\mathbf{e}_2, \mathbf{e}_4}) - \mathbf{BC} \cos(\widehat{\mathbf{e}_3, \mathbf{e}_4}) - \mathbf{CD} + \mathbf{DA} \cos(\widehat{\mathbf{e}_1, \mathbf{e}_4}) = 0 \\ \mathbf{AB} \sin(\widehat{\mathbf{e}_2, \mathbf{e}_4}) - \mathbf{BC} \sin(\widehat{\mathbf{e}_3, \mathbf{e}_4}) + \mathbf{DA} \sin(\widehat{\mathbf{e}_1, \mathbf{e}_4}) = 0 \\ \mathbf{CE} - \mathbf{ED} + \mathbf{DC} = 0 \end{cases} \quad (13.1)$$

Cette projection n'a pas nécessité la création de nouveaux angles. La base de cycles identifiée dans le graphe vectoriel a permis d'écrire les relations angulaires suivantes :

$$\begin{cases} (\widehat{\mathbf{e}_1, \mathbf{e}_4}) + (\widehat{\mathbf{e}_4, \mathbf{e}_5}) + (\widehat{\mathbf{e}_5, \mathbf{e}_1}) = 0 [2\pi] \\ (\widehat{\mathbf{e}_1, \mathbf{e}_4}) + (\widehat{\mathbf{e}_4, \mathbf{e}_2}) + (\widehat{\mathbf{e}_2, \mathbf{e}_1}) = 0 [2\pi] \\ (\widehat{\mathbf{e}_2, \mathbf{e}_3}) + (\widehat{\mathbf{e}_3, \mathbf{e}_4}) + (\widehat{\mathbf{e}_4, \mathbf{e}_2}) = 0 [2\pi] \end{cases} \quad (13.2)$$

À partir de ce système d'équations, l'analyse qualitative peut commencer afin de définir le plan de résolution. Pour cela le graphe biparti associé est construit (Figure 13.4).

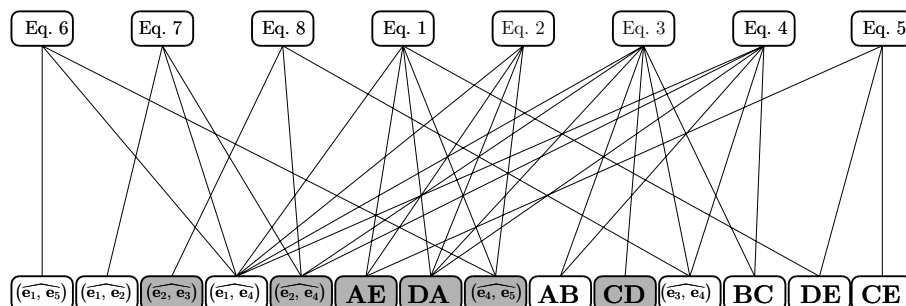


FIG. 13.4: Graphe biparti de l'exemple de la figure 13.2.

À partir de ce graphe, le plan de résolution est identifié à l'aide de l'algorithme décrit à la sous-section 10.2.1. Ce plan de résolution est illustré à la figure 13.5.

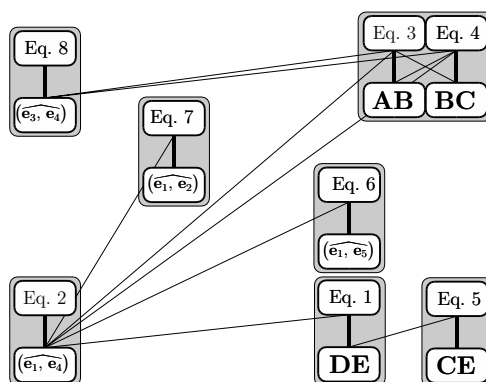


FIG. 13.5: Plan de résolution du problème de la figure 13.2.

13.2.2 Deux arcs de cercles et une parabole

Ce second exemple (Figure 13.6) est présenté afin d'illustrer la spécification des coniques. Pour cela, le modèle proposé est composé de deux arcs de cercles liés par un segment et une parabole. La longueur du segment, le rayon des arcs de cercles et la distance focale de la parabole sont spécifiés. De plus, les arcs adjacents sont continus en tangence. La dernière spécification utilisée contraint la direction de l'axe focal de la parabole par rapport au segment de droite.

La structure de cet objet, représentée à la figure 13.7, permet d'identifier les relations nécessaires au maintien de la cohérence. Les structures des arcs de cercle et de la parabole contiennent des éléments de même mesure, comme les angles $(\widehat{e_8, e_9})$, $(\widehat{e_9, e_{10}})$, $(\widehat{e_{10}, e_{11}})$. Ainsi, dans les graphes, ces ensembles d'éléments

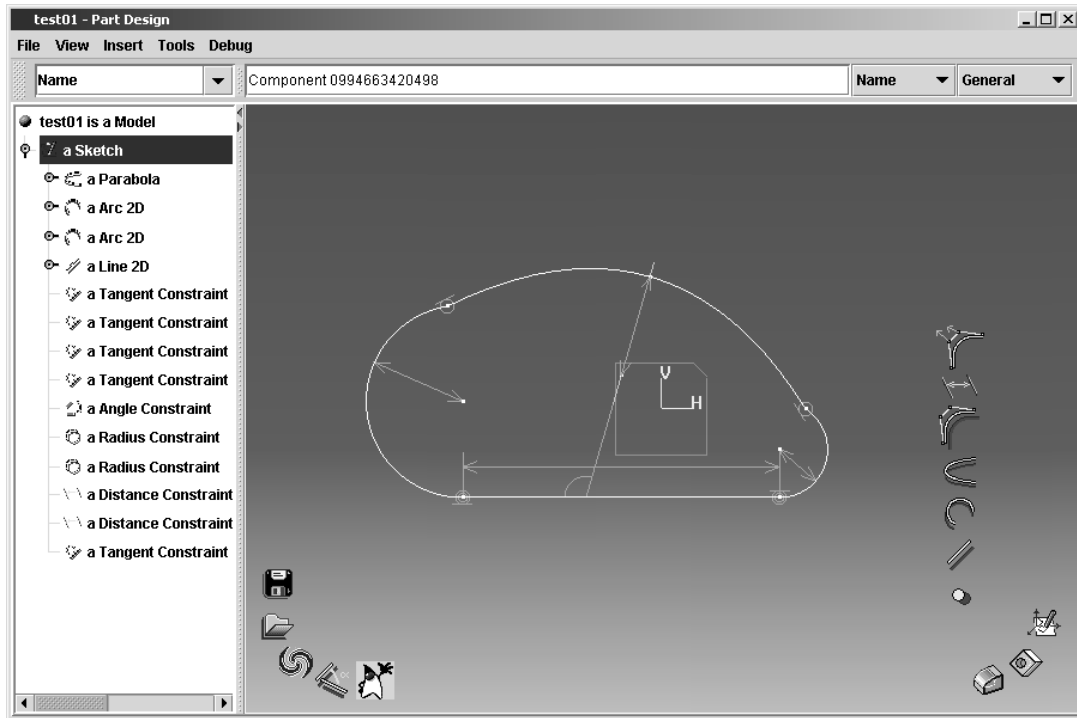


FIG. 13.6: Un exemple composé de cercles et d'une parabole

ne représentent qu'une seule inconnue. Les arcs pointillés de la figure 13.7-(b) illustrent ce point.

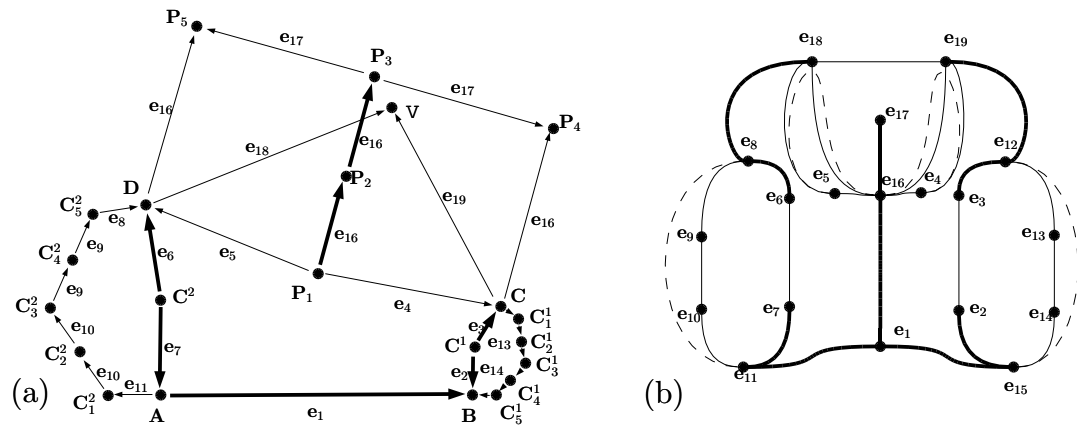


FIG. 13.7: Structure de l'objet de la figure 13.6

La base de cycles de poids minimum du graphe affine, (Figure 13.7-(a)), est :

- A, B, C¹, C, P₁, D, C² ;
- C, C₁¹, C₂¹, C₃¹, C₄¹, C₅¹, B, C¹ ;
- A, C₁², C₂², C₃², C₄², C₅², D, C² ;
- D, V, C, P₁ ;
- P₁, P₂, P₃, P₄, C ;

- $\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3, \mathbf{P}_5, \mathbf{D}$.

Contrairement au premier exemple dans lequel les trois cycles étaient projetés sur le même vecteur, le vecteur de projection est différent pour chaque cycle. En effet :

- le cycle $\mathbf{A}, \mathbf{B}, \mathbf{C}^1, \mathbf{C}, \mathbf{P}_1, \mathbf{D}, \mathbf{C}^2$ est projeté sur le vecteur \mathbf{e}_1 qui est utilisé par trois angles de mesure imposée. Ces projections permettent d'écrire les relations (1) et (2) du graphe biparti de la figure 13.9 ;
- les cycles $\mathbf{C}, \mathbf{C}_1^1, \mathbf{C}_2^1, \mathbf{C}_3^1, \mathbf{C}_4^1, \mathbf{C}_5^1, \mathbf{B}, \mathbf{C}^1$ et $\mathbf{A}, \mathbf{C}_1^2, \mathbf{C}_2^2, \mathbf{C}_3^2, \mathbf{C}_4^2, \mathbf{C}_5^2, \mathbf{D}, \mathbf{C}^2$ seront projetés respectivement sur \mathbf{e}_2 et \mathbf{e}_8 , tous deux utilisés par trois angles dont deux sont spécifiés (Équations (11), (12), (3) et (4)) ;
- les vecteurs \mathbf{e}_{18} et \mathbf{e}_{19} peuvent tous deux définir le support de la projection du cycle $\mathbf{D}, \mathbf{V}, \mathbf{C}, \mathbf{P}_1$. Le vecteur \mathbf{e}_{18} est choisi arbitrairement et permettra d'obtenir les relations (5) et (6) ;
- les cycles $\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3, \mathbf{P}_4, \mathbf{C}$ et $\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3, \mathbf{P}_5, \mathbf{D}$ sont, quant à eux, projetés sur le même vecteur \mathbf{e}_{16} (Équations (7), (8), (9) et (10)). Celui-ci est commun aux deux cycles et utilisé par six angles, dont deux sont contraints.

Les projections de ces cycles ont nécessité la création d'un ensemble d'angles. Ils sont représentés en pointillés sur la figure 13.8 illustrant le graphe vectoriel de la structure de l'objet ainsi complété.

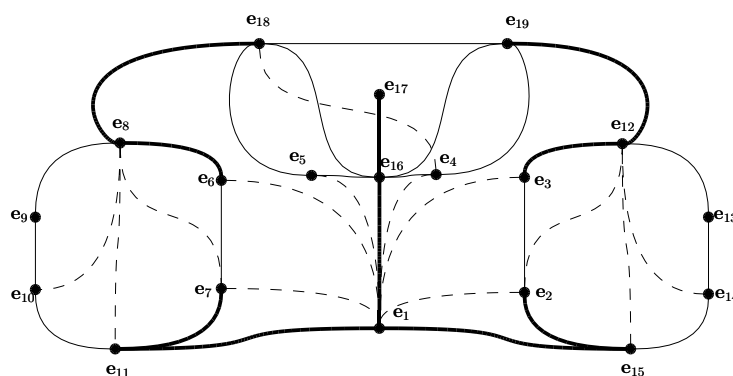


FIG. 13.8: Graphe vectoriel complété de la structure de l'objet de la figure 13.6

La base du graphe vectoriel devient :

- $\mathbf{e}_8, \mathbf{e}_9, \mathbf{e}_{10}$;
- $\mathbf{e}_8, \mathbf{e}_{10}, \mathbf{e}_{11}$;
- $\mathbf{e}_8, \mathbf{e}_{11}, \mathbf{e}_7$;
- $\mathbf{e}_6, \mathbf{e}_7, \mathbf{e}_8$;
- $\mathbf{e}_1, \mathbf{e}_7, \mathbf{e}_{11}$;
- $\mathbf{e}_1, \mathbf{e}_6, \mathbf{e}_7, \mathbf{e}_{11}$;
- $\mathbf{e}_1, \mathbf{e}_5, \mathbf{e}_{18}, \mathbf{e}_8, \mathbf{e}_6, \mathbf{e}_1$;
- $\mathbf{e}_1, \mathbf{e}_5, \mathbf{e}_{16}$;

- $\mathbf{e}_{16}, \mathbf{e}_{18}, \mathbf{e}_8, \mathbf{e}_6, \mathbf{e}_1$;
- $\mathbf{e}_{18}, \mathbf{e}_4, \mathbf{e}_1, \mathbf{e}_6, \mathbf{e}_8$;
- $\mathbf{e}_{12}, \mathbf{e}_{13}, \mathbf{e}_{14}$;
- $\mathbf{e}_{12}, \mathbf{e}_{14}, \mathbf{e}_{15}$;
- $\mathbf{e}_{12}, \mathbf{e}_2, \mathbf{e}_{15}$;
- $\mathbf{e}_{12}, \mathbf{e}_3, \mathbf{e}_2$;
- $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_{15}$;
- $\mathbf{e}_1, \mathbf{e}_3, \mathbf{e}_2, \mathbf{e}_{15}$;
- $\mathbf{e}_1, \mathbf{e}_4, \mathbf{e}_{19}, \mathbf{e}_{12}, \mathbf{e}_3, \mathbf{e}_1$;
- $\mathbf{e}_1, \mathbf{e}_4, \mathbf{e}_{16}$;
- $\mathbf{e}_{16}, \mathbf{e}_{19}, \mathbf{e}_{12}, \mathbf{e}_3, \mathbf{e}_1$;
- $\mathbf{e}_{16}, \mathbf{e}_{18}, \mathbf{e}_{19}$;

Elle permet d'obtenir les équations vectorielles nécessaires à la résolution (Équations (13) à (32) du graphe biparti de la figure 13.9).

L'analyse du graphe biparti permet d'obtenir le plan de résolution de ce problème (Figure 13.10). Contrairement au plan de la figure 13.5, seules les dépendances directes entre PPSI sont représentées afin de le rendre plus lisible. Ce plan est composé de trois grandes étapes successives :

- la première consiste à résoudre deux PPSI en parallèle. Ils sont chacun composés d'une équation linéaire et permettent de calculer les angles $(\widehat{\mathbf{e}_1, \mathbf{e}_2})$ et $(\widehat{\mathbf{e}_1, \mathbf{e}_7})$;
- la seconde est composée d'un seul PPSI. Ce PPSI est conséquent. Il est composé de 12 équations. Il illustre que l'analyse ne peut rien faire contre les problèmes dont les spécifications sont trop imbriquées. En effet, partant du segment, les deux arcs de cercles sont parfaitement positionnés, tout comme la parabole qui est, elle aussi, parfaitement définie puisque sa direction et sa distance focale sont spécifiées. Or, les secondes extrémités des arcs de cercles ne sont définies que par leurs relations avec les extrémités de l'arc de parabole qui leur est tangent. Le problème est donc globalement bien contraint, mais quelque soit la méthode, pour résoudre il faut prendre en considération une grande partie des spécifications ;
- la troisième partie complète la résolution. On distingue deux branches presque identiques qui traduisent la fin de la résolution de la structure des arcs de cercle. En effet, le troisième PPSI n'a défini que la position des centres de ces arcs. De même, la troisième branche permet d'identifier les derniers éléments de la structure de la parabole. On observe sur les branches associées aux arcs de cercles, deux PPSI sur-contrainants. La présence de ces PPSI est due à la structure sur-contrainte des arcs de cercles.

Une solution de la résolution de ce problème est représentée à la figure 13.11.

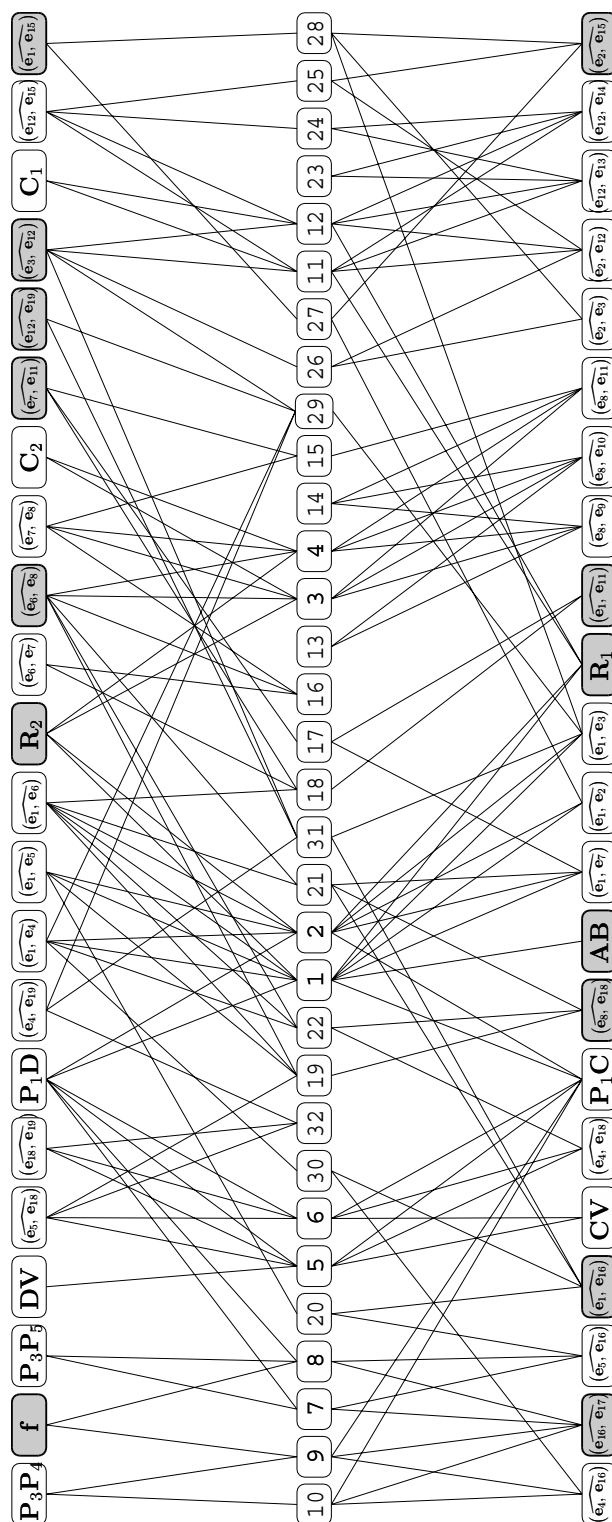


FIG. 13.9: Graphe biparti du problème de la figure 13.6. Afin de faciliter la lecture, l'ensemble des variables est divisé en deux et réparti autour des équations.

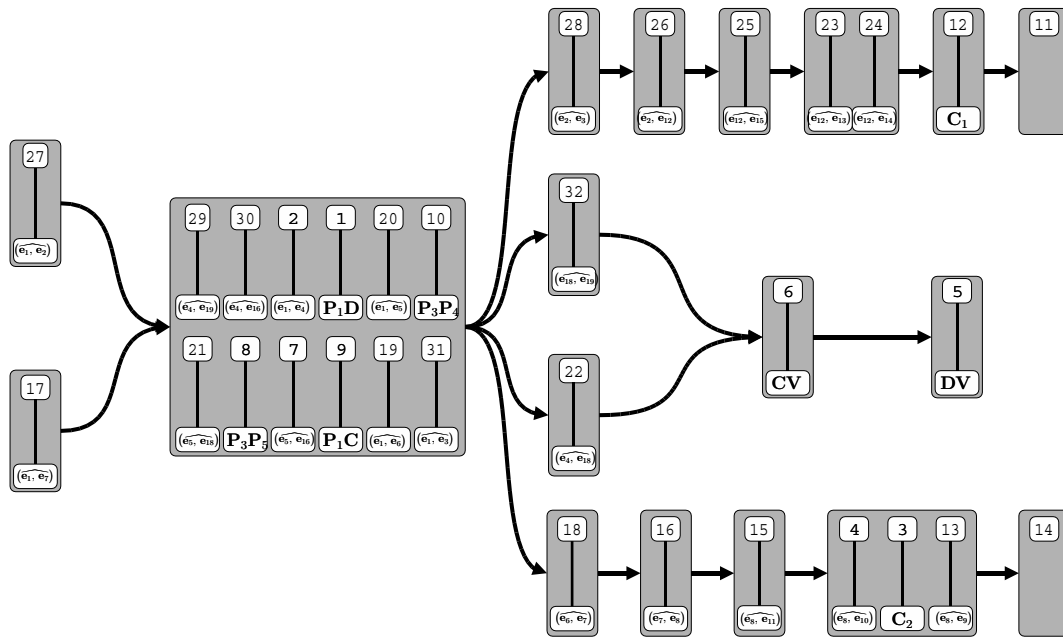


FIG. 13.10: Plan de résolution du problème de la figure 13.2.

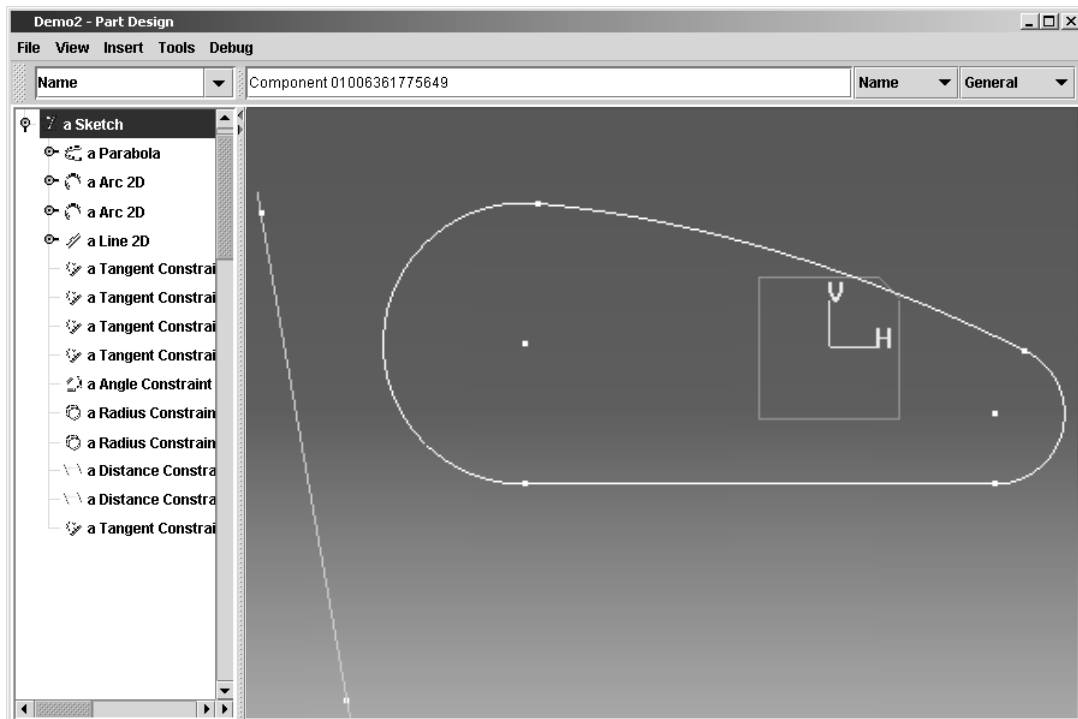


FIG. 13.11: Une solution du problème de la figure 13.6

13.2.3 Un problème d'ingénierie

La solution proposée dans ce manuscrit est parfaitement adaptée à la spécification de contraintes d'ingénierie. Afin d'illustrer ce point, un véritable exemple mécanique est présenté. L'objectif de cet exemple est de définir la section du pied d'une pale d'hélice (Figure 13.12-b). Les pales, en rotation, sont soumises à une force centrifuge importante. Les pressions qui s'exercent sur les différentes surfaces de contact peuvent alors devenir très élevées. La prise en compte de ces pressions est donc nécessaire lors de son dimensionnement.

Afin de simplifier le problème, nous ne nous occuperons que de la partie inférieure de la section médiane du pied de pale (Figure 13.12-c). La figure 13.12-d fait un bilan simplifié des forces s'exerçant sur le pied de pale en régime constant. Les deux forces \mathbf{F}_1 et \mathbf{F}_2 s'opposent à la force centrifuge F_c . Ce n'est pas l'intensité de ces forces qui est importante mais la pression de contact au niveau de leurs points d'application. Celle-ci doit être inférieure à la pression maximale admissible par les matériaux composant soit le pied de pale, soit son support.

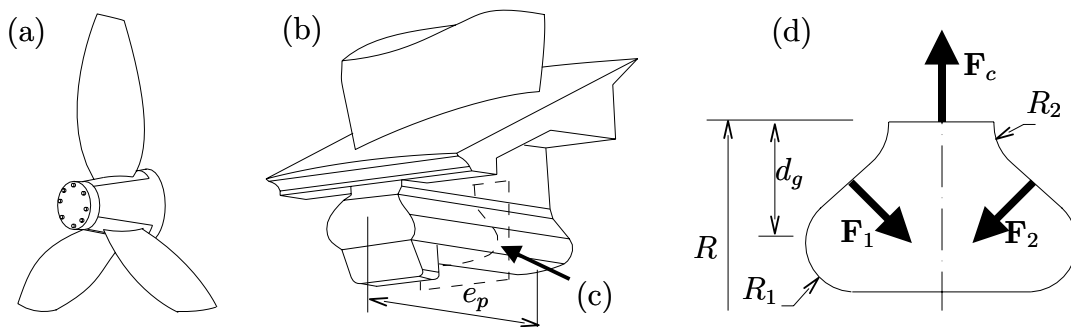


FIG. 13.12: (a) Une hélice, (b) le pied d'une pale, (c) section à dimensionner

Ce pied de pale est aussi soumis à un ensemble de contraintes technologiques déjà traduit en contraintes géométriques. La figure 13.13 représente la demi section du pied de pale sur laquelle l'ensemble des spécifications dimensionnelles est appliqué. La structure de ce pied de pale est représentée à la figure 13.14. Afin de la simplifier, les arcs de cercles ne sont définis que par une courbe de Bézier rationnelle au lieu de trois.

Afin de compléter son modèle, l'utilisateur spécifie l'ensemble des contraintes d'ingénierie à l'aide du module spécifique (Figure 13.15). Pour cela, il définit dans un premier temps l'ensemble des variables d'ingénierie du problème, à savoir :

- **epaisseur** : l'épaisseur du pied de pale par rapport à la section (e_p) ;
- **rt** : rayon du support des pales (R) ;
- **rho** : masse volumique du matériau du pied de pale (ρ) ;
- **fcp** : force centrifuge de la pale sans son pied (F_{cp}) ;
- **w** : vitesse de rotation des pales (ω) ;
- **sig** : pression maximale admissible par les matériaux (σ) ;

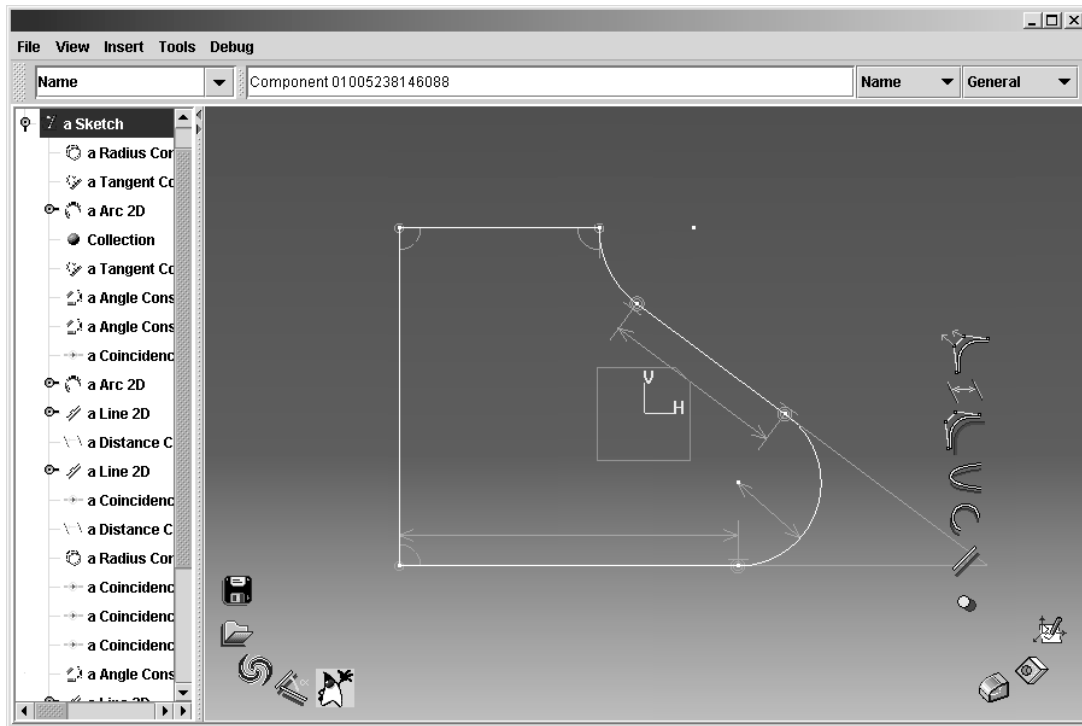


FIG. 13.13: Spécification géométrique de la section (c) du pied de pale

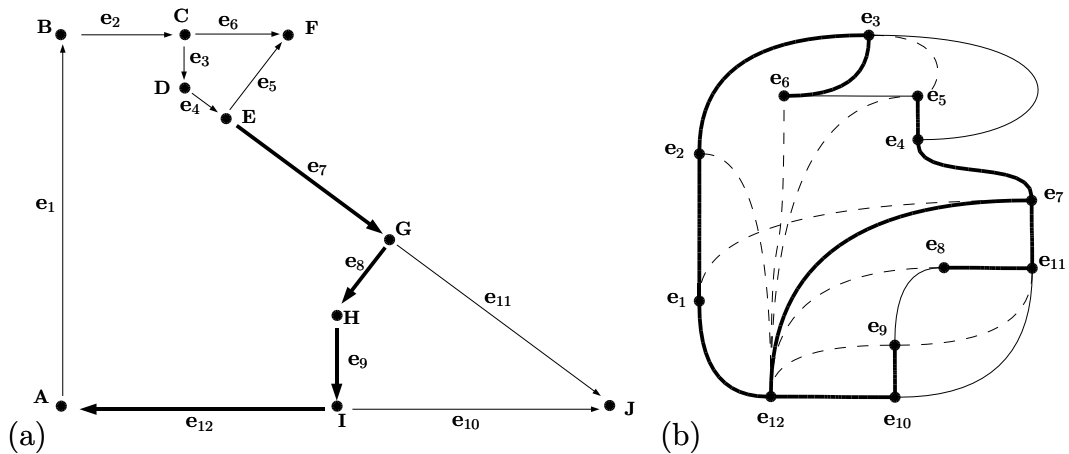


FIG. 13.14: Structure du pied de pale

- **surf** : surface de la section du pied de pale à dimensionner (S) ;
- **m** : masse du volume à dimensionner (m) ;
- **fc_b** : force centrifuge générée par la partie à dimensionner (F_{cb}) ;
- **dg** : distance entre la surface extérieure du support des pales et le centre de gravité du pied de pale (d_g).

Puis, il définit l'ensemble des relations d'ingénierie :

- Ing. 1 : relation entre la force centrifuge du pied de pale, la masse de ce pied et sa vitesse de rotation :

$$F_{cb} - m\omega^2 (R - d_g) = 0 ; \quad (13.3)$$

- Ing. 2 : relation entre la masse, la masse volumique, l'épaisseur et la surface du pied de pale :

$$m - \rho e_p S = 0 ; \quad (13.4)$$

- Ing. 3 : équilibre dynamique en régime établi, c'est à dire l'équilibre entre les forces \mathbf{F}_1 , \mathbf{F}_2 et la force centrifuge F_c :

$$2\mathbf{EG} \cos(\widehat{\mathbf{e}_1, \mathbf{e}_7}) \sigma e_p + (F_{cb} + F_{cp}) = 0. \quad (13.5)$$

À ces relations, il faut ajouter les relations entre la surface (Ing. 4) ou la position du centre de gravité (Ing. 5) avec les paramètres géométriques. Ces deux variables étant évaluées par des "boîtes noires", deux relations génériques sont ajoutées au système :

$$S - f_1(\mathbf{AB}, \mathbf{BC}, \dots, (\widehat{\mathbf{e}_1, \mathbf{e}_2}), (\widehat{\mathbf{e}_1, \mathbf{e}_{12}}) \dots) = 0 ; \quad (13.6)$$

$$d_g - f_2(\mathbf{AB}, \mathbf{BC}, \dots, (\widehat{\mathbf{e}_1, \mathbf{e}_2}), (\widehat{\mathbf{e}_1, \mathbf{e}_{12}}) \dots) = 0. \quad (13.7)$$

Dans ce problème, certains paramètres d'ingénierie ont une valeur imposée :

- **e_p** et **rt** : le support des pales est imposé ;
- **rho** et **sig** : le matériau des pales est déjà choisi. La masse volumique ρ et la pression admissible du matériau σ sont donc connues ;
- **w** : la vitesse de rotation de l'hélice fait partie du cahier des charges ;
- **fc_p** : la forme des pales a été préalablement conçue, la force centrifuge de celles-ci est donc connue.

Le problème consiste donc, connaissant l'ensemble des spécifications géométriques et d'ingénierie, à calculer le rayon R_2 du second arc de cercle.

Il est intéressant de rappeler que, contrairement aux mises en équations géométriques classiques (Cf. sous-section 3.2.1), les paramètres géométriques de l'objet sont directement accessibles. Par exemple, le paramètre **EG** utilisé par l'équation (13.5) fait partie de la structure. Son utilisation ne nécessite donc pas la création d'une relation supplémentaire définie dans un repère global tel :

$$(x_E - x_G)^2 + (y_E - y_G)^2 + (z_E - z_G)^2 = \mathbf{EG}^2.$$

```

File Edit
Scripting Language BeanShell
/* Variables */
epaisseur = new Variable(100, "ep"); /* epaisseur du bulbe */
rt = new Variable(380, "rt"); /* rayon turbine */
rho = new Variable(0.00000455, "rho"); /* masse volumique */
fcp = new Variable(150000000, "fcp"); /* force centrifuge pale*/
w = new Variable(200, "w"); /* vitesse rotation */
sigma = new Variable(26867.194, "sig"); /* coef sigma */
surface = new Variable(13515.3291, "surf"); /* surface */
masse = new Variable(6.149, "m"); /* masse */
fcb = new Variable(78707200, "fcb"); /* force centrifuge bulbe*/
dg = new Variable(60, "dg"); /* distance de gravité */

/* Equations */
/* Force centrifuge du bulbe */
equationFc = "(v|fcb| - (v|m| x v|w| x v|w| x (v|rt| - v|dg|))";

/* Masse : */
equationM = "(v|m| - (v|rho| x v|ep| x v|surf|)";

/* Équilibre dynamique :*/
equationEd = "(%2 x v|sig| x v|p10-p11| x v|ep| x cos(v|e0-e7|)) + v|fcb| + v|fcp|";

/* Distance gravité */
equationDg = "(((v|p13-p12| x sin(v|e8-e9|) / %2 x v|s6|) + ((v|p13-p12| x sin(v|e8-e9|)) +
(v|p10-p11| x sin(v|e0-e7|) / %2)) x v|s5|) + ((v|p22-p23| - v|p3-p2|) x (v|s1| +
v|s2| + v|s3| + v|s4|)) / v|surf|) - v|dg|";

Run Stop

```

FIG. 13.15: Module de définition des spécifications d'ingénierie

À partir de l'ensemble des relations géométriques et d'ingénierie, l'analyse pré-résolution et la génération du plan de résolution peuvent être réalisées. Le plan de résolution ainsi obtenu est représenté à la figure 13.16. Dans ce plan, les relations 1-2, 3-4, et 5-6 sont associées respectivement aux cycles :

- C, F, E, D ;
- A, B, C, F, E, G, H, I ;
- G, H, I, J.

Nous remarquons que dans ce problème, l'ensemble des relations angulaires (7 à 18) sont à résoudre en parallèle, dès la première étape. La seconde étape permet de calculer les derniers éléments du premier arc de cercle ($\mathbf{GJ} = \mathbf{IJ}$) et surtout de calculer la force centrifuge admissible pour la partie à dimensionner (F_{cb}) à partir de la relation d'ingénierie Ing. 5. Il reste ensuite à identifier les valeurs admissibles de la masse m , de la surface S et de la position du centre de gravité d_g fortement liées aux paramètres géométriques \mathbf{AB} , \mathbf{BC} et R_2 . Le plan se termine ensuite par la résolution de la relation 1 qui doit être vérifiée par la relation 2 pour connaître les valeurs des derniers éléments de la structure du second arc de cercle ($\mathbf{CD} = \mathbf{DE}$).

13.2.4 La spécification de courbes planes

Les problèmes comportant des courbes quelconques sont, généralement, fortement sous-contraints. La sous-section 11.2.8 a présenté l'utilisation d'un modèle

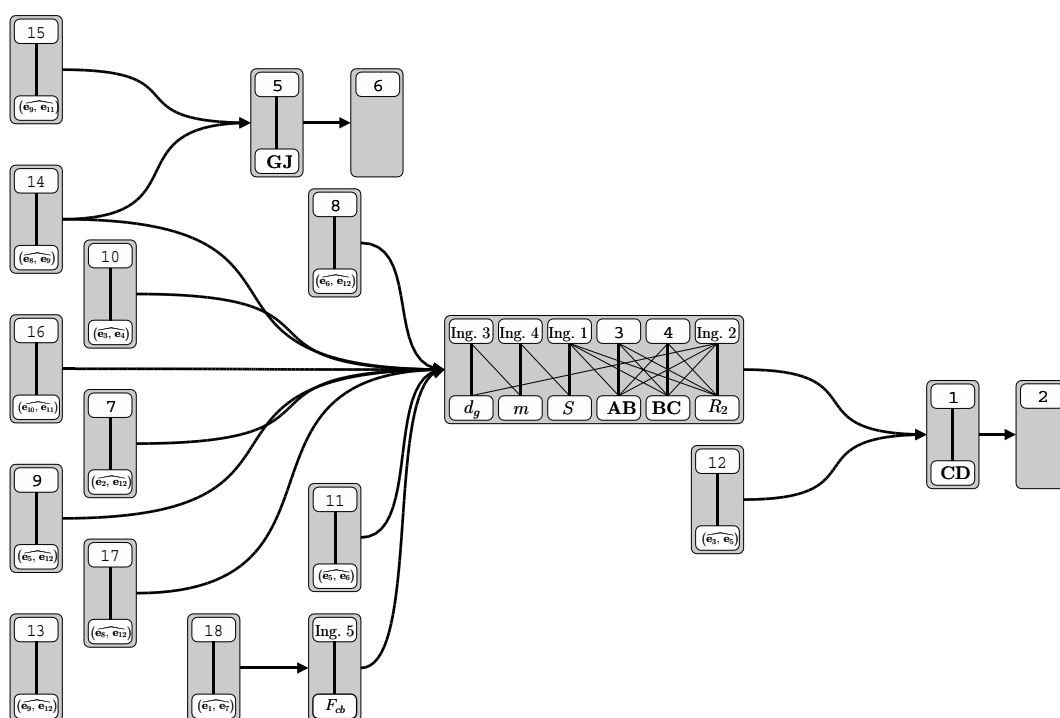


FIG. 13.16: Plan de résolution du problème d'ingénierie

mécanique afin de donner à ces courbes des comportements naturels. La figure 13.17 présente un tel problème. Une courbe de Bézier dont les deux extrémités coïncident avec celles d'une polygone bien contrainte a un point courant partiellement spécifié. En effet, seule sa position par rapport au premier segment de la polygone est imposée. Ce point a donc un degré de liberté par rapport à la polygone et un second par rapport à la courbe : la coordonnée paramétrique de ce point de la courbe n'est a priori pas imposée.

La structure d'un tel problème est représentée à la figure 13.18. On reconnaît la structure de la contrainte de distance entre une courbe et une droite (Cf. sous-section 7.7.3). À partir de cette structure, un ensemble de relations géométriques est identifié. Les quatre premières sont classiques. Elles sont associées aux cycles vectoriels :

- **A, B, C, D, E** ;
- **A, E, G**.

En revanche, le cycle **A, F, G** possède le multi-point $\widetilde{\mathbf{AG}}$. Les relations obtenues à partir de ce cycle sont donc issues de la projection de la somme vectorielle (13.8) qui se décompose en la relation (13.9) :

$$\widetilde{\mathbf{AG}} + \mathbf{GF} + \mathbf{FA} = 0 ; \quad (13.8)$$

$$\mathbf{AB}f_{1,3}(u) + \mathbf{BC}f_{2,3}(u) + \mathbf{CD}f_{3,3}(u) + \mathbf{GF} + \mathbf{FA} = 0 \quad (13.9)$$

Les dernières sont les relations de Chasles issues du graphe vectoriel (Figure 13.18-

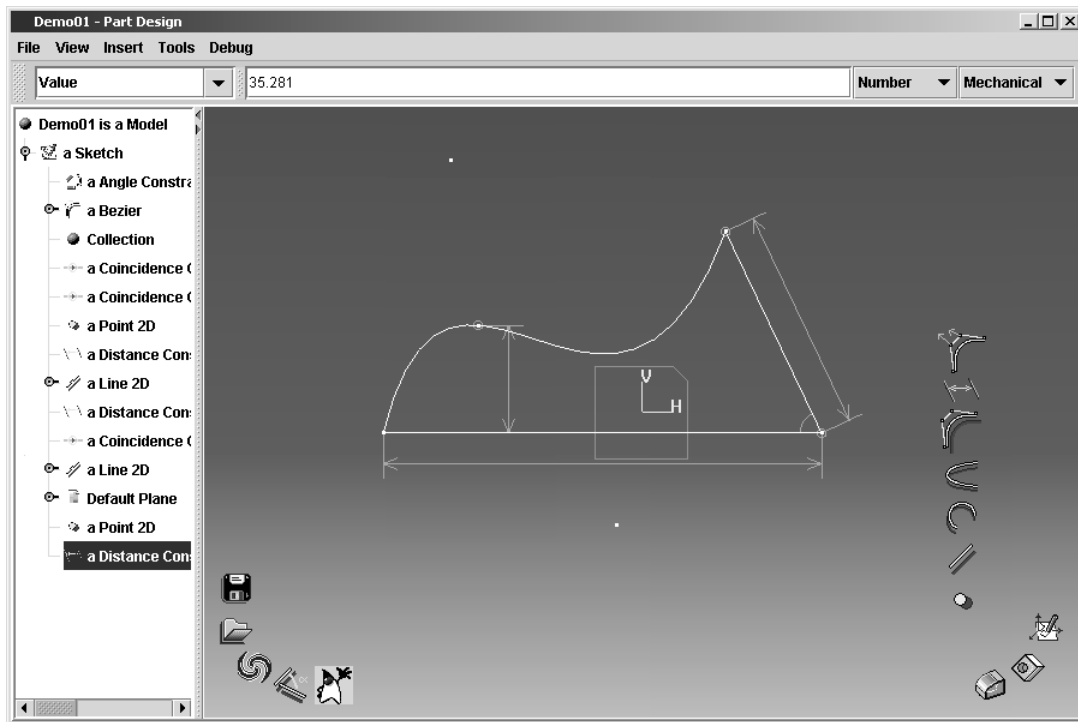


FIG. 13.17: Une courbe plane dont un point courant est partiellement spécifié

b).

Afin de donner un comportement naturel à la courbe, un ensemble de relations mécaniques supplémentaires est ajouté au système (Cf. sous-section 11.2.8) :

$$\mathbf{F}_0 + \mathbf{AB}q_1 = 0 ; \quad (13.10)$$

$$\mathbf{F}_1 + \mathbf{BA}q_1 + \mathbf{BC}q_2 = 0 ; \quad (13.11)$$

$$\mathbf{F}_2 + \mathbf{CB}q_2 + \mathbf{CD}q_3 = 0 ; \quad (13.12)$$

$$\mathbf{F}_3 + \mathbf{DC}q_3 = 0, \quad (13.13)$$

avec $q_{i \in \{1,2,3\}}$ les densités de force de chaque barre du réseau, $\mathbf{F}_{i \in \{1,2,3\}}$ les vecteurs forces appliqués aux sommets des pôles du polygone caractéristique de la courbe. Sans ces relations, ce problème est résolvable soit classiquement en fixant arbitrairement un ensemble de paramètres, soit en utilisant un algorithmes d'optimisation (Cf. sous-section 11.2.7). Mais par ces approches, le résultat, tout en vérifiant l'ensemble des spécifications, est rarement satisfaisant d'un point de vue esthétique. Par ces équations le polygone caractéristique a un comportement de câble pesant sous tension. Par conséquent, la courbe est associée à un comportement qui semble naturel pour l'utilisateur.

Pour obtenir des relations scalaires à partir de ces relations vectorielles, il suffit de projeter chaque relation sur la force associée. Par exemple, la projection

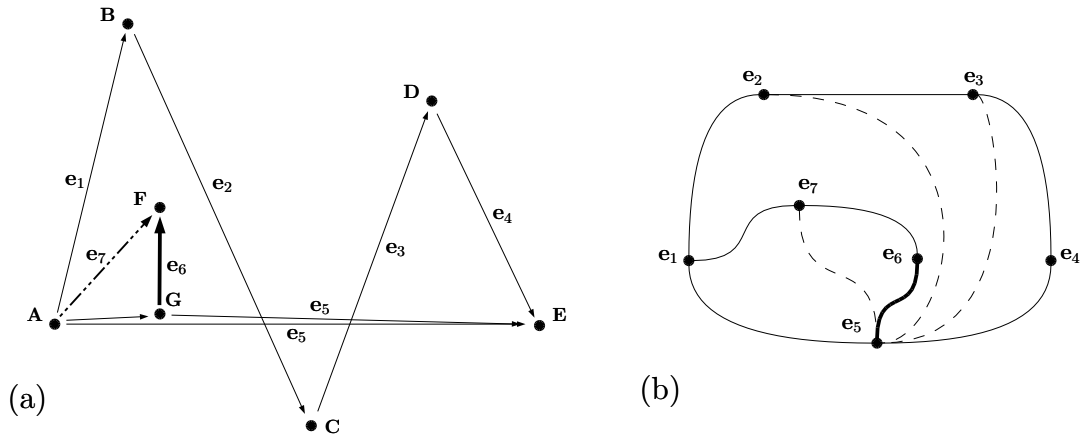


FIG. 13.18: Structure du problème de la figure 13.17

de la relation (13.11) donne le système :

$$\begin{cases} \mathbf{F}_1 + \mathbf{BA}q_1 \cos(\alpha_1) + \mathbf{BC}q_2 \cos(\alpha_1 + \widehat{(\mathbf{e}_1, \mathbf{e}_2)}) = 0 ; \\ \mathbf{BA}q_1 \sin(\alpha_1) + \mathbf{BC}q_2 \sin(\alpha_1 + \widehat{(\mathbf{e}_1, \mathbf{e}_2)}) = 0, \end{cases} \quad (13.14)$$

avec α_1 l'angle $\widehat{(\mathbf{e}_1, \mathbf{F}_1)}$.

La résolution de ce problème consiste à trouver le minimum d'une fonctionnelle de la FdM tout en vérifiant le système d'équation (Cf. sous-section 11.2.8). Dans un problème plus important, l'analyse structurelle permet d'isoler la composante sous-contrainte à résoudre par cette méthode. La figure 13.19 présente le résultat obtenu en minimisant la variation des forces par rapport à leur intensité initiale.

13.2.5 Correction du plan de résolution

Le chapitre 10 a présenté la construction du plan de résolution. Cette étape est suivie d'une analyse qualitative (Cf. section 10.3). Nous avons vu qu'elle permet de vérifier que chaque PPSI est bien contraint, c'est-à-dire qu'il est composé de n équations non-dégénérées au voisinage des solutions. Dans le cas contraire, elle identifie le ou les équations redondantes (par rapport aux premières). Cet exemple va illustrer l'étape de correction du plan de la figure 13.20.

Dans ce plan, l'analyse du PPSI de six équations a permis de dire que l'équation 19 est redondante (par rapport au cinq autres). Cette équation est extraite du PPSI et devient une équation sur-contrainante (Figure 13.21).

La seconde étape consiste à extraire du graphe complet, le plus petit sous-graphe impacté par la suppression du noeud de l'équation 19. Pour cela, la première étape consiste à trouver une équation initialement sur-contrainante

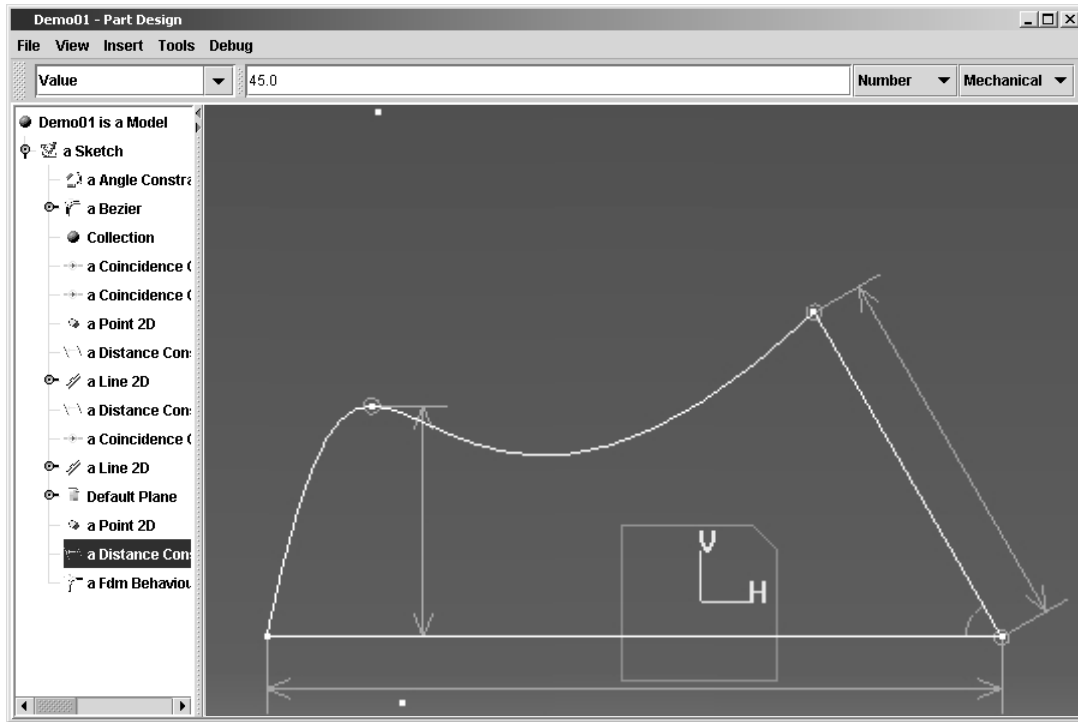


FIG. 13.19: Résultat obtenu par la minimisation de la variation des forces de la FdM

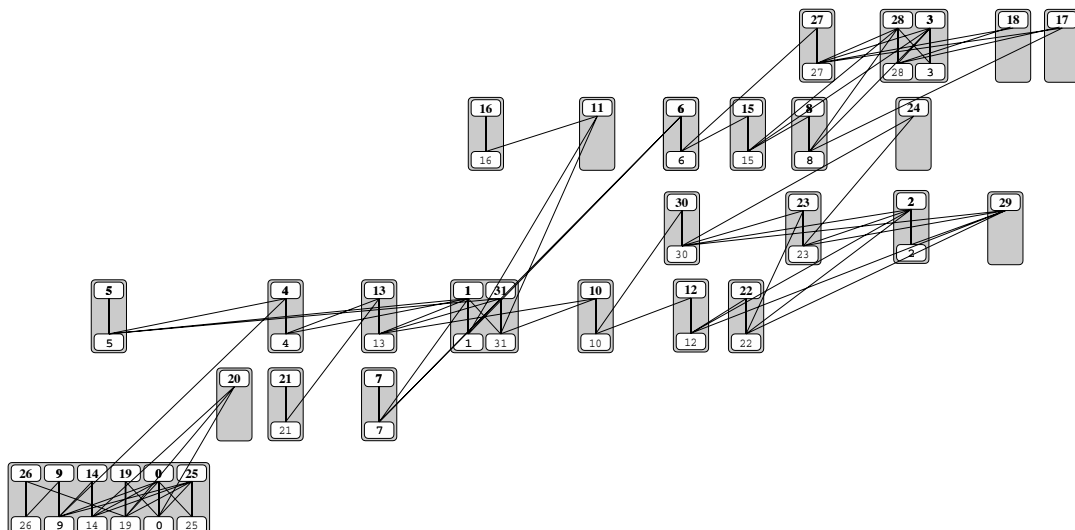


FIG. 13.20: Un plan de résolution

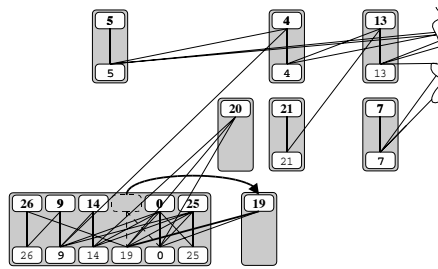


FIG. 13.21: Identification d'une équation redondante

qui pourrait rendre le couplage maximal. Dans l'exemple, l'équation 11 est sélectionnée. Le plus petit sous-graphe est mis en évidence à la figure 13.22.

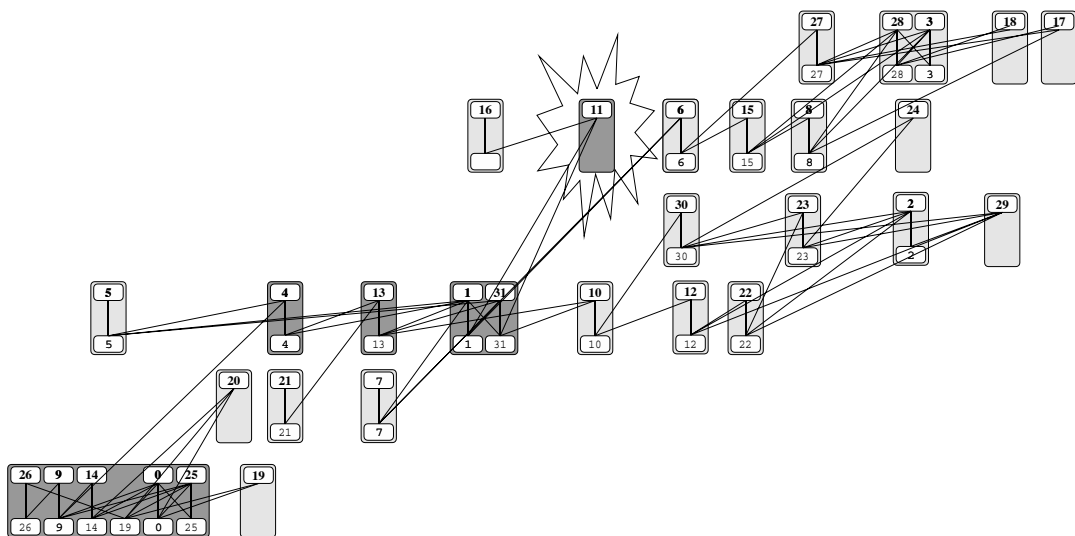


FIG. 13.22: Recherche du plus petit sous-graphe

Dans ce sous-graphe, l'analyse structurale (Cf. section 10.2) est réalisée. La recherche d'un couplage maximal est simplifiée. En effet, un couplage partiel existe déjà. Il suffit alors de rechercher des chaînes alternées augmentantes (Cf. annexe C.6.1). La figure 13.23 illustre cette recherche. Le sous-plan de résolution peut alors être défini (Figure 13.24).

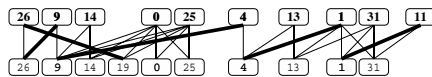


FIG. 13.23: Recherche des chaînes alternées augmentantes

La dernière étape consiste simplement à réinsérer le sous-graphe analysé dans le plan de résolution afin de mettre à jour le plan global (Figure 13.25).

On a pu voir sur cet exemple que le sous-graphe impacté est relativement réduit. La stratégie consistant à n'étudier que la zone modifiée est donc perti-

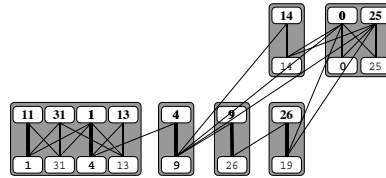


FIG. 13.24: Sous plan de résolution

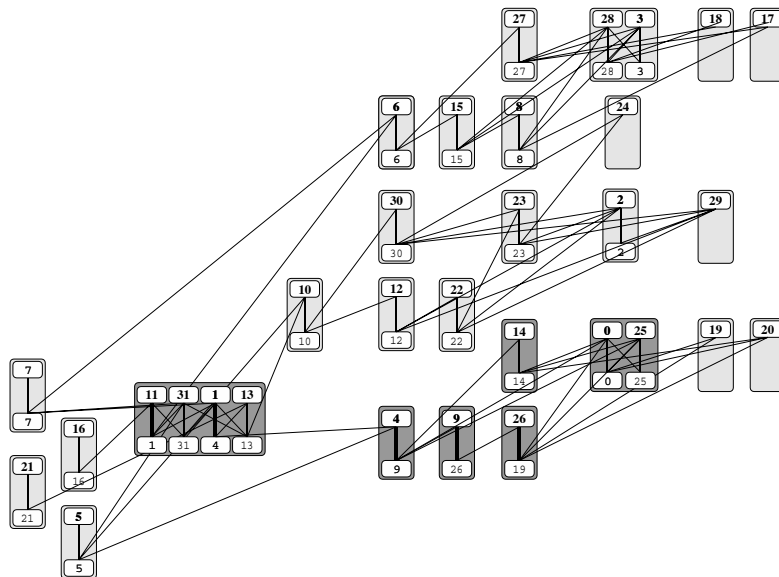


FIG. 13.25: Plan de résolution corrigé

nente. Ceci sera d'autant plus vrai sur de véritables problèmes industriels comme celui présenté ci-dessous.

13.2.6 Un exemple "industriel"

Avant de conclure, un exemple qui commence à être conséquent est présenté (Figure 13.26). Il est assez représentatif des pièces estampées de l'industrie.

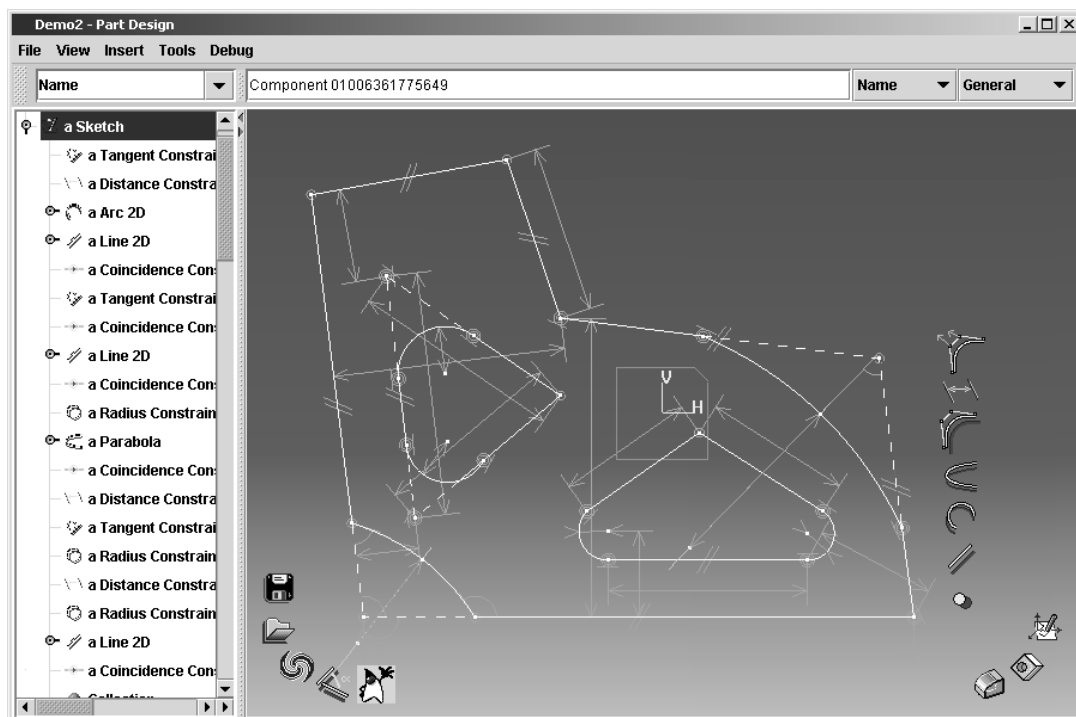


FIG. 13.26: Définition d'une pièce estampée.

Pour cet exemple, le plan de résolution obtenu par l'application est représenté à la figure 13.27. En raison de la taille du plan, seul le nombre de relations est indiqué dans chaque PPSI. On remarque que pour cet exemple, la stratégie choisie pour générer les équations est bonne. En effet, le nombre maximum de relations dans un PPSI est de deux. Une solution à ce problème est illustrée à la figure 13.28.

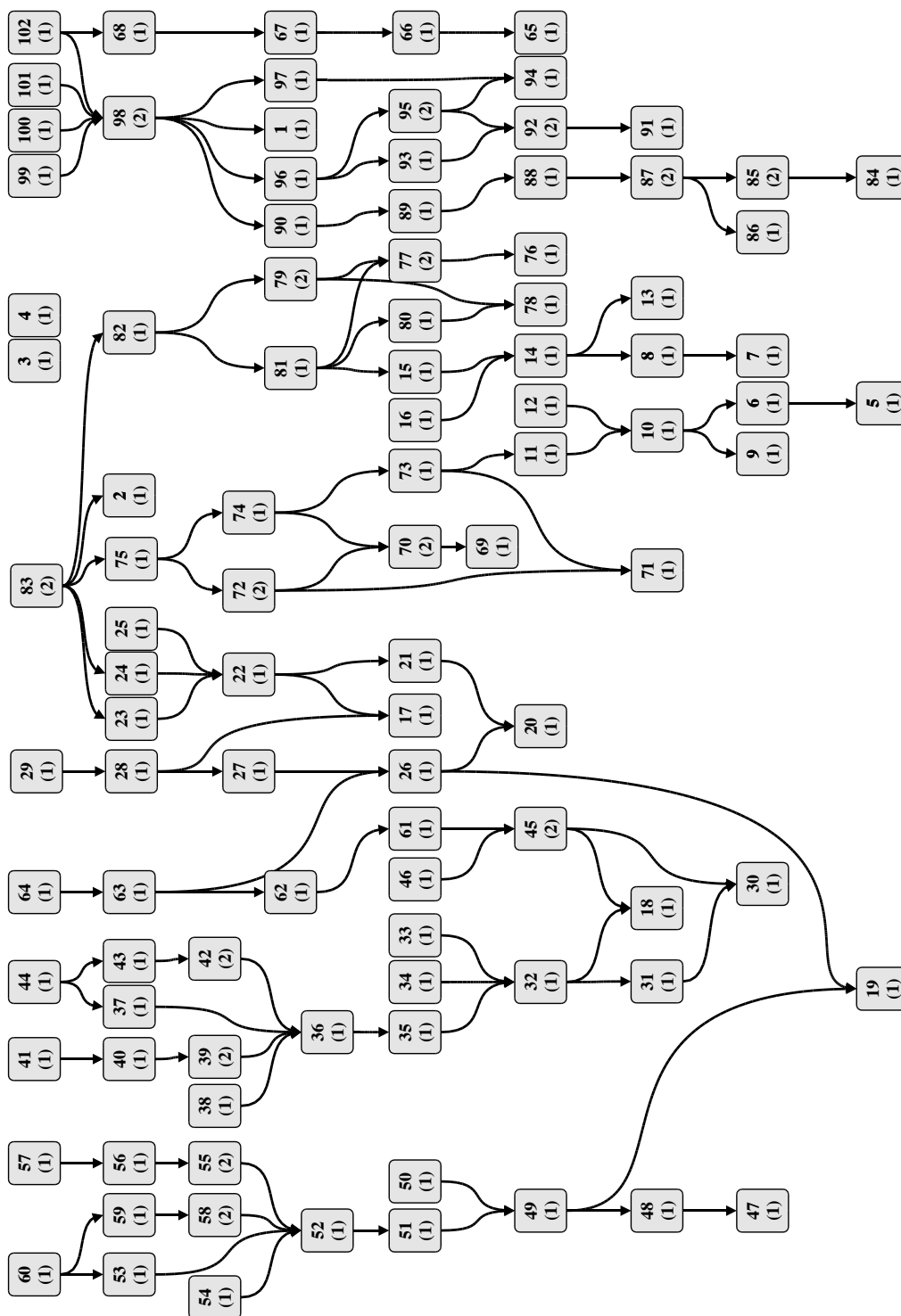


FIG. 13.27: Plan de résolution du problème de la pièce estampée

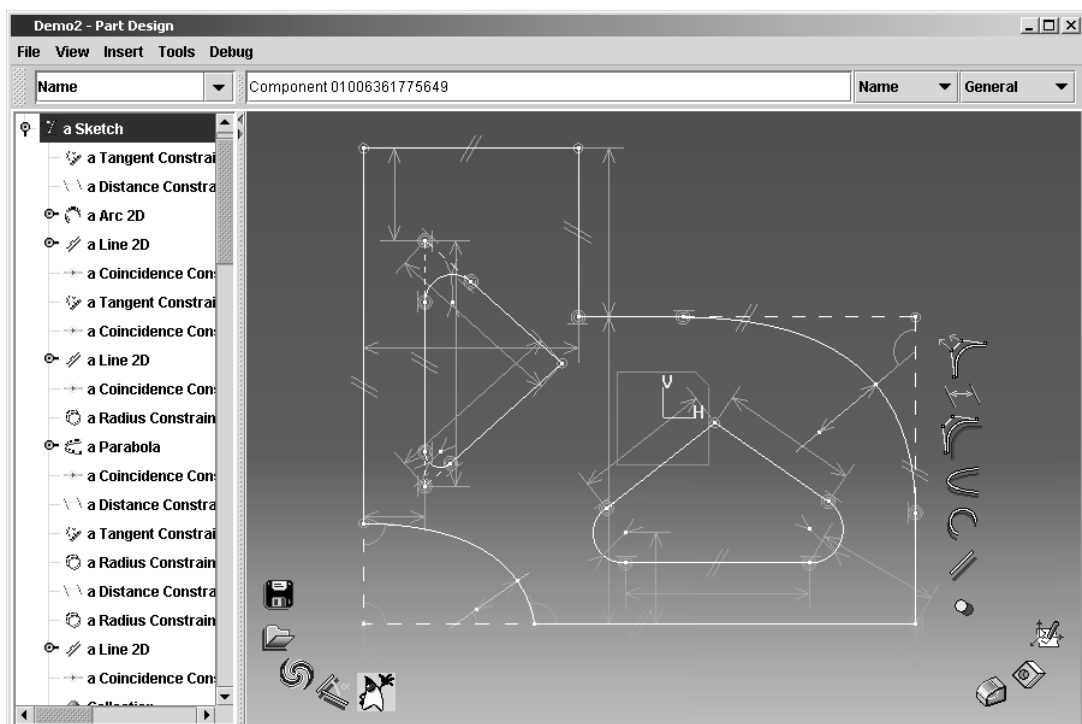


FIG. 13.28: Résultat du problème de la figure 13.26.

Synthèse, perspectives et conclusion

Synthèse

La problématique initiale de cette thèse était de définir un solveur géométrique 3D permettant de traiter les problèmes d'ingénierie. Les premières réflexions ont rapidement mis en évidence que l'architecture actuelle des applications de CAO n'est pas satisfaisante pour pouvoir simplement "brancher" un solveur géométrique 3D afin de mettre en place une approche variationnelle. Les travaux réalisés ont permis d'en proposer une nouvelle, basée autour d'un *modèle dynamique de spécifications d'ingénierie* (Cf. section 4.3). Ce modèle est divisé en trois niveaux de représentation :

- le premier regroupe l'ensemble des spécifications telles qu'elles ont été initialement formulées. C'est donc la partie publique du modèle, enrichie et modifiée par l'utilisateur ;
- le second contient une description originale de la partie géométrique des spécifications. Il distingue la situation des éléments obtenue par un positionnement relatif, de la description des différents éléments définissant la *peau* de l'objet. Par analogie avec une structure à toile tendue, l'ensemble des informations dimensionnelles représente la *structure* de l'objet et la *toile* regroupe l'ensemble des constituants structurés dans un modèle B-rep. Ce découpage a permis de séparer la partie "géométrie et topologie" de la partie dimensionnelle en raison de leur relative indépendance et afin de bien distinguer les différents domaines traités ;
- le troisième est une reformulation algébrique de l'ensemble des spécifications. Il est utilisé puisque c'est le modèle le plus adapté au maintien et à l'analyse de la cohérence d'un ensemble des spécifications de nature différente.

L'analyse et le maintien de la cohérence de ce modèle sont délégués à un solveur pour l'identification des valeurs admissibles par les différents paramètres du problème et à un modeleur pour la définition d'un modèle géométrique topologiquement cohérent.

La description générale du modèle géométrique a permis de préciser son originalité dans le positionnement relatif des différents constituants (Cf. chapitre 5), mais l'intérêt de ce positionnement n'a été exposé qu'au cours du chapitre 8. La structure et la toile des différents éléments et des contraintes géométriques ont été introduits respectivement aux chapitres 6 et 7. Ces présentations préalables étaient nécessaires pour montrer la cohérence de la structure de données proposée. Dès ces présentations, l'intérêt de ce modèle a commencé à transparaître. En effet, par l'intermédiaire de la structure, des objets complexes sont définis avec l'ensemble de leurs caractéristiques (Cf. section 6.4). De plus, les contraintes géométriques sont représentées avec le même formalisme (Cf. chapitre 6). L'analyse et le maintien de la cohérence dimensionnelle d'un objet géométrique peuvent donc être réalisés à partir d'un ensemble uniforme de données.

Suite à la présentation du modèle algébrique (Cf. section 8.2), l'intérêt du modèle géométrique a été confirmé : il permet une reformulation originale des spécifications géométriques pour le modèle algébrique (Cf. section 8.4). L'intérêt de cette reformulation est que le modèle algébrique ainsi obtenu est propice à une analyse et une résolution basées sur une décomposition du système en PPSI. Au contraire, les approches habituelles (Cf. section 3.2) sont généralement basées sur une reformulation dans un repère cartésien unique générant un système d'équations trop imbriquées. Une telle approche n'est alors pas envisageable.

La présentation de ce modèle a permis de clôturer la description du modèle dynamique. La troisième partie de ce manuscrit a porté sur la description d'un solveur d'ingénierie. Il n'a plus de raison d'être un solveur géométrique puisque sa tâche consiste à résoudre un système d'équations dans lequel l'aspect géométrique a disparu. En revanche, son rôle est passé de simple solveur à celui d'un module complet de résolution disposant d'outils d'analyse, de solveurs multiples et de diagnostics. Les possibilités ainsi offertes par ce module sont dues en grande partie à la première étape de la préparation du plan de résolution, c'est-à-dire à la décomposition du problème en PPSI. Sans une décomposition importante, les possibilités de ce module deviennent extrêmement réduites. La qualité du système initial a donc une très grande importance d'où le rôle majeur accordé à la structure dans le modèle géométrique et à la stratégie utilisée lors de sa reformulation dans le modèle algébrique. Le plan de résolution obtenu joue donc un rôle majeur dans ce module. En effet, plus que la rapidité de la résolution, il apporte la robustesse, la multiplicité des solutions et les capacités d'analyse que l'on ne retrouve pas dans les solveurs numériques classiques.

Le module de résolution et la structure du modèle géométrique ont aussi permis de proposer de nouvelles familles de spécifications comme les contraintes de liberté ou les spécifications facultatives.

La définition du modèleur géométrique, malgré l'importance de son rôle dans une application variationnelle et la richesse de sa problématique, n'a pas encore

été traitée.

Le chapitre 13 a permis de présenter la validation d'une grande partie des propositions qui ont été réalisées à partir d'une maquette informatique. Quelques exemples caractéristiques y ont été détaillés. Une grande partie ne signifiant pas la totalité, de nombreuses perspectives d'étude sont envisageables. Elles sont présentées à la section suivante.

Perspectives d'étude

Au terme de ce travail, les perspectives envisagées sont nombreuses. Certains aspects abordés méritent d'être complétés, d'autres ouvrent la voie à plusieurs directions de recherche.

Les perspectives à court terme

Un premier ensemble de perspectives dites à court terme est proposé. Leur particularité est qu'elles ne devraient pas nécessiter une grande quantité de travail. En effet, elles sont basées sur des propriétés démontrées et sur des algorithmes dont les points délicats ont déjà été testés.

La recherche de plusieurs solutions

Contrairement aux problèmes paramétriques, dans une approche variationnelle, les problèmes admettent plusieurs solutions. Pour des raisons évidentes, la première solution proposée n'a aucune raison d'être la solution attendue par l'utilisateur, sans parler des solutions refusées par le modeleur à cause d'incohérences topologiques. De plus, le parcours des différentes solutions peut amener l'utilisateur à des choix qu'il n'aurait pas imaginés.

Ce problème a déjà été traité dans le manuscrit, un ensemble de solutions a été proposé. Une partie a été validée mais il reste tout de même quelques points sombres. Par exemple, l'utilisation de l'homotopie dans le corps des complexes est une solution démontrée pour les systèmes polynômiaux, mais doit encore être testée pour les systèmes obtenus.

La mise en place d'une gestion dynamique des données

La solution proposée est destinée à une application interactive. L'utilisateur construit progressivement un objet par un enrichissement constant de l'ensemble des spécifications. Il attend donc d'une telle application, le maintien permanent de la cohérence de la description.

Jusqu'à présent, les solutions variationnelles ne sont présentes que dans des modules 2D. Les problèmes à traiter restent donc relativement petits. En effet,

ne sont traités par ces approches que les contours des primitives utilisées par un modéleur paramétré, alors que la généralisation de l'approche variationnelle au 3D va permettre de contraindre "d'un seul bloc" l'ensemble du produit en cours de conception. Pour cela, il est primordial de développer une gestion dynamique de l'ensemble des données. En effet, il n'est pas concevable de devoir vérifier l'ensemble des spécifications d'une voiture lorsque seul le matériau de l'arbre de transmission est modifié.

La mise en place de cette gestion dynamique a été partiellement réalisée. La plupart des algorithmes nécessaires ont été écrits. Il reste maintenant à réaliser l'assemblage des différents composants.

La génération de diagnostics

Le module de résolution présenté ne fait pas que résoudre les systèmes d'équations. Afin d'encadrer l'utilisateur tout au long du processus de conception, il propose un ensemble d'analyses afin de comprendre l'état de contrainte d'un problème et/ou les causes d'un échec de résolution. Le retour d'informations vers l'utilisateur peut-être réalisé par des diagnostics qui ont fait l'objet du chapitre 12. Ces principes restent à être mis en place.

La structure des contraintes 3D et la génération des équations

Les propositions faites pour le 3D sont très prometteuses pour plusieurs raisons :

- la première est la parfaite adéquation de la structure de données pour les objets 3D par rapport à celle initialement proposée pour les objets 2D. En effet, tout comme le 2D, une structure d'un objet 3D est composée des éléments élémentaires de l'espace vectoriel : le **vecteur** et le **plan vectoriel** et de ceux de l'espace affine : le **spoint**, le **bipoint**, et le **plan affine** ;
- la seconde est la simplicité des structures des contraintes 3D. En effet, tout comme pour le 2D, l'ensemble des contraintes dimensionnelles élémentaires est formulé directement dans la structure ;
- la troisième résulte des possibilités offertes par la trigonométrie sphérique. En effet, grâce à ces relations, le système obtenu par la transcription de l'ensemble des spécifications du modèle géométrique au modèle algébrique reste local.

La validation de ces propositions n'a pas pu être réalisée. Une première perspective de ce travail de thèse est donc toute tracée. Malgré toutes les propositions, cette tâche reste ardue. En effet, comme pour le 2D, une bonne génération des relations passe par la proposition d'un ensemble d'heuristiques. Or, pour le 3D, ces heuristiques risquent d'avoir beaucoup plus d'importance que pour le 2D.

De plus, afin d'améliorer la fiabilité de l'approche, elles devront évoluer en un ensemble de propriétés rigoureuses.

Le modelleur géométrique

Le processus de conception consiste à décrire un objet par un ensemble de spécifications. La problématique de ce manuscrit a porté principalement sur le positionnement des constituants du contour de l'objet. Or, nous avons vu que dans une approche B-rep (Cf. sous-section 1.3.4), ces constituants sont liés entre eux par des relations topologiques.

L'objet initialement défini a une topologie particulière représentant généralement un volume. La mise à jour de la position de l'ensemble des constituants en fonction des spécifications a pu détruire la cohérence topologique du modèle géométrique.

Le rôle du modelleur (Cf. sous-section 4.3.4) est alors de proposer le (les) nouveau(x) modèle(s) topologiquement cohérents.

Cette problématique très riche est relativement nouvelle et très peu de solutions ont été proposées. Or, ce module est nécessaire à la mise en place d'un environnement variationnel tel qu'il a été décrit dans ce manuscrit.

La définition de la structure, des contraintes géométriques et des agents de résolution spécifiques aux surfaces gauches

Les travaux présentés ne proposent qu'un ensemble restreint d'éléments et de contraintes afin de concevoir une pièce ou plus généralement un mécanisme. Bien qu'en 2D l'utilisation des courbes NURBS soit proposée, en 3D, la seule entité surfacique étudiée dans ce mémoire est le plan. Or, la plupart des pièces mécaniques comportent de nombreuses surfaces gauches.

Les surfaces gauches élémentaires sont les surfaces quadriques. La structure de ces surfaces devrait être aisément définie par la généralisation de la structure des coniques, tout comme les contraintes appliquées à ces surfaces.

Cependant, ces surfaces élémentaires ne sont en pratique que peu utilisées. En effet, malgré l'utilisation courantes de certaines d'entre elles, comme les surfaces cylindriques et sphériques pour la définition des congés de raccordement, les autres surfaces quadriques ne sont utilisées pour leur propriétés que dans des applications très particulières. Le reste du temps, les surfaces gauches sont quelconques et sont surtout utilisées pour des raisons esthétiques. Il est donc nécessaire de leur définir une structure. Tout comme les courbes, les surfaces sont difficilement spécifiables. Malgré tout, l'utilisateur peut vouloir leur imposer un ensemble de contraintes comme le propose J.-P. Pernot [Pernot *et al.* 2002], [Pernot *et al.* 2002]. Celle-ci doivent donc être définies, tout comme un agent de

résolution spécifique doit leur être associé.

Conclusion

Ce manuscrit décrit en premier lieu un solveur géométrique qui a vocation de traiter les problèmes 3D par une approche purement variationnelle. L'idée exploitée, basée sur les travaux de Serré [Serré 2000], est originale. Sa particularité n'apparaît pas dans le module de résolution. En effet, celui-ci ne fait qu'exploiter des outils déjà reconnus et dont l'utilisation dans la CAO a déjà été proposée par de nombreux auteurs. L'intérêt de l'approche se révèle essentiellement lors de reformulation du problème dans le modèle algébrique. En effet, la solution retenue permet d'obtenir un système d'équations représentatif du problème à résoudre. En d'autres termes, les PPSI issus de ce système correspondent aux étapes naturelles de résolutions du problème. Ainsi, pour les problèmes bien paramétrés, la décomposition obtenue est bien adaptée. Ce résultat a donc encouragé le développement des outils d'analyse du module de résolution. Or, le passage aux modèles algébriques dépend de la modélisation géométrique proposée. L'aspect déclaratif de celle-ci a grandement contribué à ce succès.

L'apport principal de ce travail de thèse est, malgré l'intérêt du modèle géométrique présenté et de la qualité du solveur d'ingénierie développé, l'architecture proposée pour une application variationnelle de CAO. Cette architecture n'apporte pas de gains particuliers dans l'application globale mais permet de faire évoluer les méthodes de travail. En effet, jusqu'à présent la construction d'un modèle était prescrite par la méthode imposée par le modelleur géométrique. La problématique qui consiste à répondre à un cahier des charges tout en rendant les spécifications évolutives était trop souvent mise de côté.

Pour favoriser cette méthode de travail, la solution développée a écarté le modelleur au profit du modèle d'informations. En d'autres termes, elle privilégie une description fonctionnelle du cahier des charges. Le modelleur et le solveur ne sont alors que des modules dont le rôle est de maintenir la cohérence des spécifications. Contrairement aux applications classiques où la modification d'une spécification est un casse-tête pour le concepteur, dans une application variationnelle, elle devient directe.

**Annexes
et
Bibliographie**

Annexe A

Rappels de la théorie des ensembles

Les définitions des notions issues de la *théorie des ensembles* utiles pour la modélisation volumique sont rappelées ici, elles sont en partie extraites de [Dieudonné 1990].

L'espace virtuel est l'espace usuel à trois dimension \mathbb{R}^3 . Cet espace est un espace *métrique*, c'est-à-dire un ensemble sur lequel est donnée une distance d .

Une *distance* sur un ensemble E est une application d de $E \times E$ dans l'ensemble \mathbb{R} des nombres réels, possédant les propriétés suivantes :

- $d(x, y) \geq 0$ pour tout couple d'éléments x, y de E ;
- La relation $d(x, y) = 0$ est équivalente à $x = y$;
- $d(x, y) = d(y, x)$ pour tout couple d'éléments de E ;
- $d(x, z) \leq d(x, y) + d(y, z)$ quels que soient les trois éléments x, y, z de E .

Dans l'espace \mathbb{R}^3 , la distance usuelle est la "distance euclidienne" définie par l'équation A.1.

$$d(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2} \quad (\text{A.1})$$

pour deux éléments $x = (x_1, x_2, x_3)$ et $y = (y_1, y_2, y_3)$.

Afin de définir l'intérieur, l'extérieur et la frontière d'un solide, quelques définitions supplémentaires sont nécessaires :

Boule ouverte : Étant donné un espace métrique E , avec une distance d , un point $a \in E$, et un nombre $r \in \mathbb{R}^{+*}$, la *boule ouverte* de centre a et de rayon r est l'ensemble $B(a, r) = \{x \in E \mid d(a, x) < r\}$.

Ouvert, fermé : A est un ensemble *ouvert* si et seulement si tout point a de A est le centre d'une boule ouverte contenue dans A . Un ensemble A est *fermé* si son complémentaire est ouvert.

Intérieur : Un point a est dit *intérieur* à un ensemble A , s'il existe $r \in \mathbb{R}^{+*}$ tel que la boule ouverte de rayon r centrée en a soit contenue dans A . L'ensemble $\overset{\circ}{A}$ de tous les points intérieurs à A s'appelle l'*intérieur* de A .

Extérieur : Un point $x \in E$ est dit *extérieur* à un ensemble $A \subset E$ ($A \neq \emptyset$), si $d(x, A) > 0$. L'ensemble de tous les points extérieurs à A s'appelle l'*extérieur* de A .

Adhérence : Un *point adhérent* à un sous-ensemble A de E est un point a tel que quelque soit $r \in \mathbb{R}^+$, la boule ouverte de centre a et de rayon r rencontre A . L'ensemble \tilde{A} des points adhérents à A est appelé l'*adhérence* de A .

Frontière : La *frontière* d'un ensemble A est l'ensemble ∂A des points appartenant à l'adhérence de A et à l'adhérence du complémentaire de A tel que : $\partial A = \tilde{A} \cap \tilde{\bar{A}}$.

Connexité : Un espace métrique E est dit *connexe* si les seuls sous-ensembles de E qui sont à la fois ouverts et fermés sont l'ensemble vide \emptyset et E lui-même. Autrement dit, E est connexe si et seulement si quelque soit $(a, b) \in E^2$, il existe une fonction continue $\varphi = \varphi(t), t \in [0, 1]$ telle que $\varphi(0) = a$, $\varphi(1) = b$ et, $\forall t \in [0, 1], \varphi(t) \in E$.

Variété topologique Une variété topologique de dimension m est un espace topologique séparé non vide où tout point possède un voisinage homéomorphe à un espace ou du demi-espace ouvert de dimension m [Godbillon 1971].

Annexe B

Compléments de géométrie et topologie des surfaces

Fonction de Morse Une fonction différentiable $f : V \rightarrow \mathbb{R}$ est dite fonction de Morse si tous ses points critiques sont non dégénérés. Un point critique m_0 est non dégénéré si par rapport à un système arbitraire de coordonnées locales (u, v)

$$\left(\frac{\partial^2 f}{\partial u^2} \cdot \frac{\partial^2 f}{\partial v^2} - \left(\frac{\partial^2 f}{\partial u \partial v} \right)^2 \right)_{m_0} \neq 0$$

Si ce nombre est négatif, m_0 est d'indice 1.

Si ce nombre est positif, m_0 est d'indice 0 ou 2 selon que $\left(\frac{\partial^2 f}{\partial u^2} \right)_{m_0}$ et $\left(\frac{\partial^2 f}{\partial v^2} \right)_{m_0}$ sont simultanément positifs ou négatifs.

Théorème de Morse Une fonction de Morse $f : V \rightarrow \mathbb{R}$ étant donnée sur une surface différentiable V compacte, ses points critiques sont en nombre fini. Si l'on note $c_i(f)$ le nombre des points critiques de f qui sont d'indice i ($i = 0, 1$ ou 2), on a toujours :

$$c_0(f) - c_1(f) + c_2(f) = \chi$$

Par exemple, si pour une orientation donnée, la fonction f est une fonction de hauteur, $c_0(f)$ (respectivement $c_1(f)$, $c_2(f)$) est le nombre de sommets (respectivement cols, fonds). Une approche B-rep originale basée sur ce théorème est proposée par Takahashi [Takahashi *et al.* 1997].

Annexe C

Quelques définitions et algorithmes de la théorie des graphes

Cet annexe a pour objectif de fournir quelques éléments sur la théorie des graphes. En effet, l'approche présentée utilise à différents niveaux des algorithmes classiques de ce domaine. Il présente dans un premier temps les graphes non orientés. Cette famille est particulièrement utilisée pour la gestion des différentes spécifications géométriques. Ensuite, les graphes bipartis seront expliqués. Ils sont utilisés pour l'analyse du système d'équations. Ce chapitre se termine sur les graphes orientés qui sont utilisés pour modéliser le plan de résolution.

C.1 Introduction

Le langage des graphes permet de représenter simplement la structure d'un grand nombre de situations :

- l'exemple le plus classique est la représentation d'un *réseau de communication* : réseau de routes représenté dans une carte routière, réseau de chemins de fer, de téléphone, de relais de télévision, réseaux électriques, ...
- la famille d'exemples la plus générale est la représentation d'une *relation binaire*, qu'elle soit algébrique, mécanique, chimique, sociologique ... comme les règles de certains jeux, l'ordonnancement des opérations de montage ou de démontage d'un ensemble technologique ...

Ce langage est utilisé à plusieurs reprises dans ce manuscrit. Le but de ce chapitre est de présenter brièvement les définitions nécessaires, les principales notions et les différents algorithmes utilisés. Il est basé sur l'ouvrage de Gondran et Minoux [Gondran *et al.* 1985].

C.2 Définitions et concepts de base

C.2.1 Généralités

Graphe Un graphe $G = [X, U]$ est la donnée d'un ensemble X dont les éléments sont appelés des *sommets* ou des *nœuds* et d'un ensemble U dont les éléments $u \in U$ sont des couples (ordonnés ou non) de sommets appelés des *arcs* (Figure C.1).

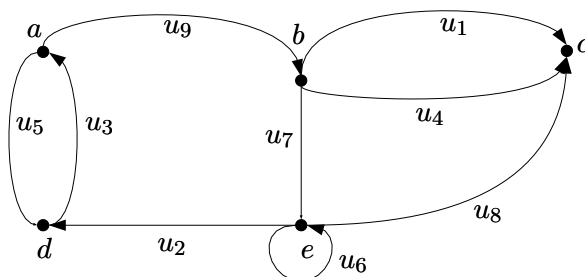


FIG. C.1: Exemple de graphe

Ordre d'un graphe L'ordre d'un graphe correspond au nombre N de ses sommets. On dit alors que le graphe est d'ordre N . Par exemple, le graphe de la figure C.1 est d'ordre 5.

Arc Un arc $u = (a, b)$ est un arc dont l'extrémité initiale est le sommet a et l'extrémité terminale, le sommet b . Un arc dont les extrémités coïncident est appelé une *boucle*. L'arc $u_7 = (e, e)$ de la figure C.1 est une boucle.

Arcs adjacents Deux arcs sont dits adjacents s'ils ont au moins une extrémité commune. Les arcs u_2, u_6, u_7 et u_8 de la figure C.1 sont adjacents.

Arête Pour certaines applications, l'orientation des arcs ne joue aucun rôle. On parlera alors de graphes non orientés. Les arcs définis par un couple non ordonné (a, b) sont appelés *arêtes* (a, b) .

C.2.2 Principales définitions

p-Graphe Un *p-graphe* est un graphe dans lequel il n'existe jamais plus de p arcs de la forme (a, b) entre deux sommets quelconques a et b , pris dans cet ordre. Le graphe de la figure C.1 est un 2-graphe.

Sous-graphe Étant donné un ensemble de sommets $A \subset X$, le sous-graphe engendré par A est le graphe G_A dont les sommets sont les éléments de A et dont les arcs sont les arcs de G ayant leurs deux extrémités dans A . Le graphe de la figure C.2-a est un sous-graphe du graphe de la figure C.1.

Graphe partiel Soit $G = [X, U]$ et $V \subset U$. Le *graphe partiel* engendré par $V \subset U$ est le graphe ayant le même ensemble X de sommets que G , et

dont les arcs sont les arcs de V . Le graphe de la figure C.2-a est un graphe partiel du graphe de la figure C.1.

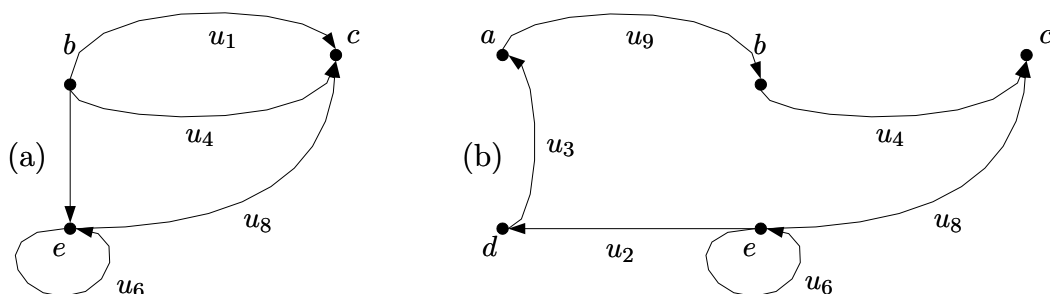


FIG. C.2: (a) Sous-graphe et (b) graphe partiel du graphe de la figure C.1

Sous-graphe partiel Soit $G = [X, U]$, $A \subset X$ et $V \subset U$, le *sous-graphe partiel* engendré par A et V est le graphe partiel G_A engendré par V : $G_A = [A, V]$.

Graphe fondamental Un graphe fondamental G^* d'un graphe $G = [X, U]$ est l'unique sous-graphe de G contenant le même ensemble d'arêtes U et ne contenant pas de sommet de degré nul.

Graphe induit Soit $G = [X, U]$ un graphe. Soit $\Pi_i, i \in \{1, \dots, n\}$, une partition de X en composants disjoints tels que $X = \cup_{i=1}^n \Pi_i$. Le graphe induit G° du graphe $G = [X, U]$ par la partition Π_i est le graphe $G^\circ = [X^\circ, U^\circ]$ tel que :

- $X^\circ = \{\Pi_i, i \in \{1, \dots, n\}\}$;
- $U^\circ = \{(\Pi_i, \Pi_j) \mid \exists x_p \in \Pi_i, \exists x_q \in \Pi_j, (x_p, x_q) \in U\}$.

Chaîne, chaîne élémentaire Une *chaîne* de longueur q est une séquence de q arcs telle que chaque arc u_r de la séquence ait une extrémité commune avec l'arc u_{r-1} et l'autre extrémité commune avec l'arc u_{r+1} .

Une *chaîne élémentaire* est une chaîne telle qu'en la parcourant, on ne rencontre pas deux fois le même sommet (voir figure C.3).

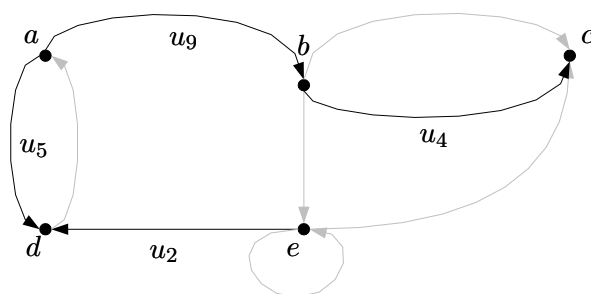


FIG. C.3: Une chaîne élémentaire du graphe de la figure C.1

Cycle, cycle élémentaire Un cycle (élémentaire) est une chaîne (élémentaire) dont les extrémités coïncident.

On dit que p cycles μ^1, \dots, μ^p sont indépendants si :

$$\lambda_1 \vec{\mu}^1 + \dots + \lambda_p \vec{\mu}^p = \vec{0} \Rightarrow \lambda_i = 0, \forall i \in \{1, \dots, p\}.$$

Dans le cas contraire, ces cycles sont dépendants.

Une *base de cycles* est un ensemble minimal de cycles indépendants $\{\mu^1, \dots, \mu^p\}$ tel que tout cycle puisse s'écrire sous la forme d'une combinaison linéaire des cycles de la base (voir figure C.4).

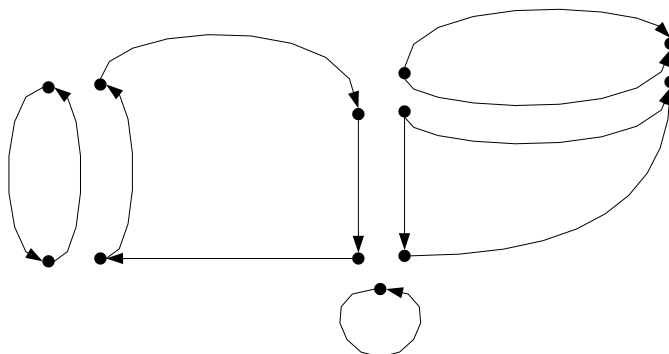


FIG. C.4: Une base de cycles du graphe de la figure C.1

On appelle nombre *cyclomatique* d'un graphe G la dimension d'une base de cycles. On le note : $\nu(G)$.

Cocycles Un cocycle d'un sous-ensemble de nœuds A d'un graphe $G = [X, U]$ est l'ensemble des arcs ayant une extrémité et une seule appartenant à A . On le note : $\omega(A)$. On peut définir le sous ensemble $\omega^+(A)$ (respectivement $\omega^-(A)$) par l'ensemble des arcs appartenant à $\omega(A)$ dont l'extrémité initiale (respectivement finale) appartient à A . On appelle *degré* d'un sommet x , le cardinal du cocycle $\omega(x)$.

Chemin (élémentaire), circuit (élémentaire) Un chemin (élémentaire) est une chaîne (élémentaire) dont les arcs sont orientés. Un circuit (élémentaire) est un chemin (élémentaire) dont les extrémités coïncident. La chaîne de la figure C.3 n'est pas un chemin alors que la chaîne $\{u_3, u_9, u_7, u_2\}$ est un circuit élémentaire.

C.2.3 Connexité

Grphe connexe, composante connexe Un graphe est dit *connexe*, si pour tout couple de sommets x_i et x_j , il existe une chaîne joignant x_i et x_j . Une composante connexe contenant un sommet x est le sous-graphe connexe d'ordre maximal contenant x . Les sous-graphes $\{a, b, c\}$ et $\{d, e, f, g\}$ de la figure C.5 sont des composantes connexes.

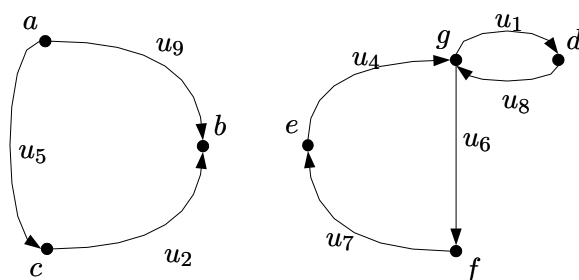


FIG. C.5: Composantes connexes et fortement connexes.

Grphe fortement connexe, composante fortement connexe Un graphe est dit *fortement connexe* si, étant donnés deux sommets quelconques x_i et x_j , il existe un chemin d'extrémité initiale x_i et d'extrémité terminale x_j . Une composante fortement connexe contenant un sommet x est le sous-graphe fortement connexe d'ordre maximal contenant x (Cf. le sous-graphe $\{d, e, f, g\}$ de la figure C.5).

Point d'articulation Un point d'articulation d'un graphe est un sommet dont la suppression augmente le nombre de composantes connexes. Les points b , e et g du graphe de la figure C.6 sont des points d'articulation.

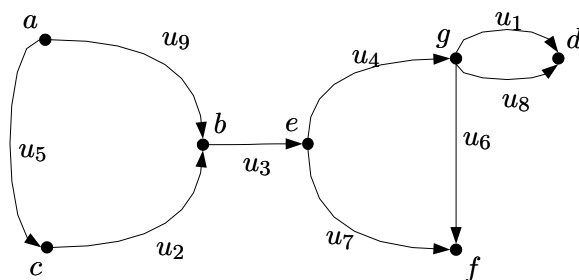


FIG. C.6: Illustration des points d'articulation et des isthmes.

Isthme Un isthme est une arête dont la suppression augmente le nombre de composantes connexes. L'arête u_3 du graphe de la figure C.6 est un isthme.

C.2.4 Quelques graphes particuliers

Grphe simple Un graphe est dit *simple* si :

- il est non orienté ;
- il est sans boucle ;
- c'est un 1-graphe.

Arbre, forêt Un *arbre* est un graphe connexe sans cycle (Figure C.7). Un graphe sans cycle qui n'est pas connexe est appelé une *forêt*. Lorsqu'un arbre A (ou une forêt F) est un graphe partiel d'un graphe G , on dira que A (respectivement F) est un arbre (respectivement une forêt) de G . Une forêt

maximale de G est un graphe F partiel sans cycle de G , et est maximum avec cette propriété.

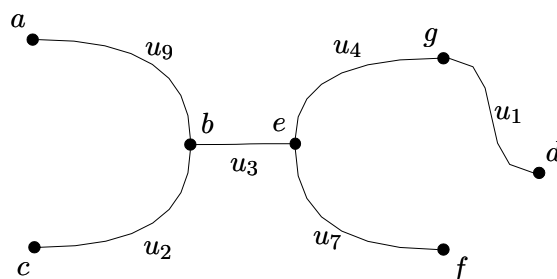


FIG. C.7: Ce graphe simple est un arbre non orienté

Soit $G = [X, U]$ un graphe connexe. On dit que le graphe partiel $H = [X, V]$ est un *coarbre* de G si et seulement si le graphe partiel $\tilde{H} = [X, U - V]$ est un *arbre* de G .

Graphe aux arêtes Le graphe aux arêtes d'un graphe G est un graphe dont les sommets représentent les arêtes de G , deux de ces sommets étant joints si les deux arêtes qu'ils représentent ont une extrémité commune dans G . La figure C.8 représente le graphe aux arêtes de l'arbre de la figure C.7.

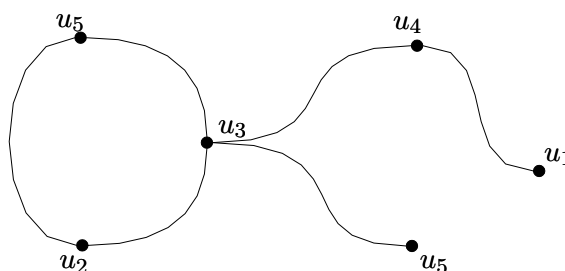


FIG. C.8: Graphe aux arêtes du graphe de la figure C.7

Graphe pondéré Un graphe $G = [X, U]$ est pondéré lorsqu'un poids $p(u) \in \mathbb{R}$ appelé "longueur de l'arc" est associé à chaque arc ou arête $u \in U$. On appelle *poids* de G le nombre P tel que :

$$P(G) = \sum_{u \in U} p(u)$$

Dans toute la suite, on utilisera des graphes dont les poids sont positifs : $p(u) \in \mathbb{R}^+, \forall u \in U$.

Graphe biparti On dit qu'un graphe $G = [X, U]$ est un graphe biparti si l'ensemble des sommets X peut-être partitionné en deux sous-ensembles S et T de telle sorte que pour toute arête $(a, b) \in U$:

$$a \in S \Rightarrow b \in T ; \tag{C.1}$$

$$a \in T \Rightarrow b \in S. \tag{C.2}$$

On parlera d'un graphe biparti $(m \times n)$ pour un graphe $G = [S \cup T, U]$ tel que $|S| = m$ et $|T| = n$. La figure C.9 représente un graphe biparti (9×8) .

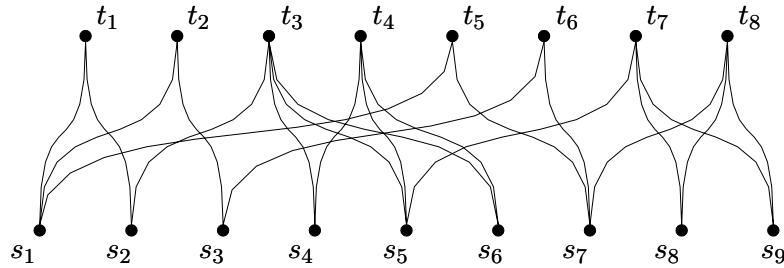


FIG. C.9: Graphe biparti

Après avoir défini les principaux termes de la théorie des graphes, les différents algorithmes nécessaires sont présentés.

C.3 Problème du plus court chemin

Le problème du plus court chemin est un des problèmes les plus anciens de la théorie des graphes. Il se rencontre soit directement, soit comme sous-algorithme pour répondre aux différents besoins de l'approche.

C.3.1 Définition

Étant donné un graphe orienté $G = [X, U]$ pondéré, le problème du plus court chemin entre deux sommets $\{x_i, x_j\} \in X^2$ est de trouver un chemin $\mu(x_i, x_j)$ de x_i à x_j dont la longueur totale, $l(\mu)$, soit minimum, avec :

$$l(\mu) = \sum_{u \in \mu(x_i, x_j)} p(u)$$

C.3.2 Algorithme de Moore-Dijkstra

L'idée de cet algorithme est de trouver le plus court chemin entre un sommet $x \in X$ et tous les autres. Pour trouver le chemin le plus court entre deux sommets, il suffit de l'arrêter dès que ce chemin est trouvé. Il est aussi basé sur une technique de propagation de front. La pertinence de l'algorithme est due à l'ordonnement des nœuds du front. En effet, ces nœuds sont ordonnées en fonction de leur distance par rapport au sommet x . Pour cet algorithme, sont définis :

- $S \subset X$, l'ensemble des nœuds traités du graphe ;
- $F \subset X$, l'ensemble ordonné des nœuds du front ;
- $\mu(x_j)$, le chemin le plus court identifié jusqu'au sommet $x_j \in (S \cup F)$;

- $p(x_j)$ le poids (ou la longueur) du chemin $\mu(x_j)$.

L'algorithme est initialisé par :

- $S = \{x\}$,
- $F = \Gamma_x$,
- $\mu(x_j) = \{u_{x,x_j}\}, \forall x_j \in \Gamma_x$,
- $p(x_j) = p(x, x_j), \forall x_j \in \Gamma_x$,

L'algorithme se décompose selon les étapes suivantes :

Tant qu'il reste des nœuds dans F faire :

Choisir le nœud x_i du front qui a la plus petite distance à x et faire :

Pour chaque nœud $x_j \in \Gamma_{x_i}$:

Si $x_j \in F$, faire :

Si $p(x_i) + p(x_i, x_j) < p(x_j)$:

$$\mu(x_j) = \mu(x_i) + u_{x_i, x_j}$$

$$p(x_j) = p(x_i) + p(x_i, x_j)$$

Fin faire.

Fin faire.

Sinon ($x_j \in X - (S \cup F)$) faire :

$$\mu(x_j) = \mu(x_i) + u_{x_i, x_j},$$

$$p(x_j) = p(x_i) + p(x_i, x_j),$$

$$F = F + \{x_j\},$$

Fin faire.

Fin faire.

$$F = F - \{x_j\},$$

$$S = S + \{x_j\},$$

Fin faire.

Fin faire.

Pour trouver la plus petite chaîne entre un sommet $x \in X$ et tous les autres, il suffit de ne pas prendre en compte l'orientation des arcs.

C.4 Recherche d'une forêt maximale d'un graphe

C.4.1 Algorithme élémentaire

Soit un graphe $G = [X, U]$. Initialement, tous les arcs du graphe sont incolores. La méthode consiste alors à examiner successivement tous les arcs du graphe et à les colorer soit en *rouge*, soit en *vert*.

Chaque fois qu'un nouvel arc incolore u est examiné, de deux choses l'une :

- soit un cycle élémentaire μ , dont tous les arcs autres que u sont rouges, passe par u . On colore l'arc u en *vert*.
- soit un tel cycle n'existe pas et, auquel cas, l'arc u est colorié en *rouge*.

Le graphe partiel rouge est alors une forêt de G puisqu'il est sans cycle. Cette forêt est maximale puisqu'en ajoutant un arc vert quelconque on crée un cycle.

Nombre cyclomatique

Soit $A = [X, U_A]$ une forêt maximale d'un graphe $G = [X, U]$. Alors, par tout arc $u \in (U - U_A)$, passe un cycle et un seul μ^u dont tous les arcs (autres que u) appartiennent à A . Le nombre cyclomatique $\nu(G)$ d'un graphe G est donc égal à :

$$\nu(G) = |U| - |U_A|. \quad (\text{C.3})$$

C.4.2 Recherche d'un arbre de poids minimum d'un graphe connexe

Définition

Le poids d'un graphe partiel $A = [X, U_A]$ est défini par la somme du poids de chaque arête $u \in U_A$ selon :

$$w(A) = \sum_{u \in U_A} w(u). \quad (\text{C.4})$$

Le problème de l'arbre de poids minimum est de rechercher un arbre $A = [X, U_A]$ de $G = [X, U]$ tel que son poids soit le minimum des poids pris sur l'ensemble des arbres de G possibles :

$$w(A) = \min_i \{w(A_i)\}.$$

Algorithme

L'algorithme utilisé est basé sur une technique de propagation de front afin d'obtenir les arêtes ordonnées du coarbre dans l'ordre d'apparition. Pour cet algorithme, sont définis :

- $Y \subset X$, l'ensemble des nœuds traités du graphe,
- $T \subset U$, l'ensemble des arêtes de l'arbre partiel,
- $V \subset U$, l'ensemble des arêtes du coarbre partiel,
- $\omega(Y)$, le cocycle de Y .

L'algorithme est initialisé par :

- $Y = \{x_0\}$, avec $x_0 \in X$,
- $\omega(Y) = \omega(x_0)$,
- $T = \emptyset$.

– $V = \emptyset$.

L'algorithme se décompose selon les étapes suivantes :

Tant que $Y \neq \emptyset$ faire :

Soit $u \in \omega(Y)$ tel que $w(u) = \min_{v \in \omega(Y)} w(v)$,

Soit $x \notin Y$ l'autre extrémité de u ,

$T = T \cup \{u\}$,

$V = V \cup ([\omega(Y) \cap \omega(x)] - \{u\})$

$\omega(Y) = \omega(Y) \Delta \omega(x)$

$Y = Y \cup \{x\}$,

Fin faire.

L'arbre et le coarbre recherchés sont respectivement les graphes partiels

$A = [X, T]$ et $H = [X, V]$.

C.5 Base de cycles d'un graphe non orienté

Trouver une base de cycles \mathcal{B} d'un graphe $G = [X, U]$ n'est pas très difficile dès qu'une forêt $F = [X, U_F]$ maximale a été identifiée. En effet, pour chaque composante connexe $G_i = [X_i, U_i]$ du graphe G , un arbre $A_i = [X_i, U_{A_i}]$ et un coarbre $H_i = [X_i, U_i - U_{A_i}]$ sont connus. Nous avons vu que le nombre cyclomatique $\nu(G_i)$ du sous graphe connexe G_i est égal au cardinal de l'ensemble des arêtes du coarbre et surtout, qu'un cycle est créé dès que l'on ajoute une arête du coarbre H_i à l'arbre A_i .

C.5.1 Base de cycles quelconques

Un premier algorithme pour trouver une base de cycles consiste pour chaque composante connexe G_i , à identifier un arbre A_i et surtout un coarbre H_i . Notons que l'identification des composantes connexes peut être obtenue en même temps que la recherche d'une forêt maximale par un algorithme utilisant une technique de propagation de front. Après cette première étape nous avons :

- $U_F \subset U$, l'ensemble des arêtes d'une forêt maximale de G ,
- $U_H = U - U_F$, l'ensemble des arêtes des coarbres de G ,
- $\nu(G) = |U_H|$, le nombre cyclomatique du graphe G .

L'algorithme est initialisé par :

- $U' = U_F$,

Il reste donc à identifier les cycles :

Pour $k = 1$ jusqu'à $k = \nu(G)$ faire :

Soit $u_k = (x_i, x_j)$ la $k^{\text{ième}}$ arête de U_H ,

Identifier une chaîne $\mu(i, j)$ d'extrémité $\{x_i, x_j\}$ dans le graphe partiel

$$\begin{aligned} G' &= [X, U'], \\ U' &= U' \cup \{u_k\}, \\ \mu^k &= \mu(x_i, x_j) + u_k \text{ est alors un cycle,} \end{aligned}$$

Fin faire.

A chaque étape, les cycles $\{\mu^0, \dots, \mu^k\}$ forment alors un ensemble de cycles indépendants pour le graphe $G = [X, U]$. En fin de procédure le cardinal de cet ensemble est donc égal à $\nu(G)$, les cycles identifiés sont toujours indépendants, une base de cycles du graphe $G = [X, U]$ est donc identifiée.

C.5.2 Base de cycles minimaux

Pour les besoins de l'approche, la base de cycles \mathcal{B} ne peut pas être quelconque. L'idéal est une base définie par l'ensemble des cycles minimaux. Elle minimise la relation C.5.

$$w(\mathcal{B}) = \sum_{i=1}^{\nu(G)} w(\mu^i), \tag{C.5}$$

avec :

$$w(\mu^i) = \sum_{u_k \in \mu^i} w(u_k).$$

Pour identifier cette base de cycles dans un graphe connexe¹, il suffit de modifier l'algorithme initial en prenant en compte les remarques suivantes :

- l'arbre ne peut pas être quelconque, il faut choisir un arbre de poids minimum, c'est-à-dire un coarbre de poids maximum,
- à chaque itération k , il faut choisir dans l'ensemble des arêtes non traitées de U_H celle qui permettra de construire le cycle de poids minimum,
- à une itération k , le cycle de poids minimum passant par l'arête $u_k = (x_i, x_j)$ est obtenu par la chaîne de longueur minimale dans le graphe partiel $G' = [X, U']$.

Comme lors de l'identification de l'arbre de poids minimum, les arêtes du coarbre ont été stockées dans l'ordre d'apparition, il n'est pas nécessaire de tester chaque arête pour déterminer celle qui permettra de construire le cycle de poids minimum. La première est une bonne candidate.

C.5.3 Gestion dynamique d'une base de cycles

Les graphes utilisés dans cette approche sont en continuelle évolution. En effet, à chaque intervention de l'utilisateur, un ensemble d'arêtes et de sommets peuvent être créés tandis que d'autres seront supprimés. Le problème est de maintenir dans le graphe une base de cycles minimaux. Une solution serait de lancer un

¹si le graphe n'est pas connexe, il suffit d'appliquer l'algorithme à chaque composante connexe séparément

algorithme d'identification sur le graphe complet mais, comme en général les modifications apportées ne sont que locales, il paraît plus judicieux de ne faire la modification de la base que localement.

L'ajout d'un ensemble de nœuds ne modifie pas la base de cycles², alors que la suppression de sommets provoque la suppression d'un ensemble d'arêtes et donc d'un ensemble de cycles. La suppression d'un nœud est donc traitée comme la suppression d'un ensemble d'arêtes.

Suppression d'une arête

Soit u une arête à supprimer dans le graphe $G = [X, U]$. Par u passe un ensemble de cycles. Or, la suppression d'une arête entraîne la suppression d'au plus un cycle dans la base \mathcal{B} pour qu'elle reste une base du graphe. Soit $A = [X', U']$, le sous graphe partiel défini par l'ensemble des arêtes des cycles passant par u . Soit \mathcal{B}_A l'ensemble de ces cycles (qui définissent une base de cycles pour le graphe A). Le sous graphe A est alors le plus petit sous graphe impacté. Il suffit alors de rechercher dans le graphe $A' = [X', U' - \{u\}]$ une base de cycle $\mathcal{B}_{A'}$. Cet ensemble de cycles est alors substitué dans \mathcal{B} à l'ensemble de cycles \mathcal{B}_A (Éq. C.6). Le nouvel ensemble de cycles \mathcal{B}' définit alors une base de cycles minimaux pour le graphe $G = [X, U - \{u\}]$.

$$\mathcal{B}' = (\mathcal{B} - \mathcal{B}_A) \cup \mathcal{B}_{A'} \quad (\text{C.6})$$

Suppression d'un ensemble d'arêtes

Pour connaître la nouvelle base de cycles \mathcal{B}' du graphe G diminué d'un ensemble d'arêtes $V : G = [X, U - V]$, il suffit de trouver le sous graphe partiel A défini par l'ensemble des arêtes des cycles contenant une des arêtes à supprimer. À partir du graphe A , on définit le graphe partiel A' de A réduit de l'ensemble des arêtes de V . Et, de la même façon que pour la suppression d'une arête, la base de cycles minimaux $\mathcal{B}_{A'}$ de A' est identifiée et cet ensemble de cycles est substitué dans \mathcal{B} aux cycles de la base \mathcal{B}_A de A .

Ajout d'une arête

Soit \mathcal{B} , la base de cycles minimaux du graphe $G = [X, U]$. Soit $u = (x_i, x_j)$ une nouvelle arête du graphe G . Si les extrémités $\{x_i, x_j\}$ de la nouvelle arête sont deux nœuds d'une même composante connexe de G , alors la base de cycle doit être augmentée d'un cycle. En effet, dans un graphe, le nombre cyclomatique est égal à $\nu(G) = |U| - |X| + p$, avec p le nombre de composantes connexes de G .

²en effet, tant que le nœud n'est pas lié à un autre par un arc, il définit à lui seul une composante connexe

Le nouveau cycle est le cycle μ défini par $\mu = \mu(x_i, x_j) + u$ avec $\mu(x_i, x_j)$ le chemin le plus court joignant (x_i, x_j) dans le graphe $G = [X, U]$. Mais cette nouvelle base $\mathcal{B}' = \mathcal{B} \cup \{\mu\}$ n'est pas obligatoirement une base de cycles minimaux. La figure C.10 représente deux composantes connexes d'un graphe $G = [X, U \cup V]$. Pour simplifier, le poids de chaque arête est égal à 1. La base \mathcal{B} de G est composé des cycles :

- $\mu^1 = \{u_1, u_2, u_3, u_8, u_4\}$
- $\mu^2 = \{u_3, u_5, u_9, u_6, u_7\}$
- $\mu^3 = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$
- $\mu^4 = \{v_1, v_2, v_3, v_{10}, v_{11}, v_9\}$
- $\mu^5 = \{v_{10}, v_{11}, v_9, v_{12}, v_{13}, v_{14}\}$

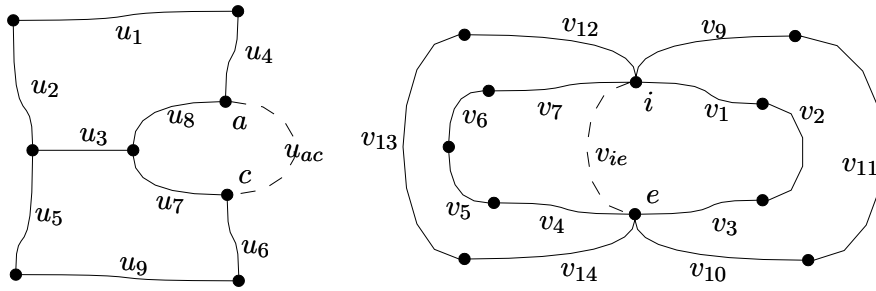


FIG. C.10: Deux composantes connexes G' et G'' chacune complétée d'une arête

Si l'on ajoute l'arête u_{ac} , les extrémités de cette arête appartiennent à une même composante connexe, alors le nombre cyclomatique du graphe est augmenté de 1. Le nouveau cycle $\mu^6 = \{u_{ac}, u_7, u_8\}$ est composé de la nouvelle arête et du chemin le plus court entre a et c . Il est ajouté à la base \mathcal{B} qui reste minimale.

Si l'on ajoute l'arête v_{ie} , un septième cycle doit donc être identifié pour compléter la base \mathcal{B} afin qu'elle reste une base. Le cycle minimal contenant v_{ie} est donc identifié : $\mu^7 = \{v_1, v_2, v_3, v_{ie}\}$. Mais la base n'est plus minimale. En effet, le poids de la base composée des cycles $\{\mu^1, \mu^2, (\mu^3 - \mu^7), \mu^4, \mu^5, \mu^6, \mu^7\}$ a un poids inférieur. La base de poids minimaux est alors constituée des cycles $\{\mu'^1, \mu'^2, \mu'^3, \mu'^4, \mu'^5, \mu'^6, \mu'^7\}$ avec :

- $\mu'^1 = \mu^1$;
- $\mu'^2 = \mu^2$;
- $\mu'^3 = \{u_{ac}, u_7, u_8\}$;
- $\mu'^4 = \{v_{ie}, v_1, v_2, v_3\}$;
- $\mu'^5 = \{v_{ie}, v_{10}, v_{11}, v_9\}$;
- $\mu'^6 = \{v_{ie}, v_{12}, v_{13}, v_{14}\}$;
- $\mu'^7 = \{v_{ie}, v_4, v_5, v_6, v_7\}$.

Pour identifier cette base sans rechercher complètement une base de cycles du graphe complet, il suffit de trouver le graphe partiel G' dont les cycles de la base de poids minimum sont impactés. Ce sous graphe est contenu dans le sous

graphe G'' dont les cycles de la base pourraient être impactés. G'' peut être défini par l'ensemble des cycles contenant aux moins deux sommets appartenant à la chaîne la plus courte entre les deux extrémités de la nouvelle arête u . Ces cycles ne seront pas obligatoirement modifiés : pour l'exemple de la figure C.10 l'ajout de l'arête u_{ac} n'a modifié aucun des cycles μ^1 et μ^2 .

À partir de $G'' = [X, U'']$, il faut identifier G' . Soit μ la chaîne la plus courte dans G entre les deux extrémités de la nouvelle arête. Soit μ_i un cycle de la base \mathcal{B} contenu dans G'' . Soit x_1 et x_2 respectivement les premier et dernier sommets de μ_i rencontrés par μ . Il est facile de montrer que μ_i n'appartiendra pas à la base de poids minimaux du nouveau graphe si la chaîne la plus courte dans G'' entre les sommets x_1 et x_2 passe par u . Si ce n'est pas le cas, G'' contient μ_i .

Ajout d'un ensemble d'arête

Pour traiter l'ajout d'un ensemble d'arêtes de façon optimale, il suffit, pour chaque arête u_i , d'identifier le plus petit graphe partiel G''_i , dont une nouvelle base est à trouver. La base du graphe complet est alors composée des cycles inchangés et de ceux de la base du graphe $G'' = \sum_i G''_i$.

C.6 Graphe biparti

Un graphe biparti $G = [S \cup T, U]$ est un graphe dont l'ensemble des sommets peut être partitionné en deux classes S et T , de sorte que deux sommets de la même classe ne soient jamais adjacents. Ce graphe est intéressant pour modéliser un système d'équations : le premier ensemble de nœuds regroupe les équations et le second les variables. Les algorithmes associés à cette famille de graphes [Murota 1987] permettent de décomposer le système d'équations en sous-systèmes indépendants et de les ordonner. Cette décomposition est réalisée en deux étapes : la première consiste à trouver un couplage maximum du graphe biparti et la seconde à décomposer ce graphe en composantes irréductibles. Après la description de ces deux étapes, les adaptations des algorithmes à une gestion dynamique des modifications seront présentées. Mais avant tout, quelques définitions nécessaires seront énoncées.

C.6.1 Couplage

Avant de présenter l'algorithme permettant d'obtenir un couplage maximal, un ensemble de termes propres à la notion de couplage est précisé :

Couplage Étant donné un graphe simple $G = [S \cup T, U]$ non orienté, un *couplage* est un sous-ensemble d'arêtes $U' \in U$ tel que deux arêtes quelconques

de U' ne sont pas adjacentes. Par convention, les arêtes d'un couplage sont dessinées en traits épais alors que les autres restent en traits fins.

Sommets et arêtes (in)saturés Un sommet x_i est dit *saturé* par un couplage $C \subset U$ s'il existe une arête de C incidente à x_i . Dans le cas contraire, on dit que le sommet x_i est *insaturé*. Les arêtes de C sont dites *saturées*.

Couplage complet, couplage parfait Un *couplage* qui ne laisse pas deux sommets adjacents insaturés est dit *complet* (Figure C.11-a). S'il sature tous les sommets du graphe, c'est un *couplage parfait* (Figure C.11-c).

Chaîne alternée Si $C \subset U$ est un couplage de $G = [S \cup T, U]$, on appelle *chaîne alternée* (relativement à C) une chaîne de G dont les arêtes sont alternativement dans C et dans $(U - C)$.

On appelle *chaîne alternée augmentante*, une chaîne alternée joignant deux sommets insaturés. La chaîne alternée $\{(s_1, t_1), (t_1, s_2), (s_2, t_2), (t_2, s_3), (s_3, t_3)\}$ de la figure C.11-a) en est un exemple.

Un *cycle alterné* est un chaîne alternée dont les deux extrémités coïncident. Le *rang* d'un cycle alterné est le cardinal de l'ensemble des arêtes saturées du cycle.

Orientons un graphe biparti tel que ses arêtes insaturées deviennent les arcs (s_i, t_j) et ses arêtes saturées, les arcs (t_j, s_i) . La *chaîne alternée supérieure* (respectivement *inférieure*) partant d'un sommet x est un chemin dans le graphe biparti partant de x dont le sens de parcours est positif (respectivement négatif). Le *sous-graphe supérieur* de racine x est le sous-graphe fondamental défini par l'ensemble des chaînes alternées supérieures partant de x .

Couplage maximum Un *couplage maximum* est un couplage de cardinalité maximale (Figure C.11-b). Un couplage C est maximum si et seulement si il n'existe pas de chaîne alternée augmentante relativement à C (Théorème de Berge, 1957).

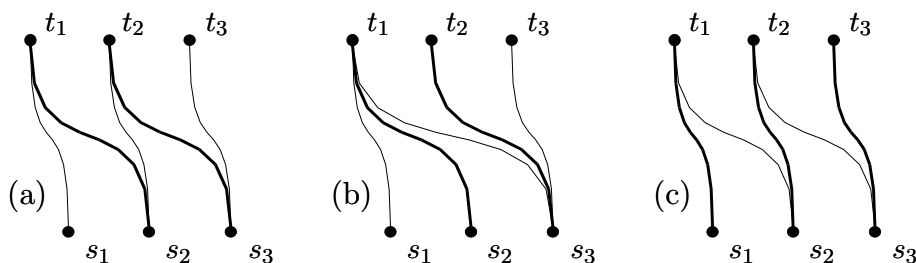


FIG. C.11: *Couplage complet (a), couplage maximum (b), couplage parfait (c)*

Arête (in)admissible Une arête u d'un graphe G est dite *admissible* s'il existe un couplage maximum contenant u . Autrement, l'arête est dite *inadmissible*.

Algorithme de couplage maximum

Hopcroft et Krap [Hopcroft *et al.* 1973] proposent un algorithme pour trouver un couplage maximum dont la complexité est proportionnelle à $(m + n) \sqrt{n}$, avec n le nombre d'arête et m le nombre de sommet (à vérifier). Cet algorithme part d'un couplage vide. A chaque étape, une chaîne augmentante est recherchée. Si elle est trouvée, la cardinalité du couplage augmente d'une unité. L'algorithme se termine lorsqu'aucune nouvelle chaîne ne peut être trouvée.

C.6.2 Décomposition du graphe biparti

Afin de présenter la structure d'un graphe biparti, un ensemble de définitions très utiles, proposées par Dulmage et Mendelsohn, sont précisées.

Recouvrement extérieur Soit $G = [S \cup T, E]$ un graphe biparti. Un couple d'ensembles de sommets $A \subset S$ et $B \subset T$ est un *recouvrement extérieur* de G si les cocycles des ensembles A et B recouvrent l'ensemble des arêtes du graphe : $E = \omega(A) \cup \omega(B)$ (Figure C.12-a). Le nombre $|A| + |B|$ est appelé la dimension d'un recouvrement. La *dimension extérieure* $E(G)$ de G est alors définie comme $E(G) = \min(|A| + |B|)$ pour tout couple $[A, B]$ qui recouvrent G . Un couple $[A, B]$ tel que $|A| + |B| = E(G)$ est appelé *couple extérieur minimal* (Figure C.12-b).

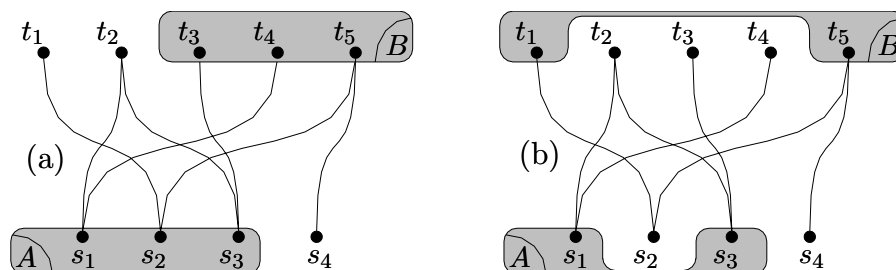


FIG. C.12: Recouvrement extérieur quelconque (a), couple extérieur minimal (b)

Grphe réductible, semi-réductible, irréductible Soit un graphe biparti $G = [S \cup T, E]$. Soit $A \subset S$, $B \subset T$. Le graphe G est dit [Dulmage *et al.* 1959] :

- réductible s'il existe un couple extérieur minimal $[A, B]$ tel que $A \neq \emptyset$ et $B \neq \emptyset$ (Figure C.13-a);
- semi-réductible s'il n'existe qu'un seul couple extérieur minimal $[A, B]$ et ce couple est tel que $A = \emptyset$ ou $B = \emptyset$ (Figure C.13-b);
- irréductible si et seulement si il n'existe que deux couples extérieurs minimaux $[A, \emptyset]$ et $[\emptyset, B]$ (Figure C.13-c).

Il est évident qu'un graphe irréductible ou semi-réductible ne contient pas d'arêtes inadmissibles.

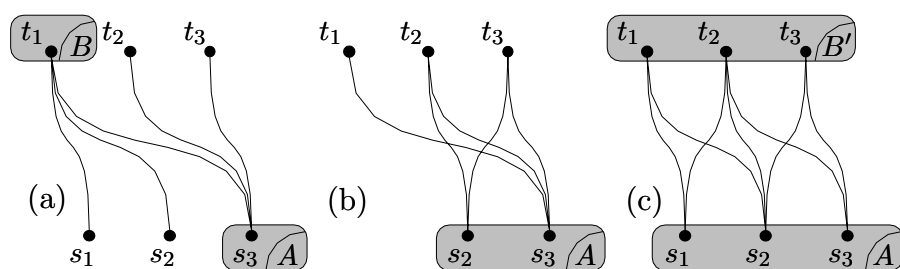


FIG. C.13: Graphes réductible (a), semi-réductible (b), irréductible (c)

Une condition suffisante pour qu'un sous-graphe biparti $(r \times r)$ soit irréductible est qu'il contienne un cycle alterné de rang r [Dulmage *et al.* 1959].

Structure d'un graphe biparti

Soit un graphe biparti $G = [S \cup T, U]$. Soit $\{[A_i, B_i], i \in \{0, \dots, k\}\}$ l'ensemble des couples extérieurs minimaux. Dans cet ensemble, il existe deux couples $[A_*, B^*]$ et $[A^*, B_*]$ vérifiant : $A_* = \bigcap_i A_i$, $A^* = \bigcup_i A_i$, $B_* = \bigcap_i B_i$, $B^* = \bigcup_i B_i$. Ces couples sont les *couples extérieurs minimaux extrêmes* de G . Pour le graphe de la figure C.14, $A_* = U_1$ et $B_* = X_1 \cup X_2$.

Si $A^* \neq A_*$, il existe $S_1 \subset \{S - A^*\}$ et $T_1 \subset T$ tels que :

- $S_1 \cap A_* = \emptyset$;
- $[A_* \cup S_1, B^* - T_1]$ soit un couple extérieur minimal ;
- $|S_1|$ soit minimal.

Puis, par récurrence, les ensembles S_i et T_i sont construits tels que :

- $S_i \cap \{A_* \cup_{j=1}^{i-1} S_j\} = \emptyset$;
- $[A_* \cup_{j=1}^i S_j, B^* - \cup_{j=1}^i T_j]$ soit un couple extérieur minimal ;
- $|S_i|$ soit minimal dans $\{S - A^* \cup_{j=1}^{i-1} S_j\}$.

En fin de procédure, on a :

- $S = A_* \cup \overline{A^*} \cup_{i=1}^k S_i$;
- $T = \overline{B^*} \cup B_* \cup_{i=1}^k T_i$.

Le graphe fondamental $G^* = [\cup_i S_i \cup_i T_i, U_n]$ est un graphe formé des composantes connexes $G_i = [S_i \cup T_i, E_i]$ qui sont les composantes irréductibles de G . Ce sous-graphe est le *noyau* de G .

Les composantes connexes $H_i = [U_i \cup V_i, E_{H_i}]$ et $L_i = [X_i \cup W_i, E_{L_i}]$ respectivement des graphes fondamentaux $G^* = [A_* \cup \overline{B^*}, \cup_i E_{H_i}]$ et $G^* = [\overline{A^*} \cup B_*, \cup_i E_{L_i}]$ sont les composantes semi-irréductibles de G .

La décomposition des ensembles S et T en ensembles disjoints U_i, S_j, X_k et respectivement V_i, T_j, W_k , définit la décomposition canonique de G [Dulmage *et al.* 1958].

À partir de cette décomposition, on peut définir différentes zones de l'espace

$S \times T$ par :

$$R_1 = (A_* \times \overline{B^*}) \cup (\overline{A^*} \times B_*) \cup_{i=1}^k (S_i \times T_i) \quad (\text{C.7})$$

$$R_2 = (A_* \times B^*) \cup (A^* \times B_*) \cup_{i < j} (S_i \times T_j) \quad (\text{C.8})$$

$$R_3^\alpha = \cup_{i > j} (S_i \times T_j) \quad (\text{C.9})$$

$$R_3^\beta = \cup_i (S_i \times \overline{B^*}) \quad (\text{C.10})$$

$$R_3^\gamma = \cup_j (\overline{B^*} \times T_j) \quad (\text{C.11})$$

$$R_3^\delta = \overline{A^*} \times \overline{B_*} \quad (\text{C.12})$$

$$R_4 = \cup_{i \neq j} (U_i \times V_j) \quad (\text{C.13})$$

$$R_5 = \cup_{i \neq j} (X_i \times W_j) \quad (\text{C.14})$$

Les arêtes admissibles de G sont contenues dans R_1 alors que R_2 ne contient que les arêtes inadmissibles. Quant à la région R_3 , elle ne contient aucune arête de G . La figure C.14 illustre la décomposition canonique d'un graphe biparti.

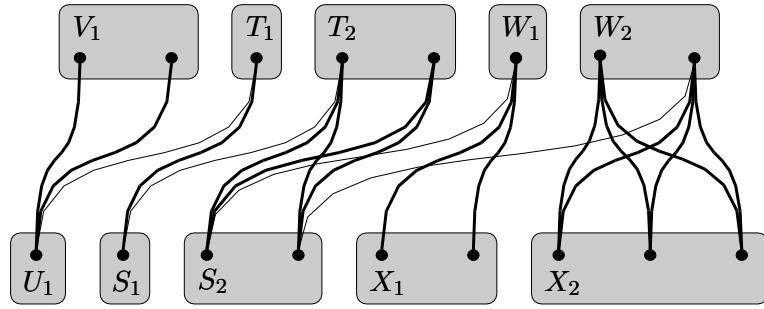


FIG. C.14: Décomposition canonique d'un graphe

Algorithme de décomposition

L'algorithme de décomposition d'un graphe biparti a été proposé par Dulmage et Mendelsohn [Dulmage *et al.* 1963]. Il se décompose en deux étapes. La première permet d'extraire de G les composantes semi-irréductibles H_i et L_i . La seconde étape consiste à décomposer le noyau de G . Cet algorithme est initialisé par l'identification d'un couplage maximal de G . Soit $(s_i, t_j)_{i \neq j}$ les arêtes non couplées du graphe et (s_i, t_i) les arêtes couplées.

Détermination des composantes semi-irréductibles [Dulmage *et al.* 1963]

Soit $G = [S \cup T, U]$, un graphe biparti. Soit le graphe disjoint $G_c = [S_c \cup T_c, U_c]$, sous graphe partiel de G , définissant un couplage maximal de G . Les sommets appartenant à S_c et à T_c sont donc tous saturés.

Soit $U_1 = [S_c \times (T - T_c)] \cap U$;

$i = 1$;

Tant que $U_i \neq \emptyset$ faire :

soit S_i et T_i , respectivement sous-ensembles de S et T tels que :

- $\forall s \in S_i, \exists u = (s, t) \in U_i \mid t \in T_i$,

- $\forall t \in T_i, \exists u = (s, t) \in U_i \mid s \in S_i$;

soit G_i, Ω_i et U_{G_i} respectivement sous-graphe de G_c , et sous-ensembles de T et U tel que :

$G_i = [S_i \cup \Omega_i, U_{G_i}]$ avec $|S_i| = |\Omega_i| = |U_{G_i}|$;

$U_{i+1} = [(S - \cup_{j=1}^i S_j) \times \Omega_i] \cap U$;

$i++$;

Fin faire

On obtient alors $A^* = \cup_i S_i$ et $B_* = T_c - \cup_i \Omega_i$. Or $[A^*, B_*]$ est un recouvrement extérieur minimal extrême de G .

Pour trouver le second recouvrement extérieur minimal extrême $([A_*, B^*],)$ de G , on peut utiliser le même algorithme en interchangeant les rôles des ensembles S et T .

La seconde étape est de décomposer les ensembles $(A_* \times \overline{B^*})$ et $(\overline{A^*} \times B_*)$ en composantes semi-irréductibles $L_i = [X_i \cup W_i, E_i]$ et $H_i = [U_i \cup V_i, E_i]$. Or, ces composantes sont des composantes connexes dans $((A_* \times \overline{B^*}) \cup (\overline{A^*} \times B_*)) \cap U$. La décomposition est donc immédiate.

Composantes fortement connexes du noyau de G

La recherche des composantes irréductibles du noyau de G est basée sur l'algorithme *depth first search* de Tarjan [Tarjan 1972]. Soit U_n l'ensemble des arêtes du noyau, $|U_n| = 2r$. Soit le graphe fondamental $G^* = [S_n \cup T_n, U_n]$ contenant le noyau de G . L'idée est de transformer le graphe G^* en un graphe direct G_d d'ordre r . Supposons que le couplage parfait soit tel que les arêtes saturées soient de la forme (s_i, t_i) et les arêtes insaturées de la forme $(s_i, t_j)_{i \neq j}$. Les arêtes $(s_i, t_j)_{i \neq j}$ insaturées de U_n deviennent des arcs (u_i, u_j) du graphe G_d . Il faut remarquer qu'il y a une bijection entre G^* et G_d . En effet, à chaque sommet u_i de G_d correspond deux sommets s_i et t_i et une arête saturée (s_i, t_i) et à chaque arc (u_i, u_j) correspond une arête (s_i, t_j) . Grâce à cette bijection, à un circuit de G_d correspond un chemin alterné de G^* .

Ensuite, il suffit de rechercher dans ce graphe les composantes fortement connexes. En effet, dans une composante fortement connexe de G_d , il existe au moins un circuit passant par toutes ces arêtes, ce qui est équivalent à un cycle alterné de G^* . Or, un cycle alterné est un sous-graphe irréductible. Donc une composante fortement connexe de G_d est une composante irréductible de G^* .

C.6.3 Gestion dynamique d'un graphe biparti.

Le graphe biparti est utilisé pour analyser, d'un point de vue structurel, un système d'équations. Ce système est amené à évoluer continuellement. La problématique est alors de savoir comment analyser un graphe après qu'il ait subi des modifications sans avoir à le traiter globalement. En d'autres termes, le problème est d'identifier le plus petit sous-graphe contenant les modifications.

Soit un graphe biparti $G = [S \cup T, U]$. Soit la décomposition canonique de ce graphe définie par les couples d'ensembles de sommets $\{X_i \cup W_i\}$, $\{S_i \cup T_i\}$, $\{U_i \cup V_i\}$. Le graphe est décomposé en sous graphes H_i , G_i , L_i . Les travaux de Dulmage et Mendelsohn sur la stabilité du graphe [Dulmage *et al.* 1959] permettent de répondre en partie à cette problématique. Leurs résultats sont résumés par les neuf propriétés suivantes :

Propriété 1 : Si un ensemble d'arêtes est ajouté à G tel que chaque arête lie deux sommets d'une même zone $\{X_i \cup W_i\}$, $\{S_i \cup T_i\}$ ou $\{U_i \cup V_i\}$ de $\{S \cup T\}$ alors la décomposition de G n'est pas altérée.

Propriété 2 : La suppression d'un ensemble d'arêtes de G telle que chaque arête appartienne à R_2 ne modifie pas la décomposition de G .

Propriété 3 : La suppression d'arêtes d'une composante irréductible G_i a les effets suivants :

- (1) si les arêtes restantes forment un graphe irréductible G'_i tel que $E(G_i) = E(G'_i)$, la structure de G est inchangée. La composante irréductible G_i est simplement remplacée par G'_i ;
- (2) si les arêtes restantes forment un graphe G'_i qui est tel que $E(G_i) = E(G'_i)$ mais G'_i est réductible. Il suffit de décomposer G'_i en composantes $H'_{(i)j}$, $G'_{(i)j}$, $L'_{(i)j}$ qui remplaceront G_i ;
- (3) si $E(G'_i) < E(G_i)$, alors la structure de la décomposition de G est fortement altérée. Mais, toutes les arêtes admissibles de $(G - G_i)$ restent admissibles et des arêtes de R_1 peuvent devenir admissibles.

Propriété 4 : La suppression d'un ensemble d'arêtes de sous-graphe semi-irréductible H_i (ou L_i) a les effets suivants :

- (1) si le sous-graphe résultant H_i^* est tel que $E(H_i^*) = E(H_i)$, alors il y a trois possibilités :
 - si H_i^* est semi-irréductible alors la structure de G est inchangée, excepté H_i qui est remplacé par H_i^* pouvant tout de même être décomposé en éléments semi-irréductibles minimaux $H'_{(i)j}$,
 - si H_i^* est irréductible alors il devient un composant G_i^* ,
 - si H_i^* est réductible alors les effets sont les mêmes que pour le cas (2) de la propriété 3;
- (2) si le sous-graphe résultant H_i' est tel que $E(H_i') < E(H_i)$, alors les effets sont les mêmes que pour le cas (3) de la propriété 3.

Propriété 5 : L'addition d'arêtes dans la région R_2 laisse la région R_1 inchangée. En effet, les arêtes ajoutées sont inadmissibles, elles ne modifient donc en rien la décomposition de G .

Propriété 6 : L'effet de l'addition d'arêtes dans R_4 . Soit (s, t) l'arête ajoutée, donc $s \in U_i$ et $t \in V_j$ avec $i \neq j$. Le sous graphe composé de l'arête (s, t) et des arêtes de H_i et H_j a pour unique recouvrement $[U_i \cup U_j, \emptyset]$. Alors, il ne peut pas être décomposé en sous graphes disjoints. Ce sous-graphe devient le sous-graphe H_k remplaçant les composantes H_i et H_j . Le reste de la décomposition de G n'est pas affecté. L'addition d'une arête dans R_5 a un effet similaire.

Propriété 7 : L'effet de l'addition dans G d'une arête de R_3^α .

Considérons le graphe fondamental G^* défini par les arêtes $U \cap (M \cup N)$ où $M = \cup S_i$ et $N = \cup_i T_i$ et considérons le sous graphe G° induit par les partitions M et N .

Soit $(s, t) \in S_i \times T_j$ l'arête à ajouter à G alors (S_i, T_j) est l'arête à ajouter à G° . Avant cette addition, G° n'avait pas de cycle alterné.

Si après l'addition, G° ne contient toujours pas un tel cycle, alors (S_i, T_j) est une arête inadmissible de G° et par conséquent (s, t) est une arête inadmissible de G . Cette addition n'affecte donc pas la décomposition canonique de G .

Dans l'autre cas, l'arête (S_i, T_j) fait partie d'un cycle alterné de G° . Supposons que ce cycle soit composé des nœuds $\{S_1, T_1, S_2, T_2, \dots, S_u, T_u\}$, alors le sous graphe fondamental de G° dont les arêtes sont contenues dans $\{(S_1, \dots, S_u) \times (T_1, \dots, T_u)\}$ est irréductible. Par conséquent, le sous graphe altéré de G composé des arêtes de $U \cap \{(S_1 \cup \dots \cup S_u) \times (T_1 \cup \dots \cup T_u)\}$ forme une composante irréductible qui remplacera l'ensemble des sous-graphes irréductibles G_1, \dots, G_u . Le reste du noyau n'est pas altéré.

Propriété 8 : L'effet de l'ajout d'une arête de R_3^β dans G .

Soit (s, t) l'arête ajoutée, $s \in S_i$, $t \in V_j$. Considérons le sous-graphe G^* induit par la décomposition canonique de G .

Soit $G^{\circ*}$ l'arbre alterné inférieur de racine S_i de G° .

Soit $\Upsilon = \{\cup(P \times Q), \forall(P, Q) \in G^{\circ*}\}$. Alors le sous-graphe fondamental composé des arêtes de $U \cap [\Upsilon \cup (s, t)]$ est une composante semi-irréductible minimale de type H_k de G .

Propriété 9 : L'effet de l'ajout d'une arête de R_3^γ dans G .

Il faut noter que la région R_3^γ est la seule région dans laquelle l'ajout d'une arête augmente la valeur $E(G)$. Soit (s, t) l'arête ajoutée, $s \in X_i$, $t \in V_j$. Considérons le sous-graphe composé des sommets de $L_i \cup H_j \cup \{s, t\}$. Ses

seuls couples extérieurs minimaux sont alors $[U_j \cup s, W_i]$ et $[U_j, W_i \cup t]$. Ce qui implique que dans le graphe modifié, les arêtes dont les extrémités sont s ou t autre que l'arête (s, t) sont inadmissibles. L'arête (s, t) devient un sous graphe irréductible de la forme G_k . Le sous graphe H_j a un sommet de moins, le sommet t . Si ce sous graphe modifié H_j est semi-irréductible ou irréductible, il devient un nouveau composant du noyau de G de la forme H_k ou G_k . Sinon, il est réductible, ces différents composants deviennent des nouveaux composants du noyau et ses arêtes supplémentaires sont des arêtes inadmissibles. Le sous graphe L_i subit un sort similaire.

Les transformations de la structure du graphe dans le cas de la suppression d'arêtes d'une composante irréductible G_i réduisant la dimension extérieure de celle-ci peuvent être précisés :

Soit $H'_{(i)j}$, $G'_{(i)j}$, $L'_{(i)j}$ la décomposition de G'_i en composantes irréductibles et semi-irréductibles. Soit G° le graphe induit du sous graphe $(G - G_i + H'_{(i)0})$. Soit V_0 et U_0 les ensembles de sommets de $H'_{(i)0}$. Soit A° l'arbre inférieur de racine U_0 dans G° . Soit $\Upsilon = \{\cup(P \times Q), \forall(P, Q) \in A^\circ\}$. Alors le sous-graphe fondamental composé des arêtes de $U \cap [\Upsilon \cup (s, t)]$ est une composante semi-irréductible minimale de type H_k de $(G - G_i + H'_{(i)0})$. L'opération est à répéter pour chaque graphe semi-irréductible $H'_{(i)j}$ dans le sous-graphe $(G - G_i + \cup_{k=1}^j H'_{(i)k})$. De même, on cherchera pour chaque composante $L'_{(i)j}$, les arbres alternés supérieurs dans le graphe induit de $(G - G_i + \cup_j H'_{(i)j} + \cup_{k=1}^j L'_{(i)k})$ afin de trouver le nouveau sous graphe semi-irréductible de type L_k .

C.7 Conclusion

Cette annexe a permis de définir le vocabulaire nécessaire à l'utilisation de la théorie de graphe. Une grande partie des algorithmes utilisés à la mise en place de l'approche proposée sont basés sur cette théorie. Les algorithmes classiques utilisés sont présentés dans cette annexe, alors que les algorithmes développés pour l'approche l'ont été dans la partie principale de ce manuscrit.

Annexe D

Les complexes cellulaires

Cet annexe a pour objectif de fournir les éléments de base de la théorie des complexes cellulaires.

D.1 Introduction

Le concept du complexe cellulaire est une notion connue en topologie combinatoire pour représenter la décomposition d'un sous ensemble singulier de R^n en une union disjointe de sous ensembles non singuliers.

La définition d'un complexe cellulaire [Rameau 1998] combine de façon récursive deux sortes d'objets : les *supports* et les *cellules*.

Un *support* est une sous variété différentielle connexe de R^n de dimension p (Cf. annexe A). Ici, ne sont utiles que l'espace R^3 (de dimension 3), les surfaces régulières (de dimension 2), les courbes (de dimension 1) et les points (de dimension 0). Tous ces éléments sont positionnés dans l'espace de travail grâce à la structure.

Une *cellule* est la restriction d'un support. Par définition, la dimension d'une cellule est celle de son support. Cette restriction est définie par un ensemble de cellules (le bord) de dimension strictement inférieure, elles-mêmes restrictions de supports, etc. Comme la dimension décroît strictement d'une cellule à son bord, la récursivité s'arrête à la dimension 0, c'est-à-dire aux points.

D.2 Complexe cellulaire : définition formelle

Un complexe cellulaire de R^n est une liste finie de sous ensembles de R^n $\{E_i, i = 1, m\}$, appelés des cellules, telles que :

1. les cellules E_i sont disjointes ;

2. chaque cellule E_i est une sous variété différentielle de dimension p_i de R^n , connexe par arcs et orientée. On appelle dimension de la cellule, la dimension de cette variété ;
3. la frontière de chaque cellule E_i est décomposée en d'autres cellules E_j de dimensions strictement inférieures. C'est-à-dire que pour chaque cellule E_i , il existe un sous ensemble d'indices $J(i) \subset \{1, \dots, m\}$ tel que

$$\widetilde{E}_i - E_i = \bigcup_{j \in J(i)} E_j$$

et

$$\dim E_j < \dim E_i \text{ pour tout } j \in J(i).$$

D.3 Graphe d'adhérence

Un complexe cellulaire est représenté par un graphe orienté. Les nœuds de ce graphe sont les cellules et les arcs représentent la relation d'adhérence.

Deux cellules E_i et E_j sont liées ($E_i \rightarrow E_j$) si la cellule E_j est adhérente à la cellule E_i . Cela signifie que la cellule E_j borde la cellule E_i , ou que la cellule E_i est bordée par la cellule E_j .

La relation d'adhérence est transitive : si E_k borde E_j et E_j borde E_i alors E_k borde E_i . Donc le graphe d'adhérence, tel qu'il est défini, est transitif. Pour faciliter son utilisation et éliminer des redondances inutiles, les liens transitifs ne sont jamais représentés.

Le graphe d'adhérence capte donc la relation "... est bordée par ..." entre les cellules.

Annexe E

Complément de géométrie différentielle

Cet annexe rappelle un ensemble de définitions nécessaires à la justification des outils d'analyses utilisés par le module de résolution. Ces définitions sont extraites du cours de M. Demazure [Demazure 1989]

Soient E un espace vectoriel de dimension fini égale à $\dim(E) = n$, U un ouvert de E , V une partie de E , a un point de V .

Système non-dégénéré On dit que des fonctions ϕ_1, \dots, ϕ_m , de classe C^∞ , définies au voisinage de a , forment un système *non-dégénéré* d'équations locales de V en a si les deux conditions suivantes sont satisfaites :

- il existe un ouvert U de E contenant a dans lequel les ϕ_i sont définies et tel que $V \cap U$ soit l'ensemble des points $x \in U$ avec $\phi_1(x) = 0, \dots, \phi_m(x) = 0$;
- les formes linéaires $d\phi_i(a)$ sont linéairement indépendantes.

Dimension, codimension On appelle *codimension* de V en a le rang de la matrice jacobienne d'un système d'équation en a et *dimension* de V en a l'entier $d = n - m$.

Sous variété de E On dit que V est une *sous-variété* de E en a si elle possède un système non-dégénéré d'équations locales en a ; on dit que V est une sous-variété de E s'il en est ainsi en chacun de ses points, c'est-à-dire si elle possède en chacun de ses points un système non-dégénéré d'équations locales, ou encore s'il existe une famille d'ouverts de E , recouvrant V , telle que V possède dans chaque ouvert de la famille un système non-dégénéré d'équations.

Partie négligeable Dans un espace \mathbb{R}^n , le cube de centre a et de côté r , ensemble des $\mathbf{x} = \{x_1, \dots, x_n\}$ avec $\sup_i |x_i - a_i| \leq r/2$, est de volume r^n . On dit qu'une partie A de \mathbb{R}^n est *négligeable* ou de *mesure nulle* si, pour tout $\epsilon > 0$, il existe une suite K_1, K_2, \dots de cubes, avec

$$A \subset \bigcup_i K_i \text{ et } \sum_i \text{vol}(K_i) < \epsilon.$$

Bibliographie

- [Ait-Aoudia *et al.* 1993] S. Ait-Aoudia, R. Jegou, D. Michelucci, *Reduction of constraint systems*, Proceedings of the Compugraphics Conference, Alvor, Portugal, pp. 83–92, 1993.
- [Ait-Aoudia 1994] S. Ait-Aoudia, *Modélisation géométrique par contraintes : quelques méthodes de résolution*, Thèse de Doctorat, Université de saint-Etienne, 1994.
- [Anderl *et al.* 1996] R. Anderl, R. Mendgen, *Modelling with constraints : theoretical foundation and application*, Computer-Aided Design, vol. 28, no. 3, pp. 155–168, 1996.
- [Bézier 1987] P. Bézier, *Courbes et surfaces*, Mathématiques et CAO, Volume 4, Éditions Hermès, 1987.
- [Bouhineau 1997] D. Bouhineau, *Construction automatique de figures géométriques & Programmation logique avec contraintes*, Thèse de Doctorat, Université Joseph Fourier, Grenoble, 1997.
- [Bouma *et al.* 1995] W. Bouma, I. Fudos, C. M. Hoffmann, J. Cai, R. Paige, *A Geometric Constraint Solver*, Computer-Aided Design, vol. 27, no. 6, pp. 487–501, 1995.
- [Brüderlin 1986] B. Brüderlin, *Constructing three-dimensional geometric objects defined by constraints*, Proceedings of the ACM Siggraph Workshop in Interactive 3D Graphics, Chapel Hill, North Carolina, pp. 111–129, 1986.
- [de Casteljaou 1986] P. de Casteljaou, *Formes à pôles*, Mathématiques et CAO, Volume 2, Éditions Hermès, 1986.
- [Cazier *et al.* 1996] D. Cazier, J.F. Dufourd, *Opérations booléennes robustes définies sous forme de systèmes de réécriture*, quatrièmes journées de l'Association Française d'Informatique Graphique, Dijon, 1996.
- [Cavalcanti *et al.* 1997] P.R. Cavalcanti, P.C.P. Carvalho, L.F. Martha, *Non-Manifold modeling : an approach based on spacial subdivision*, Computer-Aided Design, Vol. 29, no. 3, pp. 209–220, 1997.
- [Channac 1999] S. Channac, *Conception et mise en œuvre d'un système déclaratif de géométrie dynamique*, Thèse de Doctorat, Université Joseph Fourier, Grenoble, 1999.

- [Chauvat 1994] D. Chauvat, *Le projet VoluFormes : un exemple de modélisation déclarative avec contrôle spatial*, Thèse de Doctorat, Université de Nantes, 1994
- [Chou et al. 1987] S. C. Chou, W. F. Schelter and J. G. Yang, *Characteristic Sets and Gröbner Bases in Geometry Theorem Proving*, Proceedings of the workshop on Computer Aided Geometry Reasoning, pp. 29–56, INRIA, France, 1987.
- [Chung et al. 1989] J.C.H. Chung, M.D. Shussel, *Comparison of variational and parametric design*, Proceedings of the SME AutoFact'89 conference, Detroit, Michigan, USA, 1989.
- [Chung et al. 1999] J.C.H. Chung, T.-S. Hwang, C.-T. Wu, Y. Jiang, J.-Y. Wang, Y. Bai, H. Zou, *Extended Variational Design Technology – Foundation For Integrated Design Automation*, Proceedings of the fifth Symposium on Solid Modeling and Applications, Éditions ACM Press, pp. 13–22, 1999.
- [Clautrier 1991] M. Clautrier, *Difficultés de nouvelles approches de conception dans le spatial*, Séminaire GSIP, Méthodes et outils pour la conduite de projet, Grenoble, 1991.
- [Clément et al. 1994] A. Clément, A. Rivière, M. Temmerman, *Cotation tridimensionnelle des systèmes mécaniques – Théorie et pratique*, Éditions PYC, 1994.
- [Colin et al. 1997] C. Colin, E. Desmontils, J.-Y. Martin, J.-P. Mounier, *Working Modes with a Declarative Modeler*, Proceedings of the International Conference on Computational Graphics and Visualisation Techniques (CompuGraphics'97), Villamoura, Portugal, pp. 117–126, 1997.
- [Constant 1996] D. Constant, *Contribution à la spécification d'un modèle fonctionnel de produits pour la conception intégrée de systèmes mécaniques*, Thèse de Doctorat, Université Joseph Fourier, Grenoble, 1996.
- [Cubeles-Valade 1998] C. Cubeles-Valade, *Contribution à la synthèse des langages déclaratifs de spécifications géométriques*, Thèse de Doctorat, École Centrale de Paris, 1998.
- [Daniel 1995] M. Daniel, *Une première approche de la modélisation déclarative de courbes*, Rapport de recherche IRIN – 81, Nantes, 1995.
- [Demazure 1989] M. Demazure *Catastrophes et bifurcations* Collection ellipses, Éditions Marketing, 1989.
- [Desmontils et al. 1997] E. Desmontils, J.-Y. Martin, *Properties taxonomy in declarative modeling*, International Conference on Imaging Science, Systems and Technology (CISST'97), édité par H. R. Arabnia, Las Vegas, USA, pp. 130–138, 1997.

-
- [Dieudonné 1990] J. Dieudonné, *Fondements de l'analyse moderne*, Éléments d'analyse, Tome I, (Collection Cahiers scientifiques, Fascicule XXVIII), Éditions Gautier-Villars, 1990.
- [Döring *et al.* 1998] U. Döring, P. Michalik, B. Brüderlin, *A Constraint-Based Shape Modeling System*, Geometric Constraint Solving and Application, édité par B. Brüderlin, et D. Roller, Éditions Springer-Verlag, pp. 127–147, 1998.
- [Dulmage *et al.* 1958] A. L. Dulmage, N. S. Mendelsohn, *Coverings of bipartite graphs*, Canadian journal of mathematics, vol. 10, pp. 517–534, 1958.
- [Dulmage *et al.* 1959] A. L. Dulmage, N. S. Mendelsohn, *A Structure Theory of Bipartite Graphs of Finite Exterior Dimension*, Transactions of the Royal Society of Canada, Section III, vol. 53, pp. 1–13, 1959.
- [Dulmage *et al.* 1963] A. L. Dulmage, N. S. Mendelsohn, *Two Algorithms For Bipartite Graphs*, Journal of the Society of industrial applied mathematics, vol. 53, series III, section III, pp. 1–13, 1959.
- [Durand 1998] C. Durand, *Symbolic and numerical techniques for constraint solving*, Thèse de Doctorat, Université de Purdue, Indiana, USA, 1998.
- [Durand *et al.* 1999] C. Durand, C.M. Hoffmann, *Variational Constraints in 3D*, Proceedings of the International Conference on Shape Modeling and Application, Aizu, Japan, pp. 90–97, 1999.
- [Emiris *et al.* 1996] I. Z. Emiris, B. Mourrain, *Polynomial system solving : the case of a six-atom molecule*, Rapport de recherche no. 3075, INRIA, 1996
- [Fine 2001] L. FINE, *Processus et méthodes d'adaptation et d'idéalisation de modèles dédiés à l'analyse de structures mécaniques*, Thèse de Doctorat, Institut National Polytechnique de Grenoble, 2001.
- [Freeman-Benson *et al.* 1990] B. N. Freeman-Benson, J. Maloney, and A. Borning, *An Incremental Constraint Solver*, Communications of the ACM, vol. 33, no. 1, pp. 54–63, 1990.
- [Fudos 1995] I. Fudos, *Constraint solving for computer aided design*, Thèse de Doctorat, Université de Purdue, Indiana, USA, 1995.
- [Fudos *et al.* 1996–1] I. Fudos, C. M. Hoffmann, *Correctness proof of a geometric constraint solver*, International Journal of Computational Geometry & Applications, World Scientific Publishing Company, vol. 6, no. 4, pp. 405–420, 1996.
- [Fudos *et al.* 1996–2] I. Fudos, C. M. Hoffmann, *Constraint-based parametric conics for CAD*, Computer-Aided Design, vol. 28, no. 2, pp. 91–100, 1996.
- [Fudos *et al.* 1997] I. Fudos, C. M. Hoffmann, *A graph-constructive approach to solving systems of geometric constraints*, ACM Transactions on Graphics, vol. 16, no. 2, pp. 179–216, 1997.

- [Gao *et al.* 1998] X-S. Gao, S-C. Chou, *Solving geometric constraint systems. II. A symbolic approach and decision if Rc-constructibility*, Computer-Aided Design, vol. 30, no. 2, pp. 115-122, 1998.
- [Ge *et al.* 1999] J-X. Ge, S-C. Chou, X-S. Gao, *Geometric constraint satisfaction using optimization methods*, Computer-Aided Design, vol. 31, no. 14, pp. 867–879, 1999.
- [Godbillon 1971] C. Godbillon, *Éléments de topologie algébrique*, Éditions Hermann, Paris, France, 1971.
- [Gondran *et al.* 1985] M. Gondran, M. Minoux, *Graphes et algorithmes*, Collection de la direction des études et recherches d'électricité de France, Éditions Eyrolles, Paris, France, 1985.
- [Gueorguieva *et al.* 1994] S. Gueorguieva, D. Marcheix, *Non-Manifold Boundary Representation for Solid Modelling*, Proceedings of the International Computer Symposium, Univeristé nationale de Chiao Tung, Taiwan 1994.
- [Guillet 1999] S. Guillet, *Modification et construction de formes gauches soumises à des contraintes de conception*, Thèse de Doctorat, Institut National Polytechnique de Grenoble, 1999.
- [Hoffmann *et al.* 1987] C. Hoffmann, J Hopcroft, *The Potential Method for Blending Surfaces and Corners*, Geometric Modeling : Algorithms and New Trends édité par G.E. Farin, Éditions Siam, pp. 347–365, 1987.
- [Hoffmann 1994] C. Hoffmann, *Semantic Problems of Generative, Constraint-Based Design*, Parametric and variational design édité par J. Hoschev, W. Dankwort Éditions B. G. Teubner Verlag, pp. 37–46, 1994.
- [Hoffmann *et al.* 1994] C. M. Hoffmann, P. J. Vermeer, *Geometric constraint solving in R^2 and R^3* , Computing in euclidean geometry édité par D. Z. Du, et F. Hwang, Éditions World Scientific Publishing, seconde édition, pp. 266–298, 1994.
- [Hoffmann *et al.* 1997–1] C. M. Hoffmann, R. Joan-Arinyo, *Symbolic Constraints in Constructive Geometric Constraint Solving*, Journal of Symbolic Computation, Academic Press, vol. 23, no. 2/3, pp. 287–299, 1997. 1995.
- [Hoffmann *et al.* 1997–2] C. M. Hoffmann, A. Lomonosov, M. Sitharam, *Finding solvable subsets of constraint graphs*, Proceedings of Principles and Practice of Constraint Programming '97, Linz, Austria, LNCS 1330, Berlin, Springer-Verlag, pp. 463–477, 1997.
- [Hoffmann *et al.* 2001–1] C. M. Hoffmann, R. Joan-Arinyo, *Parametric Modeling*, Handbook of Computer Aided Geometric Design, Édition North Holland, 2001.
- [Hoffmann *et al.* 2001–2] C. M. Hoffmann, A. Lomonosov, M. Sitharam, *Decomposition Plans for Geometric Constraint Problems, Part I & II*, Journal of Symbolic Computation, Academic Press, vol. 31, pp. 367–427, 2001.

-
- [Hopcroft *et al.* 1973] J. E. Hopcroft, R. M. Karp *An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs*, SIAM Journal of Computer, vol. 2, no. 4, pp. 225-231, 1973.
- [ISO 128] *Dessins techniques – Principes généraux de représentation*
- [ISO 129] *Dessins techniques – Cotation – Principes généraux, définitions, méthodes d'exécution et indications spéciales*, 1985.
- [ISO 4287] *Spécification géométrique des produits (GPS) – État de surface : Méthode du profil – Termes, définitions et paramètres d'état de surface*, 1997.
- [ISO 5458] *Spécification géométrique des produits (GPS) – Tolérancement géométrique – Tolérancement de localisation*, 1998.
- [ISO 8785] *Spécification géométrique des produits (GPS) – Imperfections de surface – Termes, définitions et paramètres*, 1998.
- [ISO/TR 14638] *Spécification géométrique des produits (GPS) – Schéma directeur*, 1995.
- [ISO 14660–1] *Spécification géométrique des produits (GPS) – Éléments géométriques – Partie 1 : Termes généraux et définitions*, 1999.
- [ISO 10303] *Systèmes d'automatisation industrielle et intégration – Représentation et échange de données de produits*
- [Jackson 1995] D.J. Jackson, *Boundary Representation Modelling With Local Tolerances*, Proceedings of the third Symposium on Solid Modeling and Applications, Éditions ACM Press, pp. 247–253, 1995.
- [Juan-Arinyo *et al.* 1997] R. Juan-Arinyo, A. Soto, *A correct rule-based geometric constraint solver*. Computer and Graphics, vol. 21, no. 5, pp. 599–609, 1997.
- [Juan-Arinyo *et al.* 1999] R. Juan-Arinyo, A. Soto-Riera, *Combining Constructive and Equational Geometric Constraint Solving Techniques*, ACM transactions on graphics, vol. 18, no. 1, pp. 35–55, 1999.
- [Korn *et al.* 1961] G. A. Korn, T. M. Korn, *Mathematical handbook for scientists and engineers*, Éditions Mc Graw-Hill book company, 1991.
- [Kwaiter *et al.* 1998] G. Kwaiter, V. Gaildrat, R. Caubet, *Modelling with Constraints : A Bibliographical Survey*, Proceedings of the International Conference on Information Visualisation, IV'98, London, UK, 1998.
- [Kramer 1991] G. A. Kramer, *Using Degrees of Freedom Analysis To Solve Geometric Constraint Systems*, Proceedings of the first Symposium on Solid Modeling and Applications, Éditions ACM Press, pp. 371–378, 1991.
- [Lamure *et al.* 1995] H. Lamure, D. Michelucci, *Solving geometric constraints by homotopy*, Proceedings of the third Symposium on Solid Modeling and Applications, Éditions ACM Press, pp. 263–269, 1995.

- [Lamure *et al.* 1998] H. Lamure, D. Michelucci, *Qualitative Study of Geometric Constraints*, Geometric Constraint Solving and Application édité par B. Brüderlin, et D. Roller, Éditions Springer-Verlag, pp. 134–145, 1998.
- [Latham *et al.* 1996] R. Latham, A. Middleditch, *Connectivity analysis : a tool for processing geometric constraints*, Computer-Aided Design, vol. 28, no. 11, pp. 917–928, 1996.
- [Lee 1987] E.T.Y. Lee, *The Rational Bézier Representation for Conics*, Geometric Modeling : Algorithms and New Trends édité par G.E. Farin, Éditions Siam, pp. 3–19, 1987.
- [Lee *et al.* 1996] J. Y. Lee, K. Kim, *Geometric reasoning for knowledge-based parametric design using graph representation*, Computer-Aided Design, vol. 28, no. 10, pp. 831–841, 1996.
- [Lee *et al.* 1998] J. Y. Lee, K. Kim, *A 2D geometric constraint solver using DOF-based graph reduction* Computer-Aided Design, vol. 30, no. 11, pp. 883–896, 1998.
- [Lee *al.* 2001] S.H. Lee, K. Lee, *Partial Entity Structure : A Compact Non-Manifold Boundary Representation Based on Partial Topological Entities*, Proceedings of the sixth Symposium on Solid Modeling and Applications. Éditions ACM Press, pp. 159–170, 2001.
- [Lefebure de fourcy 1836] Lefebure de fourcy, *Éléments de trigonométrie*, Éditions Bachelier, Libraire de l'école polytechnique, Paris, 1836.
- [Lehmann *et al.* 1982] D. Lehmann, C. Sacré *Géométrie et Topologie des Surfaces*, Éditions Presses Universitaires de France, 1982.
- [Léon 1991] J-C. Léon, *Modélisation et construction de surfaces pour la CFAO*, Éditions Hermès, 1991.
- [Léon *et al.* 1995] J-C. Léon, P. Trompette, *A new approach towards free-form surfaces control*, Computer Aided Geometric Design, vol. 12, pp. 395–416, 1995.
- [Lienhardt 1992] P. Lienhardt, *Extension de la notion de carte et modélisation à base topologique*, Thèse de Doctorat, Université Louis Pasteur, Strasbourg, 1992.
- [Light *et al.* 1982] R. A. Light, D. C. Gossard, *Modification of geometric models through variational geometry*, Computer-Aided Design, vol. 14, no. 4, pp. 209–214, 1982.
- [Lin *et al.* 1981] V. C. Lin, D. C. Gossard, R. A. Light. *Variational Geometry in Computer Aided Design*, ACM Computer Graphics, vol. 15, no. 3, pp. 171–177, 1981.
- [Lucas *et al.* 1995] M. Lucas, E. Desmontils, *Les modeleurs déclaratifs*, Revue Internationale de CFAO et d'informatique graphique, Hermès, Vol. 10, no. 6, pp. 559–585, 1995.

-
- [Lukács 1989] G. Lukács, *The Generalised Inverse Matrix and The Surface-Surface Intersection Problem*, Theory and Practice of Geometric Modeling édité par W. Staßer et H.P. Seidel, Éditions Springer-Verlag, pp. 167–185, 1989.
- [Marcheix *et al.* 1995] D. Marcheix, S. Gueorguieva, *Topological Operators for Non-Manifold Modeling*, Proceedings of the Third International Conference in Central Europe on Computer Graphics and Visualisation, Université de West Bohemia, Plzen, République Tchèque pp. 173–186, 1995
- [Mata 1997] N. Mata, *Solving incidence and tangency constraints in 2D*, Technical Report LSI-97-3R, Department LiSI, Université polytechnique de Catalunya, Barcelona, 1997.
- [Mathis 1997] P. Mathis, *Constructions géométriques sous contraintes en modélisation à base topologique*, Thèse de Doctorat, Université Louis Pasteur, Strasbourg, 1997.
- [Maza 1998] S. Maza, *Méthode de génération de maillages multirésolution non imbriqués : application à la simulation d'éclairage*, Thèse de Doctorat, Institut National Polytechnique de Grenoble, 1998.
- [Morgan 1987] A. Morgan, *Solving polynomial systems using continuation for engineering and scientific problems*, Éditions Prentice-hall inc., 1987,
- [Murayama 1996] H. Murayama, *Generalization of Regularized Boolean Set-operations for Nonmanifolds*, Proceedings of the CSG-96, Set-theoretic Solid Modelling Techniques and Applications, Éditions Information Geometers Ltd, Winchester, UK, pp. 159–181, 1996.
- [Murota 1987] K. Murota, *Systems Analysis by Graphs and Matroids, Structural Solvability and Controllability*, Édition Springer Verlag, 1987.
- [Nelson 1985] G. Nelson, *Juno, a constraint-based graphics system*, Proceedings of SIGGRAPH85, vol. 19, no. 3, pp. 235–243, San Francisco, 1985.
- [Nguyen *et al.* 1996] C. Nguyen, J-C. Lafon, *Constraint-based design of exact parameterized 3D solids*, Proceedings of the CSG-96, Set-theoretic Solid Modelling Techniques and Applications, Éditions Information Geometers Ltd, Winchester, UK, pp. 49–63, 1996.
- [Noël 1994] F. Noël, *Mailleur auto-adaptatif pour des surfaces gauches en vue de la conception intégrée*, Thèse de Doctorat, Institut National Polytechnique de Grenoble, 1994.
- [Owen 1991] J. C. Owen, *Algebraic solution for geometry from dimensional constraints*, Proceedings of the first Symposium on Solid Modeling and Applications, Éditions ACM Press, pp. 397–407, 1991.

- [Owen 1996] J. C. Owen, *Constraints on simple geometry in two and three dimensions*, International Journal of Computational Geometry & Applications, World Scientific Publishing Company, vol. 6, no. 4, pp. 421–434, 1996.
- [Pernot *et al.* 2002] J.-P. Pernot, S. Guillet, J.-C. Léon, F. Giannini, C. Catalano, B. Falcidieno, *Deformation of design and styling features using character lines*, Proceedings of the International Conference IDMME 2002, Clermont-Ferrand, France, 2002.
- [Pernot *et al.* 2002] J.-P. Pernot, S. Guillet, J.-C. Léon, F. Giannini, C. Catalano, B. Falcidieno, *A shape deformation to model character lines in the early design phases*, Proceedings of the IEEE International Conference on Shape Modelling, Banff, Canada, 2002.
- [Prudhomme 1999] G. Prudhomme, *Le processus de conception de systèmes mécaniques et son enseignement*, Thèse de Doctorat, Université Joseph Fourier, Grenoble, 1999.
- [Rameau 1998] J-F. Rameau, *Modélisation des solides en Conception Assistée par Ordinateur*, Institut Universitaire Professionnel de Paris-Dauphine, Génie Mathématique et Informatique, 1998.
- [Requicha 1980] A.A.G. Requicha, *Representation for rigid solids : Theory, Methods and Systems*, Éditions ACM Computing Surveys, Vol. 12, no. 4, pp. 437–464, 1980.
- [Rivière 1993] A. Rivière, *La géométrie du groupe des déplacements appliquée à la modélisation du tolérancement*, Thèse de Doctorat, École Centrale de Paris, 1993.
- [Rossignac *et al.* 1999] J. Rossignac, D. Cardoze, *Matchmaker : Manifold BReps for non-manifold r-sets*, Proceedings of the fifth Symposium on Solid Modeling and Applications, Éditions ACM Press, pp. 31–41, 1999.
- [Rotgé 1996] J.F. Rotgé, *Principles of solid geometry design logic*, Proceedings of the CSG-96, Set-theoretic Solid Modelling Techniques and Applications, Éditions Information Geometers Ltd, Winchester, UK, pp. 233–254, 1996.
- [Roucoules 1999] L. Roucoules, *Méthodes et connaissances : contribution au développement d'un environnement de conception intégrée*, Thèse de Doctorat, Institut National Polytechnique de Grenoble, 1999.
- [Sakkalis *et al.* 2000] T. Sakkalis, G. Shen, N.M. Patrikalakis, *Representational validity of boundary representation models*, Computer-Aided Design, Vol. 32, no. 12, pp. 719–726, 2000.
- [Sannella 1994] M. Sannella, *SkyBlue : A multi-Way Local Propagation Constraint Solver for User Interface Construction*, Proceedings of the ACM SIGGRAPH Symposium on User Interface Software and Technology, Marina Del Rey, California, 1994.

-
- [Serrano *et al.* 1986] D. Serrano, D. Gossard, *Combining mathematical models and geometric models in CAE systems*, Proceedings of ASME Computers in Engineering Conference, Éditions ASME, pp. 277–284, 1986.
- [Serrano 1991] D. Serrano, *Automatic Dimensioning in Design for Manufacturing*, Proceedings of the first Symposium on Solid Modeling and Applications, Éditions ACM Press, pp. 379–386, 1991.
- [Serré 2000] P. Serré, *Cohérence de la spécification d'un objet de l'espace euclidien à n dimensions*, Thèse de Doctorat, École Centrale de Paris, 2000.
- [Sridhar *et al.* 1996] N. Sridhar, R. Agrawal, g. L. Kinzel, *Algorithms for the structural diagnosis and decomposition of sparse underconstrained design systems*, Computer-Aided Design, vol. 28, no. 4, pp. 237–249, 1996.
- [Sutherland 1963] I.E. Sutherland, *Sketchpad : A man-machine graphical communication system*, Proceedings of the IFIP Spring Joint Computer Conference, pp. 329–346, 1963.
- [Takahashi *et al.* 1997] S. Takahashi, Y. Shinagawa, T. L. Kunii, *A Feature-based Approach for Smooth Surfaces*. Proceedings of the fourth Symposium on Solid Modeling and Applications, Éditions ACM Press, pp. 97–110, 1997.
- [Tarjan 1972] R. Tarjan, *Depth-first search and linear graph algorithms*, SIAM Journal of Computer, vol. 1, no. 2, pp. 146–160, 1972.
- [Tichkiewitch *et al.* 1993] S. Tichkiewitch, H. Tiger, A. Jeantet *Ingénierie Simultanée dans la conception de produits*, Université d'été du pôle productique Rhône Alpes, Aussois, 1993.
- [Tichkiewitch 1995] S. Tichkiewitch, *Analyse Numérique*, Ecole Nationale Supérieure d'Hydraulique et Mécanique de Grenoble, Institut National Polytechnique de Grenoble, Partie 1, 1995.
- [Tichkiewitch *et al.* 1995] S. Tichkiewitch, E. Chapa, P. Belloy *Un modèle produit multi-vues pour la conception intégrée*, Proceedings of the International industrial engineering conference on the Productivity in a world without borders, Montréal, 1995.
- [Troussier 1999] N. Troussier, *Contribution à l'intégration du calcul mécanique dans la conception de produits techniques : proposition méthodologique pour l'utilisation et la réutilisation*, Thèse de Doctorat, Université Joseph Fourier, Grenoble, 1999.
- [Varady *et al.* 1989] T. Varady, R.R. Martin, J. Vida, *Topological Considerations in Blending Boundary Representation Solid Models*, Theory and Practice of Geometric Modeling édité par W. Staßer et H.P. Seidel, Éditions Springer-Verlag, pp. 205–220, 1989.

- [Véron 1997] P. Véron, *Techniques de simplification de modèles polyédriques pour un environnement de conception mécanique*, Thèse de Doctorat, Institut National Polytechnique de Grenoble, 1997.
- [Verroust *et al.* 1992] A. Verroust, F. Schonek, D. Roller, *Rule-oriented method for parameterized computer-aided design*, *Computer-Aided Design*, vol. 24, no. 10, pp. 531–540, 1992.

TITLE

A dynamic model of engineering specifications based on a variational geometry approach.

SUMMARY

For several years, CAD systems have adopted a variational approach for product modeling. This choice is justified by users' needs. Effectively, a product is often designed to fulfill requirements that are not expressed as geometric specifications. In addition, the geometric model of the product evolves continuously during the design process. During this process, the software must be able to preserve the coherence of the model specifications. To fit into this context, a new CAD system architecture is proposed. It decouples geometric modeler data from engineering solver data. This architecture is divided into a set of geometric specifications, a variational geometric modeler and an engineering solver.

This set defines the user's specifications into three representations. The first one is a classical feature representation. The second one is an original geometric representation. The third one represents all the specifications from an algebraic point of view. This algebraic model is dedicated to the engineering solver. The strength of this approach lies in the geometric representation. Indeed, the main idea of this representation is to describe the position of each element in a natural way, i.e. an element is known only with its dimensions and/or its position with regard to other elements. As a result, a well-suited set of algebraic equations is obtained.

The structure of the solver is outlined and followed by the introduction of the analysis and solving modules that are clustered into three tasks : a pre-solver analysis, a solver, a post-solver analysis in case of resolution failure. These tasks are performed by distinct software modules which will be described to show how the solving is optimized through a pre-analysis and a decomposition phase using bipartite graphs, different solutions can be identified and obtained, in case of failure of the solving process, a significant diagnosis can be provided to the user.

The document ends by a set of examples of increasing order of complexity. They describe progressively the sequences of resolution steps.

KEYWORDS

CAD, geometric modeling, geometric modeler, feature modeler, geometric solver, algebraic solver, engineering solver, specification analysis, bipartite graph.

Thèse de doctorat de l'Institut National
Polytechnique de Grenoble

**Un modèle dynamique de spécifications
d'ingénierie basé sur une approche de
géométrie variationnelle**

David LESAGE

RÉSUMÉ

Le travail de cette thèse porte sur un modèle de spécifications géométriques et d'ingénierie permettant d'assurer le maintien de la cohérence de ces spécifications. Ce travail se situe au coeur des modèles produits dans le domaine de la Conception Assistée par Ordinateur (CAO).

L'apport majeur de ce travail consiste en une architecture originale d'un système de CAO variationnel permettant une gestion dynamique des spécifications. Elle permet de découpler les parties modèle géométrique et solveur du modèle produit et ainsi, offre aux concepteurs mécaniques une véritable application de CAO variationnelle.

Pour mettre en place une telle approche, un modèle d'information et un solveur d'ingénierie sont proposés. Le modèle d'information décrit au plus haut niveau les spécifications du produit, à un niveau intermédiaire la géométrie et un niveau inférieur une version algébrique de l'ensemble des spécifications. Ce modèle algébrique est destiné au solveur d'ingénierie. L'originalité de cette approche apparaît dans le modèle géométrique. En effet, afin de faciliter une mise en équations adéquate de la partie géométrique du problème, les éléments sont positionnés, non pas dans un repère absolu mais les uns par rapport aux autres.

Le solveur d'ingénierie proposé se décompose en trois modules : un module d'analyse et de décomposition en sous-problèmes aussi indépendants que possible ; un module de résolution autorisant une recherche de toutes les solutions possibles et un module d'analyse d'échecs pour offrir un diagnostic pertinent à l'utilisateur. Le principe de décomposition utilise une modélisation à partir de graphes bipartis.

Un ensemble d'exemples illustrent cette approche et démontrent progressivement l'enchaînement des étapes sur des problèmes de plus en plus complexes.

MOTS CLÉS

CAO, modélisation géométrique, modèleur géométrique, modèle d'information, modèle produit, solveur géométrique, solveur algébrique, solveur d'ingénierie, analyse des spécifications, graphe biparti.

Laboratoire Sols, Solides et Structures

Domaine Universitaire, BP 53, 38041 GRENOBLE Cedex 9, France