

Université de Montréal

**La structure algorithmique de la représentation
géométrique d'un dôme entrelacé**

Par

Jamil Moussa

Faculté de l'aménagement

Mémoire présenté à la Faculté des études supérieures
en vue de l'obtention du grade de
Maître ès sciences appliquées (M.Sc.A.)
en aménagement

Option

conception, modélisation et fabrication assistées par ordinateur

Juin 2002

© Jamil Moussa, 2002



NA
9000
W54
2002
N. 015



Université de Montréal
Faculté des études supérieures

Ce mémoire intitulé:
**La structure algorithmique de la représentation
géométrique d'un dôme entrelacé**

présenté par :
Jamil Moussa

a été évalué par un jury composé des membres suivants:

Temy Tidafi
.....
président-rapporteur

Claude Parisel
.....
directeur de recherche

Jean-François Rotgé
.....
codirecteur

Manon Guité
.....
membre du jury

Le mémoire accepté le:..... 26 juin 2002

Résumé et mots clés

Mots clés: représentation géométrique, model mathématique, structure algorithmique, modélisation d'actions, objet physique, objet virtuel et bibliothèque numérique.

Le terme « représentation » est très usité, pour ainsi dire, dans presque tous les domaines. Les méthodes et les outils utilisés dans la représentation d'un objet sont variables et dépendent de l'objectif de la représentation. Toutefois, les domaines et les champs d'applications suivent différentes méthodes afin d'arriver à une représentation fiable. En mathématiques, en géométrie, comme en design architectural, le même objet peut être représenté de différentes manières. (Par exemple: une équation, un dessin, une maquette, etc...). Il est certain que la compréhension d'un objet est assurée par ses représentations (géométriques, structurales, etc.). Celles-ci sont le résultat final de l'ensemble des opérations de modélisation qui amènent l'objet d'un état abstrait, c'est-à-dire son idée conceptuelle sous-jacente à la forme géométrique, à son état formel.

Plusieurs problèmes surgissent lors de la modélisation d'un objet. La géométrie pose le problème de la définition numérique de la forme spatiale de l'objet dans l'espace. Les outils se portent sur le problème de la méthode de modéliser l'objet en lui-même et de son interprétation en forme modélisée. L'informatique permet actuellement de résoudre ce genre de problèmes. L'objet modélisé grâce à des moyens informatiques proviennent de sources différentes types d'informations: géométriques, constructives, historiques, stylistiques et/ou physiques. Ces informations peuvent être complexes et découlent de sources différentes. Elles peuvent être aussi regroupées d'une manière hétérogène. Il convient donc de trouver un environnement adéquat nous permettant d'organiser ces données dans le cadre d'une structure visant à une manipulation intelligente de ces dernières.

L'objectif de cette recherche est d'établir une structure informatique ou algorithmique capable de générer un objet, de le représenter d'une manière réaliste et d'en dériver de nouvelles formes de la même famille géométrique. Dans le cadre de cette structure, il est possible de créer une bibliothèque numérique (une base de données) susceptible d'insérer ses objets numériques (codes) dans un autre contexte architectural en voie de modélisation.

Cette recherche porte donc essentiellement sur la compréhension, le développement et l'approfondissement des méthodes et des outils de modélisation. Il s'agit, en fait, de parvenir à une compréhension (analyse) et une interprétation (objet modélisé). Pour y parvenir, nous avons adopté une approche théorique fondée sur une analyse globale historique, constructive et géométrique et une approche pratique basée sur l'utilisation de l'informatique dans la modélisation. Les données théoriques sont analysées et soumises à une étude comparative afin de les caractériser (organiser) dans le cadre d'un modèle numérique. Celui-ci est moulé dans une structure algorithmique (informatique) qui est, à son tour, traduite en programme. Les tâches exécutées par ce programme dans le cadre de l'expérimentation assurent la validité (interne et externe) de la recherche dans les domaines d'application appropriés. Les résultats obtenus sont analysés en vue d'optimiser le programme et de développer, voire d'améliorer les méthodes de modélisation.

Résumé and key words (Résumé et mot clés en anglais)

Key words: geometrical representation, mathematical model, algorithmic structure, actions modelling, physical object, virtual object and numerical library.

The term "representation" is well known, almost in all the fields. The methods and the tools used in the representation of an object are various and depend on the goal of the representation. However, the domains and the fields of applications follow various methods in order to arrive to a reliable representation. In mathematics, in geometry, as in architectural design, the same object can be represented in different manners. (For example: an equation, a drawing, a model, etc.). It is certain that the comprehension of an object is ensured by its representations (geometrical, structural, etc). Those are the final result of the ensemble of the modeling operations, which bring the object from its abstracted state (its conceptual idea subjacent with the geometrical form, toward its formal state.

Several problems emerge during the object modeling. The geometry poses the problem of the numerical definition of the object's spatial form in space. The tools add to the problem of the method of object modelling in itself and its interpretation in modelled form. The computer currently makes it possible to solve this kind of problem. The modelled object, thanks to computer processing, comes with standard information from various sources: geometrical, constructive, historical, stylistics and/or physics. This information can be complex and arise from different sources. It can also be gathered in a heterogeneous manner. It is thus advisable to find an adequate environment allowing us to organize this data within the framework of an algorithmic structure aiming at intelligent data handling.

The objective of this research is to establish a data processing or algorithmic structure able to generate an object, to represent it in a realistic manner and to derive new forms in the same geometrical family. Within the framework of this structure, it is possible to create a numerical library (a data base) that will likely allow the insertion of its numerical objects (code) in another architectural context, which is in the process of modeling.

This research thus relates primarily to the comprehension, the development and the profundity of the methods and the tools of modeling. It means, in fact, to arrive to a comprehension (analyzes) and an interpretative (modelled) object. For that purpose, we adopted a theoretical approach based on a historical, constructive and geometrical analysis and a practical approach based on the utilisation of computer modeling. The theoretical data is analyzed and subjected to a comparative study in order to characterize (organize) it within the framework of a numerical (mathematical) model. This model is moulded in an algorithmic structure (data-processing) that is, in turn, translated into a program. The tasks carried out by this program within the framework of the experimentation ensure the validity (internal and external) of the research in the suitable application fields. The results obtained are then analyzed in order to optimize the program and to develop the methods of modeling.

Table des matières	Page
Résumé et mots clés.....	i
Résumé and key words.....	ii
Table des matières.....	iii
Liste des figures.....	v
Remerciement.....	viii
CHAPITRE-I-----	
1-Introduction.....	2
2-L'arc et la voûte dans l'antiquité.....	3
3-Description du sujet.....	4
4-Définitions et mots-clefs dans le contexte de l'étude.....	6
5-Problématique.....	7
6-La compréhension géométrique d'une surface.....	9
6.1-La représentation mathématique d'une surface	9
6.2-La représentation procédurale d'une surface	9
6.3-Transformation géométrique d'une surface et/ou d'un type surfacique.....	12
7-Objectif et méthode de modélisation.....	15
7.1-Objectif de modélisation.....	15
7.2-Méthode de modélisation.....	16
Conclusion du chapitre-I.....	18
CHAPITRE-II-----	
1-Le choix du cas type.....	20
2-Analyse globale des dômes entrelacés.....	21
2.1- Analyse historique.....	22
2.1.A- Histoire du développement de la voûte.....	22
2.1.B- la famille des dômes entrelacés.....	27
2.1.C-Les relations topologiques entre les arcs et les dômes.....	31
2.2-Analyse des règles d'assemblage (constructives).....	32
2.2.A-Règle d'assemblage de dômes entrelacés avec une fréquence constante $n=1$	32
2.2.B-Règle d'assemblage de dômes entrelacés avec une fréquence variable $n=$ variable	33
2.2.C- Résumé de la règle d'assemblage (constructive) des dômes entrelacés.....	34
2.3-Analyse géométrique des arcs.....	35
2.3.A- Analyse géométrique des arcs subissant des discontinuités surfaciques.....	36
2.3.A.1-Étude des trois cas de la position du centre $C1$ par rapport au centre C	38
2.3.A.2-Étapes de construction de l'arc subissant des discontinuités surfaciques.....	41
2.3.A.2.I-Étape-1: Étudier les relations de la construction verticales.....	42
2.3.A.2.II-Étape-2: Étudier les relations de la construction horizontale.....	44

2.3.A.2.III-Étape-3: Rétablir la géométrie des dômes entrelacés.....	47
2.3.A.3-Analyse géométrique des cas particuliers.....	53
2.3.A.3.I-Analyse géométrique de la règle gothique.....	53
2.3.A.3.II-Analyse géométrique de la règle islamique.....	55
2.3.B-Analyse géométrique des dômes entrelacés dont l'arc de base est elliptique.....	57
3-Étude contextuelle.....	59
3.1-Les paramètres et les données.....	60
3.2-Établissement un modèle logique.....	61
3.3-Établissement un modèle mathématique, numérique.....	62
Conclusion du chapitre-II.....	64
CHAPITRE-III-----	
1-Le choix des outils de travail.....	66
1.1-Critères de choix et motivations.....	66
1.2-La base des des outils informatiques (système géométrique et langage informatique).....	68
2-Survol de la technologie SGDL.....	71
3-Structure informatique.....	73
Conclusion du chapitre-III.....	75
CHAPITRE-IV-----	
1-Expérimentation et domaines d'applications.....	77
2-Expérimentation et test de validité du programme «prog_pointu».....	78
2.A-Combinaison des fonctions et tâches du programme « prog_pointu ».....	81
3-Expérimentation et test de validité du programme «prog-ellipse».....	87
3.A-Combinaison des fonctions et tâches du programme « prog_ellipse ».....	89
4-Insertion des dômes entrelacés dans un contexte architectural.....	92
4.1-Construction du contexte architectural.....	93
4.2-Dérivation des nouvelles formes du contexte architectural modélisé.....	95
4.3-Configuration surfacique des dômes entrelacés (remplissage).....	96
Conclusion du chapitre-IV.....	97
Conclusion générale.....	98
Bibliographie.....	101
Annexe-couple: (la modélisation du contexte architectural).....	ii
Annexe-A: (les fonctions des dômes entrelacés dont l'arc de base est pointu).....	xviii
Annexe-B: (les fonctions des dômes entrelacés dont l'arc de base est elliptique).....	xxxv
Annexe-C: (les fonctions du contexte architectural).....	lvii
Annexe-lisp: (les fonctions de transformations géométriques).....	xliii

Liste des figures

Figure	Titre de figures	Page
1	Dôme d'arête.....	3
2	Dôme pendentif.....	3
3	Dôme entrelacé.....	3
4	Schéma de modélisation.....	5
5	Schéma montrant la représentation d'un objet par catégorie d'informations.....	5
6	La génération d'une surface par translation.....	9
7	La génération d'une surface par rotation.....	10
8	La génération surfacique par processus.....	11
9	Les trois types de courbes gaussiennes.....	12
10	La transformation géométrique d'un type de surface.....	14
11	La déformation continue d'une surface.....	14
12	Arc assyrien.....	22
13	Voûte persane (ovale)	23
14	La méthode assyrienne de la construction de voûte (l'encorbellement)	23
15	Un dôme souterrain construit par les Mycéniens en XIII,ième siècle av.J.C.....	24
16	Le développement de berceau chez les romains	25
17	Dôme byzantin pendentif.....	26
18	Dôme d'arête (romain).....	26
19	La construction d'un pendentif.....	26
20	L'arc islamique.....	27
21	Arc islamique plus qu'un demi-cercle	28
22	Le dôme entrelacé de la grand Mosquée de Cordoue.....	29
23	Les types des dômes entrelacés.....	29
24	La distribution spatiale des arcs entrelacés.....	30
25	La distribution linéaire des arcs entrelacés.....	30
26	Tableau de mosaïque.....	30
27	Topologie de dômes	31
28	L'alternatif de sommets.....	32
29	Fréquence constante des dômes entrelacés.....	32
30	Fréquence variable des dômes entrelacés.....	33
31	Exemples des dômes entrelacés avec une fréquence variable.....	34
32	Les configurations d'un arc pointu.....	36
33	Plans de la construction de l'arc pointu.....	37

Figure	Titre de figures	Page
34	La construction d'un arc pointu.....	38
35	La construction d'un arc pointu (demi-cercle).....	39
36	La construction d'un arc pointu (brisé).....	40
37	Les étapes de la construction de l'arc pointu.....	41
38	La construction verticale de l'arc pointu.....	42
39	La construction verticale de l'arc pointu (illustration en 2D).....	43
40	Les trois cas de la construction horizontale d'un arc pointu.....	44
41	Construction horizontale (illustration 3D).....	45
42	Construction horizontale (illustration 2D).....	46
43	Rétablir la géométrie du dôme entrelacé.....	47
44	Les paramètres et les variable	48
45	Le principe géométrique du rétablissement de la position de l'arc de base.....	49
46	Rétablir la géométrie des dômes entrelacés (illustration 3D).....	51
47	Rétablir la géométrie des dômes entrelacés (illustration 2D).....	52
48	Les données graphiques de l'arc gothique	53
49	La règle gothique.....	54
50	Les divisions de l'arc islamique.....	55
51	La règle islamique.....	56
52	La méthode de tracé d'un arc demi-ellipse.....	58
53	Schéma d'étude contextuelle.....	59
54	Les paramètres de la construction de l'arc de base.....	60
55	Schéma du modèle logique.....	61
56	Schéma du modèle numérique.....	62
57	Le modèle mathématique (numérique).....	63
58	Méthode de tracé d'une ellipse.....	67
59	La représentation géométrique d'une quadrique	72
60	Schéma de la structure informatique	73
61	Le domaine numérique de l'angle G_a	78
62	Les types des dômes entrelacés générés par le programme (prog_pointu).....	79
63	Le génération d'un arc pointu selon la règle gothique.....	80
64	Le génération d'un arc pointu selon la règle islamique.....	81
65	Arc pointu dans le cas général où l'angle B_e non zéro.....	82
66	Les types des arcs pointus générés par le programme (prog_pointu).....	83
67	Tableau de neuf types d'arc pointus.....	84
68	La génération des dômes entrelacés avec le remplissage.....	85

Figure	Titre de figures	Page
69	Tableau de neuf types des dômes entrelacés.....	86
70	Le domaine numérique du facteur de la déformation (Fa).....	87
71	Le traceur d'un arc elliptique.....	87
72	Le traceur d'un arc elliptique (deux cas généraux et trois cas particuliers).....	88
73	Les changements formels d'un type d'arcs elliptiques.....	89
74	Les principaux types des dômes entrelacés ellipsoïdaux.....	91
75	Schéma montre le rôle de la bibliothèque numérique dans la modélisation	92
76	L'insertion d'une unité dans un ensemble (contexte architectural).....	93
77	Les éléments du contexte architectural.....	94
78	La dérivation de nouvelles formes de la coupole (contexte architectural).....	95
79	La configuration du remplissage des dômes entrelacés.....	96

List des figures de l'annexe-coupole

Figure	Titre de figures	Page
c.1	La modélisation d'éléments principaux de la coupole.....	ii
c.2	la base de la coupole.....	iii
c.3	Le cadre.....	iv
c.4	La dalle (la base de la coupole).....	v
c.5	La base de la colonne.....	vi
c.6	L'assemblage de la base de la colonne.....	vii
c.7	La quadrique de la construction de la colonne	viii
c.8	Les trois types des colonnes.....	viii
c.9	Quadrique plane de la délimitation de la colonne.....	viii
c.10	Les trois quadrique qui forment l'arcade.....	ix
c.11	Illustration de la construction de l'arcade.....	ix
c.12	Modélisation de l'unité primitive de l'arcade.....	x
c.13	La modélisation de la base du chapiteau.....	xi
c.14	La projection latérale de la base du chapiteau.....	xi
c.15	La partie inférieur du chapiteau.....	xii
c.16	Le plan rotatif qui construire les arcs radiaux.....	xiii
c.17	Les arcs du dôme radial.....	xiii
c.18	Le remplissage du dôme radial.....	xiv
c.19	Le remplissage sphérique des dômes entrelacés.....	xv
c.20	Le remplissage cylindrique des dômes entrelacés.....	xvi

Remerciements

Je remercie Messieurs Claude Parisel et Jean-François Rotgé directeur et co-directeur de cette recherche. Leur ouverture d'esprit, les nombreux conseils et leur confiance ont été vivement appréciés. Leur soutien dans l'élaboration de ce travail m'a permis d'approfondir mes recherches sur la représentation géométrique ainsi que sur la modélisation assistée par ordinateur. De par la discipline et leurs recherches scientifiques se sont trouvées à l'origine de mon intérêt marqué pour la géométrie et l'informatique. Grâce à eux, j'ai découvert l'importance de la forme, la nécessité de la géométrie et la particularité de l'informatique comme outil de représentation.

Je tiens à remercier ici les membres du GRCAO (Groupe de Recherche en Conception Assistée par Ordinateur) et surtout les professeurs Manon Guité et Temy Tidafi, qui m'ont aidé à orienter ma recherche. Leurs suggestions, critiques et conseils furent une source constante d'inspiration qui m'a aidé à concrétiser mon travail et à formaliser les questions de la problématique et de la méthodologie. La discipline proposée par Temy et la très grande disponibilité de Madame Guité m'ont apporté beaucoup en tant qu'expérience professionnelle. Qualités que j'ai très vivement appréciées.

Je n'oublie pas mes amis de la compagnie SGDL-Système: M.Olivier, Laurent Daniel, et Marius Bogdan qui m'ont présenté un support technique inestimable. Leur soutien m'a aidé à résoudre les problèmes de programmation, à examiner les résultats de ma recherche et à organiser mes données dans un cadre formel et scientifique. Par leur soutien, leur gentillesse et leur générosité inoubliables, ils ont contribué à l'élaboration de la partie informatique de ce travail.

Il convient d'associer à ces remerciements toutes les personnes de la Faculté de l'Aménagement et du département d'informatique de l'Université de Montréal qui m'ont encadré, aidé et soutenu à la réalisation de cette recherche. Je pense en particulier au professeur Roger-Bruno Richard avec qui j'effectue un travail sur la modélisation de projets architecturaux. Les recherches sur «les noyaux porteurs» ont ajouté à mon expérience dans le domaine de la modélisation de projets architecturaux.

Enfin, toute ma gratitude va à ma famille qui m'a encouragé à continuer mes études ainsi qu'à mes proches amis qui m'ont aidé à achever ce travail. Je remercie tout particulièrement Pierre Huppé de LaGroÿe et Taïeb Meriane pour la lecture de ce texte et le contrôle du bon usage de la langue.

Que toutes ces personnes sachent combien j'apprécie leur contribution humaine et scientifique, tout en espérant que leur collaboration et leur générosité subsistent.

CHAPITRE - I

**INTRODUCTION, PROBLÉMATIQUE, OBJECTIF ET
MÉTHODE DE TRAVAIL**

CHAPITRE - I

INTRODUCTION, PROBLÉMATIQUE, OBJECTIF ET MÉTHODE DE TRAVAIL

1-Introduction

La construction et la conception sont deux activités dont la première vise à représenter la réunion d'éléments matériels dans un ensemble d'un objet ou d'une forme artificielle suivant un processus et la deuxième l'idée conceptuelle sous-jacente à la forme géométrique de l'objet (Emmerich, 1981, p.19). Le passage de l'idée qui préside à la conception, à sa matérialisation physique, exige une simulation de sa forme; c'est-à-dire sa modélisation. L'outil informatique permet aujourd'hui de construire un modèle dans un environnement virtuel comme étape transitoire vers la matérialisation dont la facture dépend des objectifs de simulation envisagée.

Entre la conception et l'exécution, il y a une étape fondamentale, celle de la géométrisation. Celle-ci définit la forme de l'objet dans l'espace. Cette étape met en évidence l'intention artistique ou constructive de l'objet architectural. Par exemple: une sculpture peut être représentée artistiquement par sa forme générale et par les proportions entre ses principaux éléments, sans définir leur référence géométrique, alors qu'un objet dont la structure est définie par sa géométrie peut être représentée concrètement par des données numériques.

La représentation d'un modèle géométrique doit prendre en considération la mémoire de son origine, c'est-à-dire le concept sous-jacent à la forme et spécifier sa matérialité, c'est-à-dire sa construction et sa résistance physique.

L'objectif de cette recherche est d'explorer les formes de représentation géométrique qui permettent ce passage du concept à la construction. Pour ce faire, nous limiterons le travail à une étude de cas dont la généralisation pourra mener à divers domaines d'application.

Nous avons choisi les dômes comme cas d'étude particulier en raison de leur pertinence pour le problème que nous voulons traiter. Les dômes sont des formes courbes dont la structure géométrique nous ouvre un champ de recherche important dans tous leurs aspects d'ordre structural, architectural, historique et géométrique. Ce champ de recherche est riche et alimenté par de nombreuses études structurales et géométriques.

2-L'arc et la voûte dans l'antiquité

Historiquement, le dôme et principalement l'arc étaient omniprésents dans les anciennes architectures, surtout dans les édifices publics et religieux. L'arc était connu des Égyptiens avant le II^{ème} millénaire av. J.C. Les Grecs l'ont utilisé comme élément décoratif. À l'époque romaine, l'arc était largement employé dans les travaux de génie civil: (aqueducs, viaducs, portes de cités). Il était aussi utilisé comme symbole de triomphe devenant ainsi un élément commémoratif dans les cités romaines.

Les architectes romains ont développé une nouvelle technique pour couvrir de larges surfaces avec une voûte. La forme de cette voûte est le résultat de la translation d'un arc sur son axe longitudinal. Cette idée a conduit à créer la voûte d'arête ou croisée, qui est le résultat de l'intersection de deux voûtes semi-cylindriques ou une voûte berceau (figure-1). La découverte par les Romains d'un nouveau matériau de construction «ciment plus sable volcanique» a permis aux architectes de cette époque de construire de grands dômes, comme celui du Panthéon de Rome, dont le diamètre est de 40 mètres, et de tirer profit à la fois du potentiel esthétique des dômes et de cette nouvelle matière.

Les Byzantins ont résolu le problème de la construction d'un dôme en demi-sphère sur un plan carré en créant un système de dôme à pendentifs (figure-2).

Dans le cadre de l'architecture islamique, les Arabes ont développé le système entrelacé (figure-3). Ils ont construit le premier dôme entrelacé vers le VIII^{ème} siècle. Selon (Adam, Robert, 1984, p.178), les dômes entrelacés sont caractérisés et classés architecturalement parmi les dômes complexes.

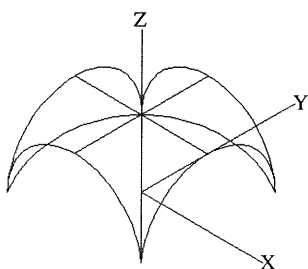


Figure-1. Un dôme d'arête

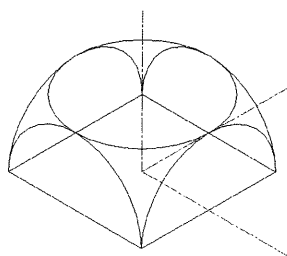


Figure-2. Un dôme pendentif

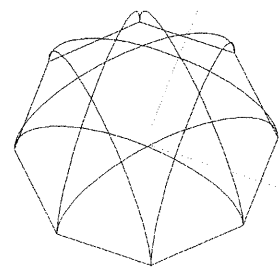


Figure-3. Un dôme entrelacé

En fait, ces éléments, l'arc et le dôme, présentaient des solutions efficaces pour couvrir des surfaces relativement larges. Tout en s'appuyant sur la façon de distribuer des charges aux matériaux de construction utilisés et sur les modes d'assemblage des éléments primitifs. (Besenval, 1984, p.25) d'après Salvadori et Heller «les structures à résistance de forme sont des structures dont la résistance est obtenue en donnant au matériau une forme adaptée aux charges qu'elles doivent supporter ».

Les connaissances de comportement structural d'une forme, c'est-à-dire l'adaptation de la forme géométrique de l'objet aux charges appliquées, sont très développées au début du siècle dernier. Cette période était marquée par des recherches géométriques sur de nouvelles formes obtenues par la combinaison des différents types de courbes gaussiennes lesquelles peuvent être positives, négatives ou nulle (Gaston, Peter, 1983, *Geometry of spatial forms*. p-358). Elle était aussi marquée par des recherches sur les lignes directrices des surfaces courbes générées à partir de ces types de courbes.

3-Description du sujet

Dans toutes les méthodes de représentation (mathématique par des équations, écritures ou descriptions, dessins, maquettes solides, maquettes numériques, etc.), on traduit des données géométriques dans le but de modéliser ou de représenter un objet d'une manière plus réaliste. L'objectif de cette étude est de fournir à la représentation de l'objet, que nous envisageons, les informations nécessaires qui puissent identifier cet objet, le spécifier et ainsi rendre possible sa modélisation dans un environnement virtuel. Trois types d'informations sont requises dans cette opération de représentation de l'objet: des informations géométriques, des informations historiques et des informations structurales. Les informations géométriques permettent d'établir la règle géométrique de la construction de l'objet, de déterminer ses relations mathématiques (numériques) et de définir sa forme spatiale dans l'espace. Les informations historiques s'occupent du développement de l'objet durant son histoire et attestent sa forme par rapport aux autres objets semblables. Enfin, les informations structurales définissent la façon de construire et le mode d'assemblage de ses principaux éléments. Toutes ces données et ces informations font partie d'une problématique globale.

En fait, un objet peut être défini par ses informations et/ou ses données numériques, géométriques, architecturales, historiques et/ou structurales. On analyse l'objet afin d'obtenir ces informations qui, par la suite, pourront être représentées dans un but de modéliser ou de reconstruire cet objet. Puisque l'objet contient un grand nombre d'informations ayant besoin d'être traitées et représentées, il est important de choisir une méthode de représentation efficace, capable de manipuler ces différents types d'informations. Prenons par exemple, la méthode de représentation par maquette solide ou par dessin Cette méthode peut représenter l'objet et/ou n'apporter que certaines informations. Il est difficile, par exemple, de simuler le processus de construction ou d'assemblage d'un objet sur papier ou maquette solide. En fait ces méthodes de représentation manquent de flexibilité et de généralité dans le traitement des données. Elles sont aussi limitées par les outils et les instruments qui sont traditionnellement utilisés dans ce genre de méthodes. Contrairement aux méthodes de représentation sus-mentionnées, les outils informatiques peuvent être de beaucoup plus ouverts et flexibles. Ils permettent d'intégrer toutes les informations de représentation d'un objet dans une structure logique qui serait en mesure de modéliser cet objet et de générer des

nouvelles formes appartenant à une même famille géométrique. Il s'agit, en fait, d'établir une maquette numérique. (De Paoli, Ph, 2000, p.168) dans son introduction sur la modélisation, mentionne qu'une maquette architecturale est un objet qui peut être vu sous tous ses angles et qui représente les aspects de l'objet. Transformer ce modèle (maquette architecturale) pour qu'il soit vu à l'écran d'un ordinateur signifie créer une maquette numérique de l'objet, mieux, en faire une description géométrique qui puisse être lue, grâce à un outil informatique et dont le point de départ est une analyse systématique de l'objet. Le but de la modélisation est de fournir de l'objet toutes les informations qui puissent le représenter d'une manière réaliste. (De Paoli, 2000; p.168) a montré également les étapes successives de la modélisation à travers lesquelles, l'objet passe par plusieurs étapes de représentations avant qu'il devienne un objet modélisé: «le passage de l'objet externe (objet réel ou physique) à l'objet interne (objet modélisé ou virtuel) s'appelle modélisation suivant un schéma proposé par C.Pair et M.Gaudel» (figure-4).

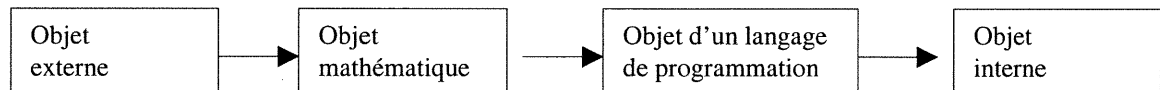


Figure-4. Schéma de modélisation selon C.Pair et M.C.Gaudel. (De Paoli, 2000; P.168)

À partir de cette description, le sujet de recherche porte sur trois points principaux :

- 1- Représenter un type de dôme avec toutes ses caractéristiques s'appuyant sur la compréhension géométrique, structurale et historique de cet objet (figure-5).
- 2- Adapter les outils informatiques disponibles aux besoins de la modélisation en décrivant le processus de la génération au lieu de géométrie résultante.
- 3- Établir un programme permettant de décrire numériquement l'objet et de générer ce dernier dans un environnement virtuel.

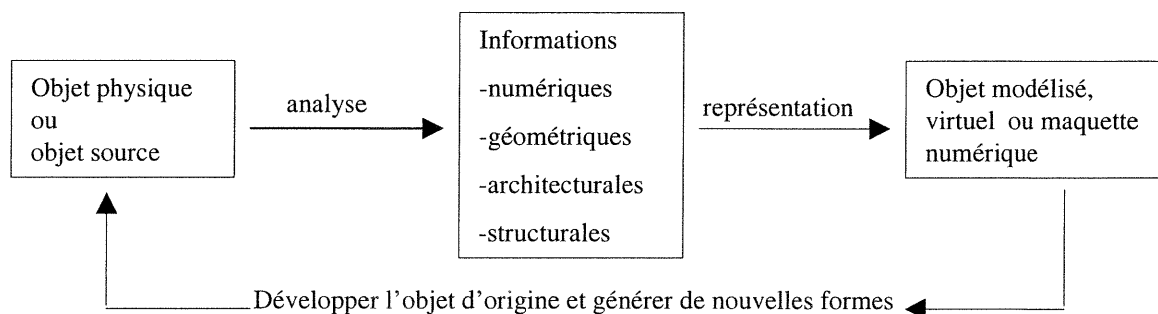


Figure-5. Schéma démontrant la représentation d'un objet par catégories d'informations

4-Définitions et mots-clefs dans le contexte de l'étude

1- Un algorithme:

C'est une structure logique décrivant un processus de traitement permettant d'obtenir une solution à un problème. Il s'agit ici d'un traitement informatique pour résoudre un problème de géométrie.

2- La géométrie:

C'est un ensemble de représentations mathématiques qui définit la forme d'un objet dans l'espace.

3- Un modèle mathématique:

C'est un ensemble de données et de relations mathématiques organisées dans un contexte logique, visant à fournir une solution dimensionnelle et/ou numérique dans un cadre formel et scientifique.

4- Une règle géométrique:

C'est un ensemble de relations contrôlant les rapports entre les paramètres géométriques d'un ou de plusieurs objets.

5- Une règle architecturale:

C'est l'équivalent d'une règle géométrique. Cependant, une règle architecturale apparaît souvent comme un cas particulier d'une règle géométrique qui est générale. Cette limitation définit un élément architectural et le spécifie par rapport à l'ensemble des données et/ou aux autres éléments architecturaux.

6- La génération d'un objet:

C'est l'opération qui vise à la production d'un objet particulier par la mise en œuvre d'une procédure dans un contexte défini.

7- L'assemblage:

La notion d'assemblage est utilisée au sens de regroupement des éléments constructifs, architecturaux et/ou décoratifs d'un objet, selon une règle structurale, géométrique et/ou architecturale.

8- La simulation:

La simulation est une opération d'imitation du comportement d'un objet ou d'un phénomène physique, naturel et surtout géométrique dans un milieu virtuel.

9- La méthode procédurale:

C'est la description d'un procédé unique produisant divers résultats selon les valeurs contextuelles utilisées. L'ensemble des résultats obtenus par un même procédé forme une famille ou une classe d'objets.

5-Problématique:

Dans la description du sujet, nous avons parlé de deux points principaux qui peuvent être au fondement de la problématique. Le premier est la représentation d'un objet ou bien la modélisation d'un objet grâce à l'utilisation des moyens informatiques, le deuxième est la génération de nouvelles formes à partir de cette forme de base.

La génération et la création de nouvelles formes ne sont pas seulement un problème architectural, elles sont aussi un problème industriel et structural. Aujourd'hui, les nouvelles procédures utilisées en architecture passent par l'industrialisation. La problématique majeure est, sans contredit, la production en série laquelle se traduit du point de vue géométrique par la recherche de formes décomposables, identiques et répétitives. Dans le domaine de la construction, les nouvelles formes sont obtenues en fonction de forces appliquées à un objet. Par exemple, grâce au calcul structural et sous l'effet des forces, se produit non pas une seule forme mais bien une succession continue de déformations formelles. Celles-ci sont représentées par l'enregistrement d'une série de formes géométriques. L'ensemble de ces formes et de leur ordre de succession constituent un processus qui a, par lui-même, une forme supérieure, Puisque contenant tous les états spatiaux d'un être existant dans le temps. En fait toute génération ou création se déroule dans une unité espace-temps (Emmerich, 1981). Aussi la recherche de nouvelles formes est-elle un problème général qui dépend de l'objectif envisagé.

Quel que soit l'objectif de la génération: architecturale, structurale et/ou industrielle, la difficulté apparaît non seulement lors de la génération d'une série de formes mais aussi dans l'organisation des multiples données. L'essentiel est de saisir les relations qui existent entre les formes générées qui se regroupent dans un objet et/ou une forme plus complexe. Pour comprendre l'essence d'une forme c'est-à-dire son schéma de construction, il faut pouvoir définir les lois, les règles qui gouvernent la coordination de ses éléments, et enregistrer ainsi leurs positions mutuelles.

En effet, l'objectif de cette recherche n'est pas d'inventer ou de créer de nouvelles formes, mais plutôt d'inventorier les formes possibles qui sont, en fait, les représentations enregistrées des mouvements possibles dans l'espace. (Emmerich, 1981, p.21) « selon Bergson, toute forme a son origine dans le mouvement qui la trace. La forme n'est que mouvement enregistré ». À partir de cet énoncé, on comprend que la génération de nouvelles formes soit contrôlée par un ordre dit communicable. Or, toute communication suppose un code, une convention qui permet de la représenter. Et c'est ce qui permet code permet de communiquer, d'organiser et de contrôler les formes d'un mouvement enregistré.

Dans le cadre de la première partie de la problématique la question qui se pose est la suivante:

Comment peut-on décoder et contrôler l'ordre qui assume la responsabilité de la communication d'une série de formes générées ou enregistrées et découvrir la règle qui organise l'ensemble de ces formes dans un contexte plus complexe?

Évidemment, toute recherche a besoin d'un milieu et d'outils de travail pour sa réalisation. Nous avons déjà mentionné, dans la description du sujet, que nous effectuerons les opérations de modélisation en nous servant de la méthode de représentation par maquette numérique fondée sur les outils informatiques. Or, les moyens et les outils de la modélisation doivent, premièrement, répondre au besoin de génération de nouvelles formes, basée essentiellement sur la liberté de déformation ou de transformation géométrique de l'objet, et ce, dans le cadre d'un ordre ou d'une règle; ce qui implique que la forme géométrique de l'objet à modéliser est paramétrable et indépendants de ses mesures. Deuxièmement, ces outils doivent répondre aux transformations géométriques de l'objet, qui nécessitent plusieurs opérations: translation, rotation et /ou déplacement. Ces opérations se déroulent, d'une manière isolée ou en combinaison dans un environnement dynamique. On comprend, dès lors, que la représentation géométrique de l'objet passe par la modélisation d'action «espace-temps». Ainsi l'opération de modélisation implique des tâches qui rendent possible la construction de l'objet selon des actions organisées dans le cadre d'une structure logique. Comme ces actions ont besoin d'être transférées à l'ordinateur pour traitement. Nous chercherons alors les moyens qui nous permettent de représenter ces actions, tout en respectant leurs structures logiques.

Dans le cadre de la deuxième partie de la problématique se pose la question suivante:

Quels sont les moyens et les outils informatiques qui permettent de manipuler et d'organiser les informations numériques et géométriques qui contrôlent la modélisation d'un objet par ordinateur ?

En résumé, la problématique repose sur deux points principaux: celui de l'ordre géométrique qui assure la communication de l'objet modélisé et celui du système géométrique sur lequel vont s'appuyer les outils informatiques. Il faut par conséquence, établir une règle géométrique, capable de contrôler la forme géométrique de l'objet et de trouver un environnement ou un procédé de génération qui permet un passage continue entre les divers types de surfaces. Pour ce faire on propose comme hypothèses:

1-L'adoption d'un modèle mathématique, géométrique et/ou graphique se base sur les résultats de données analytiques résumant toutes nos connaissances structurales, géométriques et historiques sur l'objet modélisé, de façon que l'on puisse; modéliser le comportement physique (déformations) de cet objet, communiquer sa forme spatiale à partir d'informations et de relations numériques et généraliser l'application du modèle.

2-L'utilisation des descriptions informatiques comme outils de modélisation basées d'une part sur la géométrie projective qui assure une représentation géométrique unique pour l'ensemble d'éléments de l'objet modélisé et d'autre part sur la programmation. Celle-ci permettra de représenter les actions dans le cadre d'un langage informatique ou symbolique^[1] et de les transférer à l'ordinateur pour traitement.

[1]-Un langage symbolique est un moyen d'expression de la pensée mettant en œuvre des symboles. Ce langage doit permettre d'exprimer la façon de réaliser une action et la manière dont des actions ont un effet les unes sur les autres (Tidafi, 1996, p.255).

6- La compréhension géométrique d'une surface

En raison de l'importance particulière de la représentation géométrique de l'objet étudié et des caractéristiques géométriques des descriptions informatiques, la géométrie occupe une large partie de la problématique. La modélisation d'un objet ou/et des processus de construction de cet objet par ordinateur est réalisée au moyen d'un langage symbolique. Ce langage peut comporter un vocabulaire géométrique. Dans ce qui suit, il s'agit d'identifier un langage symbolique et un système géométrique. (Selon Tidafi,T; Ph, p.257) «un système géométrique est un ensemble organisé d'équations mathématiques permettant de définir et de transformer une forme géométrique. Il est généralement articulé autour de deux principaux groupes de concepts mathématiques; les concepts permettant de traduire en équations des primitives géométriques comme une droite ou une sphère et les concepts servant à opérer les transformations sur ces primitives ». Ainsi, la représentation mathématique et procédurale d'une forme et les transformations géométriques de cette forme jouent un rôle majeur dans la modélisation du processus de construction et la génération de nouvelles formes.

6.1- La représentation mathématique d'une surface

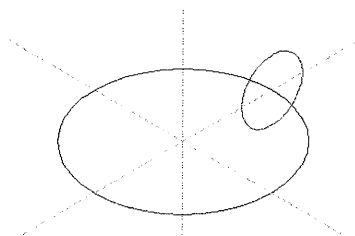
Une surface géométrique peut être représentée mathématiquement par une fonction $F:(x, y, z)=0$ où (x, y, z) sont trois variables qui représentent un point sur la surface et F est la relation numérique qui les lie. Par exemple, l'équation suivante $(x^2/a^2+y^2/b^2+z^2/c^2-1=0)$ est du deuxième degré et représente un ellipsoïde. Les surfaces qui sont représentées par ce type d'équations s'appellent les surfaces quadriques. Par exemple une sphère, un cylindre ou une paraboloid hyperbolique sont représentées mathématiquement par ce type d'équations.

6.2- La représentation procédurale d'une surface

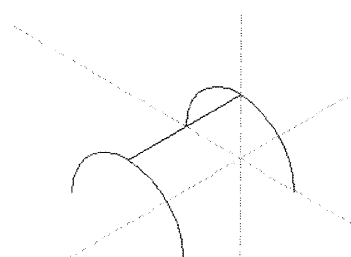
Il s'agit de représenter le processus qui engendre la surface en décrivant les actions qui la définissent. Il y a trois types de représentations:

I- Représentation obtenues par translation:

La translation d'une courbe sur une autre courbe: par exemple le tore est un cercle qui se développe sur un autre cercle (figure-6.a), Une paraboloid hyperbolique est une courbe positive qui se développe sur une autre courbe négative. Dans le cas où une des deux courbes génératrices serait une ligne droite, on la définit comme une courbe dont le centre est à l'infini (selon la géométrie projective). Par exemple, un cylindre peut être obtenu par la translation d'un cercle sur une ligne droite (figure-6.b).



(a). Un tore



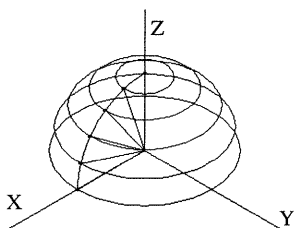
(b). Un demi-cylindre

Figure-6. La génération d'une surface par translation

II- Représentation obtenues par rotation:

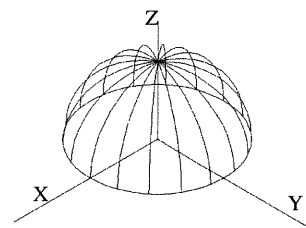
La rotation d'une courbe autour de son axe peut générer une sphère et celle d'une ligne droite autour d'un axe parallèle à cette ligne génère un cylindre. Dans le cas où la ligne et l'axe de rotation ne sont pas parallèles, la forme qui en résulte est un cône. Il faut mentionner que la courbe génératrice et l'axe de rotation se situent dans le même plan.

Sur le plan pratique, la rotation et/ou la translation sont définies en fonction du résultat envisagé. Lorsque l'on pense à un dôme, la première forme qui nous vient à l'esprit est la sphère. Dans l'histoire de l'architecture, bien avant le XV^{ème} siècle, avant même la découverte de la géométrie analytique[1], les constructeurs de dômes concevaient la surface de cet élément du point de vue de la construction, sans en connaître les propriétés mathématiques. Ils décrivaient la forme comme étant le résultat d'un phénomène et non pas comme une réalité mathématique. Par exemple, un dôme romain était géométriquement le résultat du développement d'un cercle de base sur un arc de 90° (figure-7-a). On remarque que le cercle de base subit un changement d'échelle durant son développement et/ou sa translation sur l'arc. Cette définition tient à la manière de construire ce type de dôme. La deuxième méthode de la génération de cette sorte de surface est la rotation d'un arc autour de son axe de 360 degrés (figure-7-b). Les deux méthodes sont différentes tout en utilisant les mêmes primitives géométriques, arc et/ou cercle. On peut dire que les constructeurs de dômes ont défini la manière de construire un objet par le processus de création de sa forme géométrique.



(a-). Développement d'un cercle sur un arc
(translation plus changement d'échelle)

Figure-7. La génération d'une surface par rotation



(b-). Développement d'un arc sur un
cercle (rotation)

III- Représentation obtenue par déplacement

En mathématiques, on démontre que tout déplacement peut se ramener à une combinaison de rotation et de translation. (D'après Besenval, 1984, p.25) «le déplacement est plus général que la rotation et la translation. Il est défini par 1, 2 ou 3 courbes directrices sur lesquelles glisse un élément générateur en des conditions déterminées avec ou sans déformation». Un déplacement peut être réalisé à partir d'une application de plusieurs matrices sur une forme géométrique. Le déplacement obtenu par une combinaison matricielle, est appelé génération d'une surface par processus.

[1]-la géométrie analytique est une branche de géométrie qui combine l'algèbre et les calculs avec la géométrie afin de résoudre deux types de problème: (1)-étudier les propriétés géométriques d'une courbe dont l'équation est connue. (2)-définir l'équation d'une courbe dont les propriétés géométriques sont connues (THOMAS, George, 1968)

Les formes géométriques de la figure-8 représentent un exemple typique de transformation géométrique par déplacement. Ces formes montrent la génération des différentes formes spirales et hélicoïdales. La structure géométrique de ces formes est constituée de groupes de points dont les coordonnées peuvent être modifiées ou relocalisées en utilisant différentes combinaisons des matrices. Celles-ci sont constituées d'éléments variables, constants et/ou d'équations paramétriques comme « $x=a*\cos u$, et $y=b*\sin u$ $z=c*\cos v$ » où x , y et z les coordonnées d'un point, a , b et c des variables et u et v deux angles (Schwartz, *Analytic geometry and calcul*, 1960). La génération surfacique par processus implique plusieurs types de multiplication matricielle. L'exemple ci-dessous montre les opérations principales de déplacements ou de transformations géométriques qui peuvent être utilisées dans cette recherche:

1- Translation et Rotation: (figure-8-a)

Chaque système informatique utilise ces opérations d'une manière différente. Il y a des logiciels DAO de système CSG qui peuvent réaliser deux opérations combinatoires translation et rotation. Par exemple: le logiciel ARC+ peut réaliser des formes hélicoïdales en utilisant la combinaison matricielle «rotation * translation».

2- Rotation et Rotation : (figure-8-b)

Il y a très peu de systèmes qui peuvent réaliser cette opération. Dans la majorité des logiciels CSG, on exécute cette opération d'une manière séquentielle. Les systèmes programmables effectuent ces opérations en utilisant la concaténation matricielle, c'est-à-dire la multiplication des deux matrices de rotation. Pov-Ray et SGDL sont parties intégrantes de ces systèmes.

3- Rotation, Rotation et Translation: (figure-8-c)

La transformation sinusoidale d'une forme sphérique ou ellipsoïdale est le résultat de multiplication de trois matrices «rotation * rotation * translation».

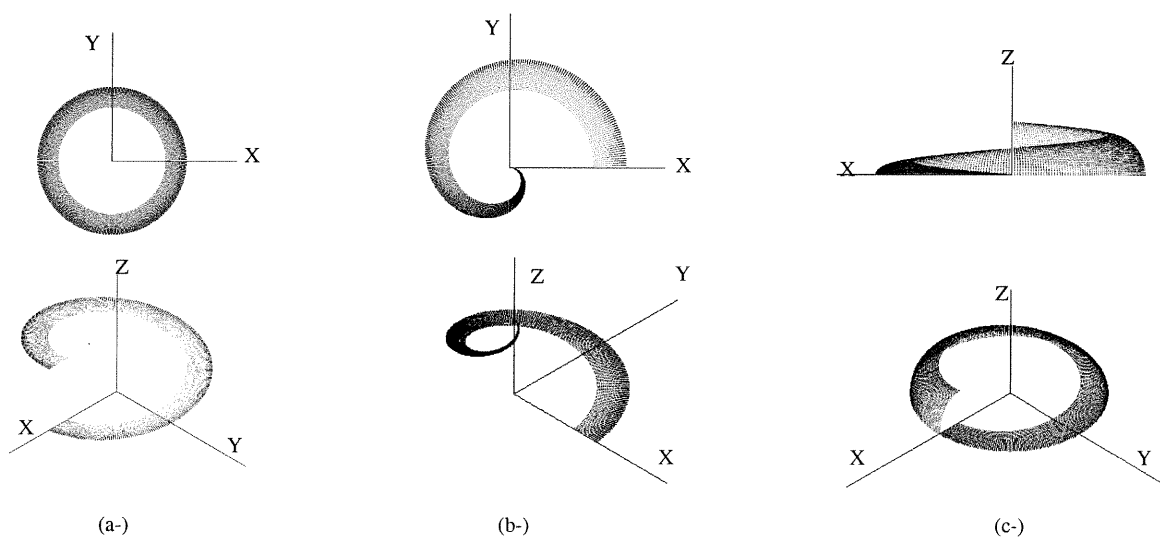


Figure-8. Génération surfacique par processus «Voir l'annexe-lisp fonction lisp-4, Ex-3»

(a-). Translation * rotation «Voir l'annexe-lisp fonction lisp-4, Ex-1»

(b-). Rotation * rotation «Voir l'annexe-lisp fonction lisp-4, Ex-2 »

(c-). Rotation * rotation * translation

6.3-Transformation géométrique d'une surface et/ou d'un type surfacique:

D'une façon intuitive, on peut considérer les transformations géométriques comme des actions qui déforment ou déplacent les objets. On a l'habitude de les classer par rapport aux propriétés géométriques qu'elles conservent généralement. On appelle ces propriétés des invariants.

Types de surfaces et de courbes génératrices:

La classification surfacique montre que le passage d'un type surfacique à un autre demande un changement de courbes génératrices ou de courbes gaussiennes[1]. (Gaston, Peter, 1983, *Geometry of spatial forms*. p-358) a montré qu'il existe trois types de surfaces qui peuvent être générées à partir différentes combinaisons des courbes gaussiennes:

1-Surfaces paraboliques et/ou cylindriques: un cylindre est le résultat du développement d'une courbe positive $C+$ sur une ligne droite L . (figure-9-a).

2-Surfaces ellipsoïdales: un ellipsoïde est le résultat du développement d'une courbe positive $C+$ sur une courbe positive $C+$ (figure-9-b).

3-Surfaces hyperboloïdes: un hyperboloïde est le résultat du développement d'une courbe positive $C+$ sur une courbe négative $C-$ (figure-9-c).

La possibilité de transformer un type surfacique en un autre dépend du niveau de la géométrie utilisée: euclidienne, projective ou affine. En géométrie projective, les trois types surfaciques sont représentés mathématiquement par une équation du deuxième degré (une quadrique). En fait, les trois types surfaciques sont représentés par une seule surface quadrique. Cette possibilité permet d'unifier la représentation géométrique de primitives et de réduire le nombre et la variété de primitives géométriques utilisées dans la construction d'un objet.

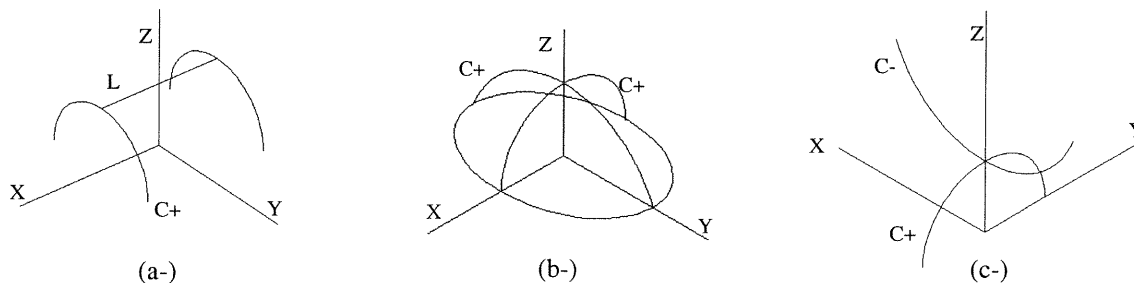


Figure-9. Les trois types de courbes gaussiennes et les types de surfaces

(a-). une courbe $C+$ développée sur une ligne droite L engendre une surface cylindrique ou parabolique. Le demi-cylindre est un cas particulier d'un parabolôïde.

(b-). une courbe $C+$ développée sur une courbe $C+$ engendre une surface ellipsoïdale. La sphère est un cas particulier d'un ellipsoïde.

(c-). une courbe $C+$ développée sur une courbe $C-$ engendre une surface hyperboloïdale. Le cylindre est un cas particulier d'un parabolôïde.

[1]-Courbes gaussiennes: selon (Gaston, 1983) il y a trois types de courbes gaussiennes. Une courbe positive, une courbe négative et une courbe nulle dont le centre est à l'infini..

La transformation géométrique d'une surface est contrôlée par certaines régularités mathématiques. Comme on l'a déjà vu, les trois types de surfaces qui sont engendrées par des courbes gaussiennes peuvent être représentés mathématiquement par une équation du deuxième degré. C'est-à-dire que le passage d'un type de surface à un autre est possible à condition que le niveau de la géométrie utilisée dans cette représentation le permette.

Contrairement à la géométrie euclidienne qui repose essentiellement sur les notions de distance, d'échelle et de proportion des formes morphologiquement identiques, la géométrie topologique est une géométrie de lieu et de position, permettant une transformation élastique entre différents types de surfaces. Une transformation topologique appliquée à une surface élastique revient généralement à déformer celle-ci sans jamais la perforer ou la déchirer; il est possible, par exemple, de déformer un cube élastique pour en faire une sphère. Les deux formes qui dérivent de cette transformation sont «topologiquement» équivalentes.

En géométrie projective, les transformations géométriques sont plus restreintes. En fait, les transformations projectives se produisent lors de la projection d'une forme bidimensionnelle sur un autre plan non parallèle au plan de la forme projetée. Selon Fleury, 1986 «en transformation projective, la figure est passablement déformée, mais plusieurs propriétés géométriques sont préservées. Du fait, dans cette transformation les droites restent des droites et tous les invariants des transformations topologiques sont conservés».

En résumé, la généralité de transformations géométriques d'une forme dépend de la géométrie adoptée et du nombre des invariants imposés par cette géométrie. Au fur et à mesure que le nombre des invariants diminue, les transformations géométriques deviennent plus élastiques et la géométrie, plus flexible. En effet, dès que l'on introduit des invariants, la correspondance entre les formes devient plus stricte et la géométrie plus rigide. Partant de la géométrie topologique, exigeant de plus en plus de propriétés invariantes, on passe par divers types de géométries: projective, affine et euclidienne.

L'adoption des transformations projectives dans les processus de génération de nouvelles formes nous permet d'abandonner provisoirement les données quantitatives et d'attacher plus d'importance aux données qualitatives de l'objet: ses articulations, l'assemblage des ses parties et le rapport entre leur nombre, bref, à sa composition structurale. Il faut mentionner que la génération de nouvelles formes par ce genre de transformation est un processus ou une mise en rapport originale d'éléments connus pour en faire un nouvel objet. Cet objet ne sera pas la somme de ces éléments, mais un objet tout à fait différente, de part ses ingrédients. Ainsi, une transformation géométrique peut effectuer un objet avec ou sans déformation, c'est-à-dire consister en un déplacement simple de l'objet ou en un déplacement avec déformation. L'importance dans notre recherche, est d'opérer transformation géométrique ou une déformation de l'objet, sans en changer ses propriétés topologiques. À partir de cette

explication, les déplacements, les déformations et/ou les transformations géométriques peuvent se produire entre les trois types de surfaces: d'une surface cylindrique à une surface ellipsoïdale ou à une surface hyperboloïde. Elles peuvent aussi se produire dans le même type de surface, par la déformation d'une surface ellipsoïdale. L'aplatissement d'une sphère en ellipsoïdale ou en tore est un exemple typique démontrant cette déformation. Une surface ellipsoïdale peut être générée par deux courbes gaussiennes positives se développant l'une sur l'autre. Cependant, la sphère et le tore sont des cas particuliers d'un ellipsoïde (figure-10).

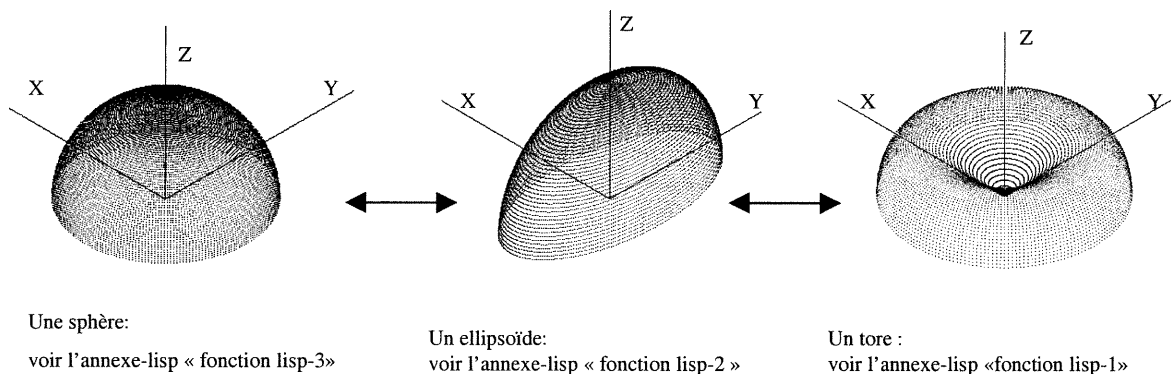


Figure-10. La transformation géométrique d'un type de surfaces

En fait, les objets ou les éléments utilisés dans l'architecture ou la construction, ne sont pas souvent de nouvelles créations. Il sont des éléments connus ou inconnus dont les formes géométriques peuvent être découvertes lors de l'enregistrement de la transformation géométrique de leurs formes. L'architecture, la géométrie et le calcul structural collaborent, dans ce cas, afin de créer de nouvelles formes. Un des exemples les plus classiques de déformations dans le domaine de la construction est celui de la dalle plane qui se transforme en hyperboloïde ou ellipsoïde positif ou négatif (figure-11-a), ou celui de la poutre qui, sous une charge se transforme en arc parabolique (figure-11-b). Cette déformation nécessite d'assurer une continuité géométrique de surface et un contrôle de la forme. La simulation d'une telle transformation demande un système permettant la continuité entre une ligne droite et une parabole, c'est-à-dire un système pouvant gérer un point à l'infini.

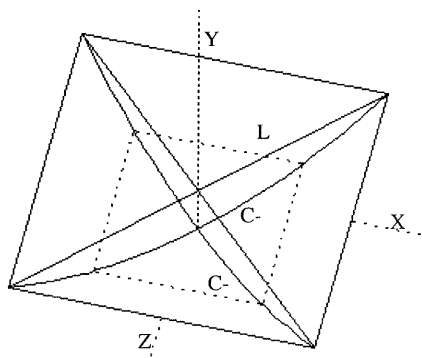
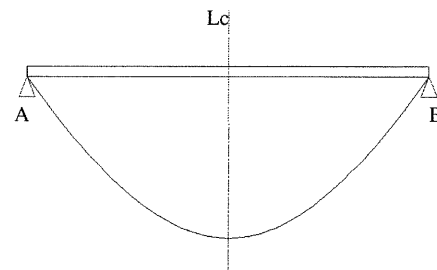


Figure-11. La déformation continue d'une surface
(a). La déformation d'une dalle plane



(b). Une poutre transformée en forme parabolique sous une charge

7-Objectif et méthode de modélisation

7.1-Objectif de la modélisation

Notre principale préoccupation est l'utilisation des moyens informatiques afin d'explorer les formes de représentations géométriques permettant le passage d'un objet réel physique à un objet virtuel. Cela veut dire que la modélisation d'une forme géométrique par ordinateur nous conduit à créer un modèle virtuel ou une maquette numérique. Nous avons déjà dit qu'un objet physique peut passer par différentes étapes de représentations avant d'être transformé en un modèle virtuel. En d'autres termes, une représentation est une étape dans une opération plus large qu'on appelle la modélisation. C'est pourquoi, il est nécessaire de décrire chacune de ces étapes de représentations de l'objet: l'objet réel ou physique, la représentation mathématique de l'objet, la représentation informatique de l'objet et enfin le modèle ou l'objet virtuel. Tout cela est entrepris pour parvenir à une représentation réaliste de l'objet à modéliser. En effet, le but est d'organiser toutes les étapes de la modélisation ou de représentations d'un objet dans le cadre d'une structure logique qui peut être traduisible informatiquement par un langage symbolique. Cette structure pourra instruire l'ordinateur pour générer un modèle virtuel, exploitant une maquette numérique afin d'obtenir de nouvelles formes appartenant à la même famille géométrique.

L'objectif fixé pour élaborer un programme, (un algorithme ou une structure algorithmique^[1]), s'appuie sur un modèle mathématique numérique^[2] inspiré de la compréhension géométrique de l'objet modélisé. Ce programme permettra alors:

1-de générer un objet ou un modèle virtuel^[3] guidé par une règle d'assemblage et par la définition de ses éléments constructifs, tout en respectant les interrelations géométriques et constructives.

2-de simuler les changements formels de la forme géométrique de l'objet en inventoriant les formes possibles qui peuvent être exportées ou insérées dans d'autres environnements architecturaux [4].

[1]- Un algorithme ou une structure algorithmique est une structure logique décrivant un processus de traitement permettant d'obtenir une solution à un problème. Il s'agit ici d'un traitement informatique pour résoudre un problème géométrique.

[2]- Un modèle mathématique numérique est un ensemble de données et de relations mathématiques organisées dans un contexte logique visant à fournir une solution dimensionnelle et/ou numérique dans un cadre formel et scientifique.

[3]- Un modèle virtuel en informatique est une forme géométrique élaborée par ordinateur ayant comme but la représentation d'un objet physique d'une manière réaliste.

[4]- Un environnement architectural est un ensemble d'éléments architecturaux variés ou identiques formant un objet cohérent, architecturalement et géométriquement.

7.2-Méthode de modélisation

Pour atteindre les objectifs de la recherche qui viennent d'être présentés, c'est-à-dire élaborer un programme, établir un modèle virtuel et générer de nouvelles formes, nous suivrons une stratégie de travail basée sur l'organisation de la recherche divisée en trois parties:

La première partie constitue une recherche qui porte sur l'analyse de l'objet à modéliser. Cette partie analytique est essentielles pour élaborer un champ d'informations représentant les principales sources qui alimentent l'opération de modélisation. L'adoption de la méthode analytique permet de disjoindre les unités élémentaires de l'objet, de connaître les relations entre les éléments composant l'objet et enfin de regrouper ces éléments selon la règle d'assemblage de cet objet. Selon (Temy, Tidafi, 1996, p.252) «une approche analytique consiste donc à décomposer un phénomène perçu en éléments simples et indépendants et à les rassembler ensuite. En d'autres termes, il s'agit de transformer des unités en entités, par définition détachées du phénomène à modéliser, et de reconnaître des liens qui existent». En effet la méthode analytique peut convenir à la modélisation d'un phénomène dont les unités et leurs liens sont déterminables et demeurent stables. Dans le cadre de la partie analytique, nous organisons en trois groupes principaux, les informations provenant de l'analyse de l'objet.

1-L'analyse historique visant le développement de l'histoire de l'objet et les effets de ce développement sur sa forme géométrique et sur la manière de construire et d'assembler ses éléments constructifs.

2-L'analyse constructive qui cherche à décoder les règles d'assemblage ou de construction de l'objet à modéliser et à déterminer les relations et les données numériques contrôlant cette règle. Celle-ci sera représentée par des équations et des relations mathématiques.

3-L'analyse géométrique qui permettra d'élaborer un système géométrique capable de préciser les données numériques, géométriques et les relations mathématiques obtenues au cours de cette analyse, et ce, dans le but d'organiser ces données dans le cadre d'une structure géométrique globale.

L'ensemble des informations, issues de ces trois groupes d'analyse est impliqué dans une étude contextuelle qui déterminera les relations entre les différents types de paramètres et de variables en vue d'établir un modèle numérique. Celui-ci représentera la base mathématique du programme futur.

La deuxième partie vise le choix d'instruments de modélisation. Les outils de modélisation devraient répondre à l'objectif précité, qui est de créer un modèle virtuel ou une maquette numérique de l'objet et de générer de nouvelles formes appartenant à la même famille géométrique et d'insérer ces dernières formes dans d'autres contextes architecturaux. Dans cette partie, nous établirons les critères et les motivations sur lesquels s'appuiera le choix des systèmes ou des descriptions informatiques utilisées en modélisation. Ces critères seront basés sur les capacités du système informatique de traitement des données numériques, géométriques et/ou graphiques, ce système ayant pour but d'organiser ces données et de les transférer à l'ordinateur, grâce à un langage informatique. En fait, les descriptions informatiques utilisées en modélisation seront

identifiées à partir d'un langage de programmation et d'un système géométrique. Le langage de programmation permet de traduire le modèle numérique, qui est déjà préparé dans le cadre de l'étude analytique, en un modèle informatique ou un programme. Le système géométrique, quant à lui, permettra de définir une forme géométrique représentée par une équation mathématique et d'opérer les transformations géométriques réalisées sur cette forme. Les résultats de ces opérations, c'est-à-dire la codification des données, la définition géométrique de l'objet et les transformations géométriques qui effectuent cet objet, seront évalués et validés dans l'expérimentation.

La troisième partie représente l'ensemble des expérimentations destinées de valider les résultats de la recherche. (Tidafi, 1996, p.262) selon J.P. Deslauriers, la validité des résultats d'une recherche signifie que «la méthode de recherche utilisée a été capable de répondre à la question de recherche posée». Il s'agit, effectivement, d'examiner les résultats de la recherche basés, d'une part, sur la création d'une maquette numérique pouvant générer de nouvelles formes et, d'autre part, sur la possibilité d'exportation et/ou d'insertion de ces formes dans un contexte architectural plus complexe. Ainsi le programme ou la structure algorithmique, établie en deuxième partie, sera soumise à un test de fonctionnement qui sera exécuté en deux étapes:

Première étape: tester le programme sur des paramètres permettant de générer l'objet de base représenté par un modèle virtuel ou par une maquette numérique. Cette dernière devrait correspondre à la forme géométrique de l'objet original ou l'objet physique. Ce test pourra répondre à notre premier objectif qui est la modélisation d'un objet ou l'établissement d'une maquette numérique. Dans le cadre de ce test, on essaiera d'inventorier toutes les formes possibles pouvant être enregistrées ou générées en ne changeant que les paramètres. Ce qui nous permettra aussi de caractériser les principales formes correspondant aux points d'inflexion d'une série de valeurs données à un paramètre.

Deuxième étape: s'assurer de la validité externe du programme qui d'une part, sur la génération de deux cas particuliers de l'objet modélisé et, d'autre part, sur l'insertion dans un contexte architectural, d'une forme générée dans les deux étapes précédentes. Ce dernier implique la modélisation de ses principaux éléments. Pour cela nous choisirons un exemple d'architecture parmi les éléments: soit le dôme entrelacé. Cet exemple peut être modélisé et utilisé, par la suite, comme champ d'application du programme.

En résumé de la méthode de modélisation: la stratégie de travail débute par le choix d'un cas d'étude et/ou d'un objet, sujet de la modélisation: un type de dôme. Une étude analytique sera réalisée sur l'objet en vue d'établir un champ d'informations pouvant être utilisé comme base théorique dans la modélisation de l'objet. Par la suite, nous choisirons les outils informatiques nous permettant de traduire ces informations en programme. Celui-ci pourra passer plusieurs tests et expérimentations dans le but d'assurer la validité du programme en lui-même ainsi que dans d'autres domaines d'application.

Conclusion du chapitre-I

La représentation est un ensemble de techniques permettant de formaliser certaines informations concernant un objet et de reconstruire ce dernier afin qu'il soit perçu de différents points de vue. Un mathématicien, par exemple, voit une sphère à partir de son équation et un géomètre à partir de sa forme spatiale. Il est évident que chaque méthode de représentation vise à représenter certains types d'informations. En architecture, la représentation signifie la création d'un objet ou d'un modèle virtuel devant avoir la valeur d'un objet réel. Comme on l'a vu, l'ensemble des étapes qui permettent de représenter un objet d'une manière réaliste dans un environnement virtuel s'appelle la modélisation. Effectivement, un modèle virtuel est une maquette architecturale ou solide pouvant être traduisible à l'aide d'un langage informatique en maquette numérique.

Chaque méthode de représentation exige certains types d'informations: historiques, structurales et/ou géométriques. Ces informations sont analysées et organisées dans le cadre d'un modèle numérique pouvant être traduisible par un langage informatique en programme. L'élaboration de ce dernier nous permet d'établir une maquette numérique de l'objet modélisé, de générer de nouvelles formes à partir d'une forme de base et enfin d'insérer les formes générées dans d'autres contextes architecturaux. Dans la cas de la représentation d'un objet par une maquette numérique ou par des moyens informatiques, les données géométriques sont essentielles pour donner à l'objet à modéliser sa forme spatiale, sa règle géométrique et/ou sa règle d'assemblage. En effet, la géométrie est considérée comme un moyen essentiel pour communiquer l'objet et le représenter visuellement par un modèle virtuel. Celui-ci est ordonné par un programme permettant de contrôler la visualisation et la déformation de sa forme géométrique.

On a déjà mentionné que le but de cette recherche n'est pas de créer de nouvelles formes mais plutôt d'inventorier les formes possibles pouvant être générées et/ou enregistrées en modifiant les paramètres du programme. Par conséquent, cette recherche repose sur l'étude d'un cas réel pouvant être analysé et transféré à l'ordinateur en vue d'obtenir une représentation informatique réaliste d'un objet. Les résultats seront examinés et validés dans le cadre du programme en lui-même et dans d'autres domaines d'application.

CHAPITRE - II

CHOIX DU CAS TYPE

ANALYSE HISTORIQUE, CONSTRUCTIVE ET GÉOMÉTRIQUE

CHAPITRE - II

CHOIX DU CAS TYPE

ANALYSE HISTORIQUE, GÉOMÉTRIQUE ET CONSTRUCTIVE

1-Le choix du cas type

Dans l'histoire de l'architecture classique, la structure des dômes a pris différentes formes. Elle a été influencée par les techniques de construction ainsi que les formes des surfaces couvertes par les dômes. Parmi la grande variété de dômes qui ont été construits durant l'histoire de l'architecture, certains ont une très grande importance architecturale. Par exemple, les dômes pendentifs, les dômes d'arête romains et d'autres, sont considérés comme des oeuvres révolutionnaires; il en est de même des dômes entrelacés. Les dômes romains ont été essentiellement développés à partir d'une forme simple demi-cylindrique ou demi-sphérique. Pour construire un dôme, on utilise une seule forme géométrique en demi-sphère. Mais lorsqu'il s'agit de construire une voûte d'arête, on est obligé d'utiliser une seule forme en demi-cylindre deux demi-cylindre. Cela veut dire qu'un dôme en demi-sphère est équivalent à une primitive demi-sphérique. Les Byzantins n'étaient pas loin de cette technique. Ils ont gardé la même forme demi-sphérique de dôme et inventé le système pendentif qui permet de reposer cette demi-sphère sur un plan carré. Ces deux types de dômes, romain et byzantin, furent suivis de la création du système entrelacé. Celui-ci utilise une primitive simple comme l'arc afin de construire des dômes complexes. Contrairement aux dômes romains et byzantins, les dômes entrelacés utilisent une forme géométrique et une règle d'assemblage plus complexes et variées, enrichissant ainsi la construction des voûtes par de nouvelles formes géométriques des dômes entrelacés.

Ces trois types de dômes sont considérés comme une référence architecturale, structurale et géométrique pour la majorité des dômes que l'on voit actuellement dans des édifices historiques et modernes. En raison de l'importance historique et de la domination des dômes romains, byzantins et entrelacés dans l'architecture des monuments, nous limiterons notre étude à un type particulier de dôme. Notre choix est fondé sur certains critères qui nous poussent de privilégier ce de dômes et à l'adopter comme objet de recherche.

Trois critères peuvent être considérés comme étant des facteurs essentiels dans le choix du cas type de dôme: la géométrie de dôme, la manière d'assembler ses éléments constructifs et enfin sa valeur architecturale et historique.

1-Le premier critère est celui de la géométrie: les trois types de dômes déjà mentionnés (romains, byzantins et entrelacés) sont construits d'après des règles géométriques différentes. Les dômes romains et byzantins s'appuient sur l'esthétique de la forme géométrique (la demi-sphère et/ou le demi-cylindre) utilisée dans leur construction. Les dômes entrelacés, au contraire, reposent sur un système géométrique plus développé que les deux autres types de dômes. La forme de dômes entrelacés tire son importance du système géométrique entrelacé et de la règle d'assemblage de ce type. Cette caractéristique permet de jouer sur le nombre, la forme et la position des arcs formant la structure spatiale des dômes entrelacés. En d'autres termes, le système entrelacé contient de nombreuses variables, s'adaptant à une discipline procédurale particulière.

2-Le deuxième critère est celui du système structural: le système de construction des dômes romains s'appuie sur la manière de construire et de tailler les entités de construction (pierres taillées ou briques), alors que le système entrelacé est basé sur une règle d'assemblage qui en fait, s'adapte à la manière de la construction moderne, c'est-à-dire l'industrialisation de l'architecture. On comprend pourquoi le système entrelacé permet de construire des dômes complexes, au sens architectural et structural, à partir d'éléments constructifs ou primitifs simples, c'est-à-dire les arcs.

3-Le troisième critère est celui de la valeur architecturale: contrairement aux dômes romains et byzantins utilisés comme éléments architecturaux externes, les dômes entrelacés sont considérés comme élément décoratif (dôme intérieur) et comme élément architectural structural (dôme extérieur). En fait, en architecture la construction d'un dôme est coûteuse car souvent ce dernier est un élément unique et/ou principal dans l'édifice. C'est pour cette raison qu'il faut tirer profit du potentiel, à la fois esthétique, architectural et structural, de la construction d'un dôme.

Nous fondant sur ces critères, nous privilégions le système de construction de dômes entrelacés et on l'adopterons comme cas particulier d'étude dans le cadre de cette recherche.

2-Analyse globale des dômes entrelacés

Nous avons déjà dit que l'objectif de cette recherche est de modéliser un objet, c'est-à-dire d'établir une maquette numérique, de générer de nouvelles formes et d'utiliser les formes générées dans d'autres applications architecturales. Or, pour réaliser cet objectif il nous faut analyser l'objet, c'est-à-dire les dômes entrelacés. En effet, nous analysons non seulement les dômes entrelacés, en tant qu'objets physiques, mais aussi leurs systèmes géométriques, structuraux et même leurs références historiques. Les informations obtenues lors de cette analyse, seront organisées et précisées dans le cadre d'une structure logique. Cela nous permettra de définir et de manipuler ces informations d'une manière formelle. En résumé, le système des dômes entrelacés sera soumis à trois formes d'analyses: historique, structurale et géométrique.

2.1- Analyse historique

L'analyse historique vise à retracer le développement des différents types de dômes représentant des points de repère importants dans l'architecture des dômes. Le point essentiel de cette analyse portera sur la continuité de l'expérience de la construction de chacun de ces types à même le développement de nouvelles expériences ou de nouveaux types de dômes.

2.1.A- Histoire du développement de la voûte

Le dôme est la configuration la plus sophistiquée des formes architecturales. La majorité des monuments importants sont couverts par des dômes ou par des éléments y ressemblant. L'utilisation des dômes et des arcs en architecture dès la préhistorique. Au cours de cette période, la construction de ces éléments prenait son origine dans l'observation de la nature. La compréhension géométrique de l'arc n'a commencé qu'après le IIème millénaire av. J.C. À partir de cette époque, l'homme a commencé à comprendre cet élément de manière plus abstraite. La relation entre la forme géométrique et le comportement physique d'un objet devient plus claire. Les outils et les matériaux y sont plus développés. Les nouveaux besoins de l'homme (par exemple: construire des routes, des aqueducs et des viaducs et couvrir de larges surfaces) ont commandé le développement de ce genre d'éléments structuraux, c'est-à-dire l'arc et la voûte.

L'arc de *l'Asaseef* est le premier arc connu dans l'ancienne Égypte. Sa construction date du IIème millénaire av. J.C. Le développement de l'arc à trois dimensions, destiné à la construction d'un berceau a commencé vers le XIVème siècle av. J.C. Les plus anciennes voûtes clavées connues du VIIème siècle av.J.C, sont celles des deux chapelles d'Amon en Égypte. Les Assyriens utilisaient l'arc pointu vers le VII'ème siècle av. J.C, mais ils le construisaient sans utiliser de coffrage. Selon leur méthode de construction, les arcs qui forment la voûte sont inclinés par rapport au plan de la construction verticale (figure-12).

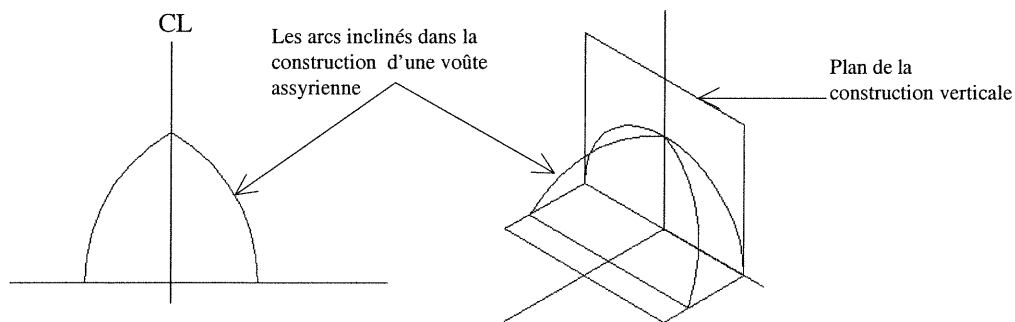


Figure-12. Arc assyrien construit d'une façon inclinée afin d'éviter l'utilisation de coffrage.

Les Perses, vers le Vème siècle avant J.C, construisaient leurs voûtes en utilisant la méthode assyrienne avec ou sans coffrage. En ce qui concerne le dôme, leur géométrie était plus proche d'un ovale; ce qui facilitait le processus de construction et réduisait les forces des poussées horizontales (figure-13). La ligne de transmission des charges du dôme est représentée par une courbe ovale. Chaque point de cette ligne peut se transformer en une poussée horizontale et une force verticale. C'est la poussée horizontale qui pose le problème de la stabilité de la voûte. C'est pour cette raison que les constructeurs ont essayé d'approcher l'angle situé entre la ligne de la transmission des forces et la ligne de support, d'angle droit (90 degrés). À cette époque le dôme était construit de sorte que les rangs de briques ou de pierres reposaient de manière horizontale en assurant une surface de plus en plus large entre les rangs superposés. Cette façon de construire la voûte s'appelle l'encorbellement (figure-14). Selon (Besenval, 1984, p.39) «la technique de l'encorbellement consiste en un surplomb successif d'une assise sur l'autre. Ce surplomb est rendu possible par l'horizontalité des assises, ce qui a théoriquement pour conséquence de ne transmettre que des charges verticales aux supports. Cette caractéristique lui a valu très souvent l'appellation de fausse-voûte».

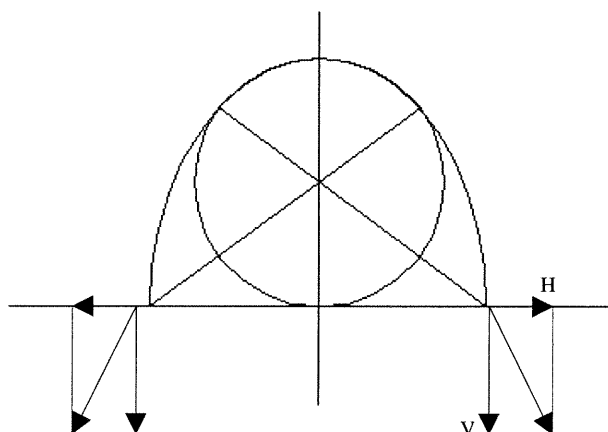


Figure-13. Voûte ovale montrant les forces horizontales H et les forces verticales V.

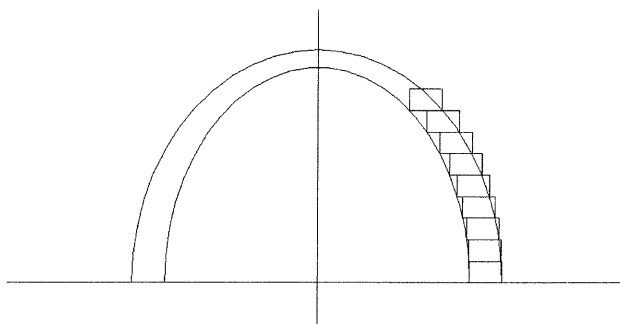


Figure-14. La méthode assyrienne de la construction de voûte ou ce qu'on appelle encorbellement, s'adapte mieux à la forme parabolique ou ovale d'un arc ou d'une voûte.

L'arc et la voûte étaient connus des Grecs, mais ces éléments n'étaient pas utilisés comme éléments essentiels dans leur architecture. Ces éléments servaient plutôt d'éléments décoratifs. Par contre, les Mycéniens ou les anciens Grecs ont construit le dôme pointu souterrain au XIII^{ème} siècle av. J.C pour couvrir la tombe de Tholos. La forme géométrique de ce dôme s'adapte à la méthode de construction dite d'encorbellement et à la manière de distribution des forces (figure-15).

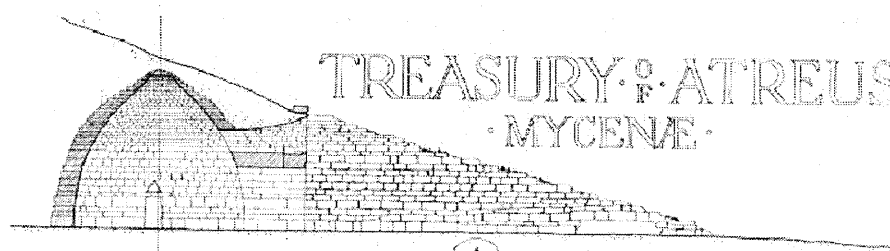


Figure-15. Un dôme souterrain construit par les Mycéniens au XIII^{ème} siècle av.J.C (Mitchell Beazley, 1988, page-135)

Cette période historique de la « Haute Antiquité » et ce, jusqu'à l'époque romaine était marquée par une utilisation rare de l'arc et de la voûte. Le dôme n'a pas joué un rôle important dans l'architecture égyptienne, mésopotamienne et grecque. Il a rarement été utilisé à l'intérieur de leurs architectures sacrées. Il n'apparaît jamais de manière visible dans l'aspect extérieur de leurs édifices.

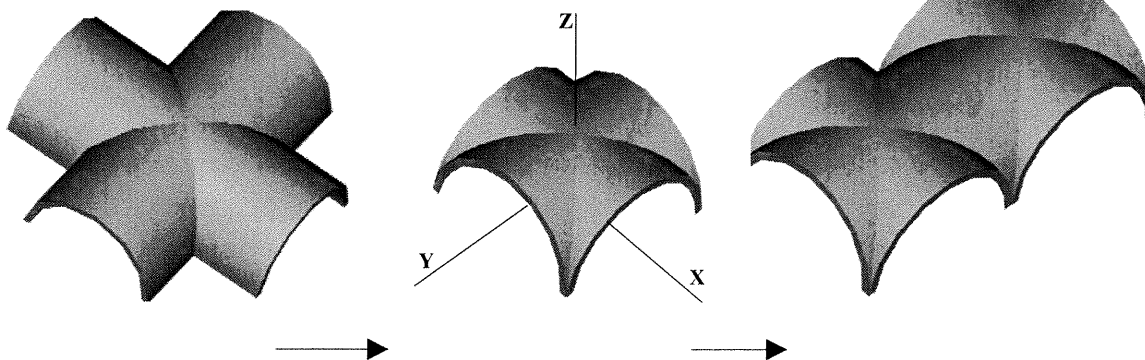
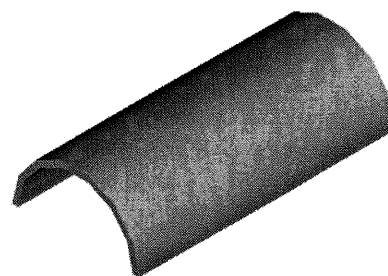
Avec les Romains, la voûte se généralise, devient une technique de construction noble. Elle n'est plus confinée aux constructions souterraines, elle n'est plus un trou dans la masse d'un mur. Les Romains ont dégagé l'espace de la voûte afin qu'elle soit visible et plus signifiante sur le plan architectural et constructif (Sakarouvitchev, 1998). En fait, cette révolution architecturale, qui a donné à l'arc et à la voûte leur valeur, débuta vers le II^{ème} siècle av.J.C La technologie, les nouveaux matériaux de construction et les progrès de la géométrie ont permis aux Romains de construire et de développer le berceau (la voûte en demi-cylindre) et la voûte d'arête ainsi que d'autres formes

et configuration d'arcs et de dômes (figure-16). Les voûtes d'arêtes répétitives étaient une révolution architecturale à cette époque. Il était possible d'organiser ou de claustrer l'espace par des voûtes d'arêtes identiques en profitant du système de construction par coffrage. En effet, la complexité de la taille des pierres, qui limitait la construction des voûtes n'existe plus grâce à l'invention du ciment. Plusieurs configurations de voûtes croisées ont été construites pour couvrir de larges surfaces comme dans les Thermes de Caracalla et le Tepidarium à Rome. D'autres sont couverts par des dômes en demi-sphères comme le Pantoum dont la structure du dôme est basée sur l'utilisation des arcs radiaux ou méridiens et des cercles horizontaux délimitant la structure du dôme aux niches.

Dans Haute Antiquité, la voûte était souvent utilisée comme élément constructif, à la place de poutres. Par contre, les ingénieurs ont commencé à comprendre le comportement structural de cet élément et ils l'ont utilisé comme un éléments porteur et structural dans leurs constructions. De par l'économie des matériaux, permettant de ce fait la légèreté structurale, une pure beauté s'en dégage .

On peut dire que l'expérience romaine dans la construction des voûtes, a contribué, d'une certaine manière, au développement de nouveaux systèmes de construction de dômes tels que le système pendentif et le système entrelacé.

(a-). Un berceau ou une voûte cylindrique: le berceau est la voûte à partir de laquelle les Romains ont développé d'autres types de dômes.



(b-). Percer un berceau par un autre berceau de la même hauteur

(c-).L'intersection de deux berceaux est une voûte d'arête

(d-). Voûtes d'arêtes répétitives

Figure-16. Le développement du berceau chez les Romains

Les Byzantins sont partis de l'expérience romaine pour créer un nouveau système de construction de dômes. En 420 après.J.C, les Byzantins ont construit le premier dôme sur pendentifs (figure-17). Le nouveau système byzantin vise à couvrir une surface carrée par un dôme demi-sphère. Les Romains, avant les Byzantins, ont construit un dôme d'arête sur un plan carré à intersection de deux semi-cylindres (figure-18).

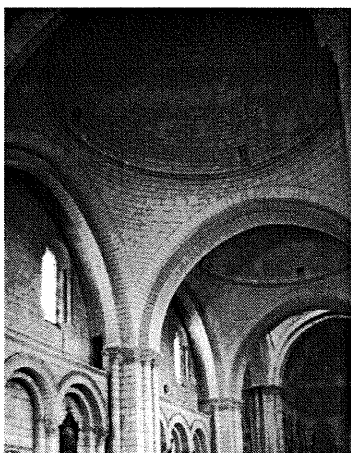


Figure-17. un dôme pendentif, selon (Saudin, 1989)

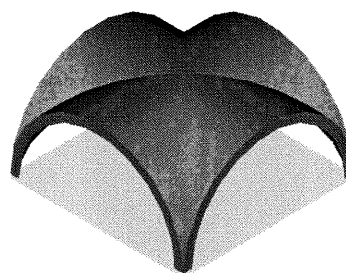


Figure-18. Dôme d'arête couvre un plan carré.

La méthode la plus facile de couvrir une surface carrée par un dôme demi-sphère est, selon (Acland. J, 1972, p.36), de transformer en carré la forme circulaire de la projection de dôme. Par conséquent, les pendentifs byzantins peuvent être le résultat de l'intersection d'une demi-sphère et d'un demi-cube (figure-19-a). Selon (Besenval, 1984, p.63) théoriquement cette forme peut être définie géométriquement de la façon suivante: une surface d'une demi-sphère inscrite dans un demi-cube et ayant:

- pour centre, le centre du carré de base du cube.
- pour rayon la demi- diagonale du carré de base.

La résultat de cette description géométrique est un ensemble de quatre triangles sphériques permettant de construire un dôme en demi-sphère sur un plan carré. L'ensemble de pendentifs ou de triangles sphériques représente la base sur laquelle le dôme demi-sphère s'installe (figure-19-b).

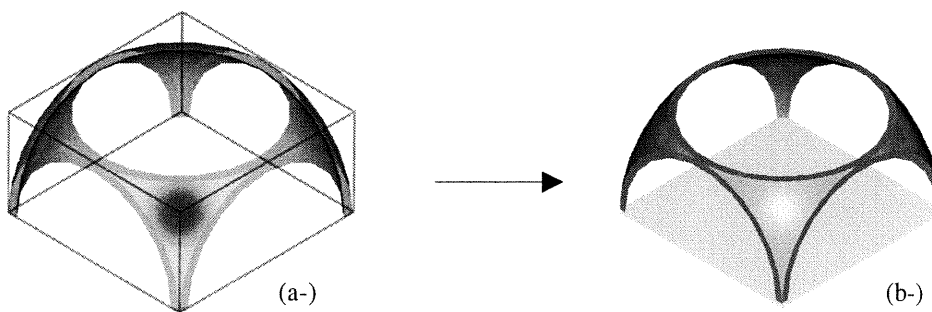


Figure-19. La construction d'un pendentif
 (a-). L'intersection d'une demi-sphère et d'un demi-cube produit un système pendentif.
 (b-). Les quatre triangles sphériques formant les pendentifs.

Le système structural d'un dôme byzantin pendentif est basé sur la transmission des charges d'un dôme demi-sphérique aux sommets du carré de la base du dôme. La coupole demi-sphère repose sur quatre triangles sphériques qui transfèrent la charge de la coupole à quatre arcs périphériques, ces derniers étant regroupés sur les quatre côtés d'un carré. Les pendentifs ont un double rôle. Ils transfèrent la charge du dôme vers les arcs périphériques et ils forment une étape transitoire entre la forme circulaire de la projection du dôme et la forme carrée de la surface couverte par le dôme pendentif (Besenvale, 1984, p.62).

En résumé, la différence entre les deux expériences romaine et byzantine, réside dans la forme géométrique de primitives utilisées dans la construction de ces dômes. Le dôme d'arête romain est basé sur l'utilisation des deux semi-cylindres, alors les pendentifs sont basés sur l'utilisation d'une demi-sphère. En d'autres termes, les deux expériences utilisent deux types différents de primitives. Cependant, elles aboutissent au même résultat, qui est de couvrir un plan carré par un dôme.

2.1.B- la famille de dômes entrelacés

Le VII^{ème} siècle a vu une nouvelle expérience architecturale. Durant la période islamique, l'arc a pris une importance particulière chez les constructeurs musulmans. Ces derniers ont posé la règle islamique qui fixe tracé d'un arc. Les ingénieurs de cette époque ont traité l'arc comme un élément géométrique constructif architectural et symbolique.

Un des arcs islamiques le plus connu est basé sur la division de sa portée en nombre impair. D'après (K. A. C. Creswell, 1979, page-166) l'arc islamique est tracé géométriquement selon la règle suivante:

- 1-Diviser la base ou la portée AB de l'arc en 3, 5 ou 7 divisions (nombre impair).
- 2-Choisir les divisions C1 et C2 les plus proches de la ligne centrale L comme deux centres des deux arcs de cercle (BD et AD). L'ensemble des deux arcs forme l'arc islamique. Celui-ci est géométriquement classé parmi les arcs pointus (figure-20).

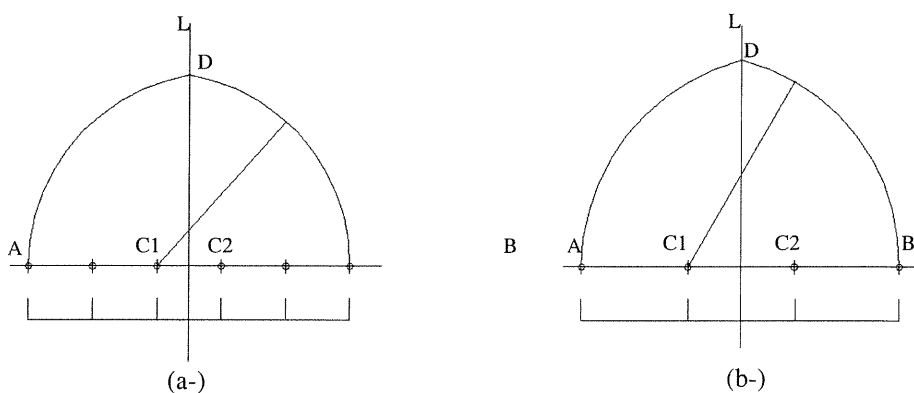
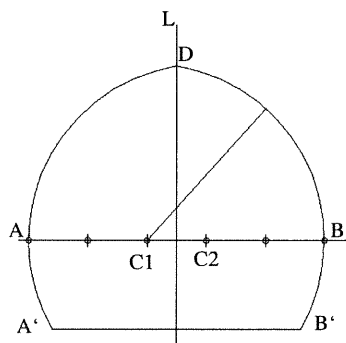
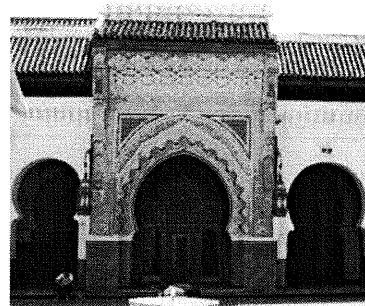


Figure-20. Arc islamique (a-). Arc islamique en 5 divisions. (b-). Arc islamique en 3 divisions.

Du point de vue de la construction, l'arc islamique devient plus grand qu'un angle de 180 degrés. Les constructeurs ont allongé les deux extrémités de l'arc dans le but d'optimiser sa résistance aux forces des poussées horizontales appliquées sur les points A et B. Cette technique a permis aux lignes de compression B`BD et A`AD de suivre les points de la transformation de forces. Selon (Besenval, 1984, p.25) cette technique s'appelle la structure à résistance de forme, qui consiste à donner au matériau, d'un arc par exemple, une forme adaptée aux charges qu'il doit supporter (figure-21-a-b).



(a-)



(b-)

Figure-21. Arc islamique plus qu'un demi-cercle.

(a-). Arc islamique en 5 divisions, les deux extrémités de l'arc sont dans un plan parallèle et au dessous du plan de référence horizontale XOZ où les centres C1 C2 se trouvent.

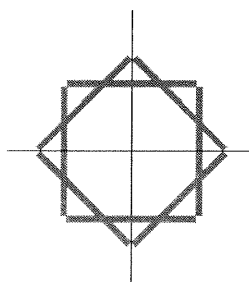
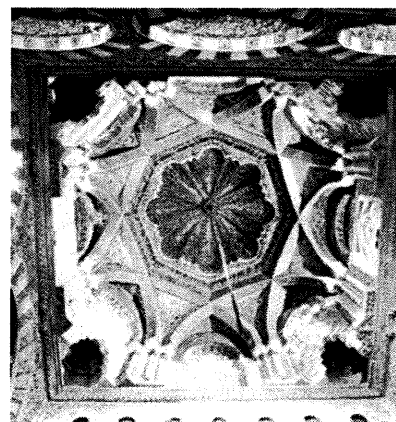
(b-). Deux arcs circulaires latéraux et un arc pointu islamique au milieu (M. Frishman, H et Uddin Khan, 1994, page-108)

Sur le plan architectural, l'arc islamique devient la primitive la plus importante dans la construction des façades à 2D, des structures spatiales à 3D et même dans la construction du système d'aération et d'éclairage. La construction du premier dôme entrelacé, «structure spatiale à 3D», a été réalisée au VIIIème siècle après.J.C. Ce dôme fut construit durant l'occupation arabe de l'Andalousie au sud de l'Espagne. Il couvre le Mihrab de la grande Mosquée de Cordoue. Ce type de dôme fit partie de l'architecture islamique reposant essentiellement sur le respect de la loi islamique et de ce fait, utilisa la géométrie afin de développer l'architecture .

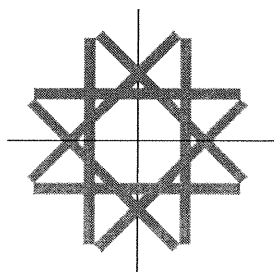
Ce qui est remarquable dans l'architecture islamique, c'est la domination de l'élément géométrique dans la création des éléments architecturaux, constructifs et décoratifs. Cette forte présence de la géométrie dans l'architecture islamique vient de l'interdiction faite par l'Islam de représenter des figures humaines, des animaux et même des dessins artistiques représentant la nature. C'est pour cette raison que les constructeurs et architectes musulmans décoraient leurs édifices, en utilisant des formes géométriques, des mosaïques ou des éléments constructifs comme les arcs et les dômes entrelacés.

On peut dire que le dôme entrelacé est à la fois un élément de construction et un élément de décoration, puisqu'il est utilisé comme dôme externe et comme dôme interne (figure-22). Par exemple, le Mihrab de la grande Mosquée de Cordoue en Andalousie est couvert par ce type de dômes (figure-22). Ce dôme repose sur un octogone qui, à son tour, repose sur une arcade de forme carrée. À partir de cette époque les dômes entrelacés sont adoptés comme un système constructif dans d'autres cultures. Par exemple les Turcs ont utilisé cette technique dans la construction de la Mosquée de *Gol Gumbad* au début du X^{ème} siècle. Ils ont donc adopté la même géométrie que celle du dôme de Cordoue (*domes Istanbul congrès*, 1988) (figure-23-a). À l'époque Baroque, en 1668, l'architecte italien Guarini a construit l'Église de St-Lorenzo en utilisant un dôme entrelacé (figure-23-b). Selon (Adam. R, 1984, p.178) la projection de ce dôme sur un plan horizontal est un octogone, son arc de base est une courbe caténaire et la distribution de ses arcs a une fréquence $n=2$ (la fréquence des dômes entrelacés sera étudiée dans le paragraphe suivant: 2.2). À la même époque, en 1704 *Vittone* a construit le dôme entrelacé du sanctuaire *Vallinotto* avec une base hexagonale et une fréquence $n=1$ (figure-23-c).

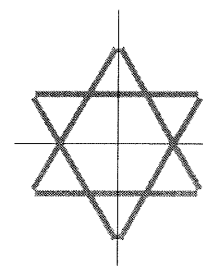
Figure-22. Le dôme entrelacé qui couvre le Mihrab de la grande Mosquée de Cordoue Andalousie (Saudin. M, 1989, p.168).



(a-). Le dôme de Gol Gumbad



(b-). Le dôme de St-Lorenzo



(c-). Le dôme de Vallinotto

Figure-23. Dômes entrelacés

Au cours de l'histoire de l'architecture, les dômes entrelacés sont construits selon différents types d'arcs de base, le nombre d'arcs générés étant variable. Les types de dômes entrelacés dépendent du style de leurs arcs de base, de leurs fréquences et du nombre d'arcs dans leurs structures (voir la figure-23 ci-dessus). Vu la variété des dômes entrelacés, on se demande comment ce système s'est développé.

On peut dire que la relation entre la création du système entrelacé et sa référence historique est fondée sur les points suivants:

1-Le premier point est le transfert du savoir-faire hérité des anciennes civilisations. L'expérience romaine et byzantine a joué un rôle important dans la création du dôme entrelacé. Celui-ci peut être considéré comme le résultat de la rotation d'une voûte d'arrête de 45 degrés autour de son centre O. (figure-24-a-b)

2-Le deuxième point est la généralisation du système entrelacé linéaire à une structure spatiale tridimensionnelle. Les arcs entrelacés d'une façade, par exemple, peuvent être redistribués sur une base octogonale afin d'obtenir un dôme entrelacé (figure-25, 24.b).

3-Le troisième point est l'importance prise par la géométrie dans le développement de l'architecture islamique. Plusieurs primitives, employées dans des tableaux mosaïques peuvent représenter des plans en 2D d'un dôme entrelacé (figure-26). Autrement dit, les constructeurs ont probablement généralisé les données géométriques à 2D, en les effectuant à une structure spatiale tridimensionnelle.

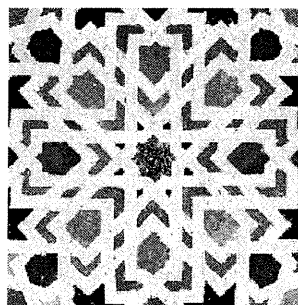
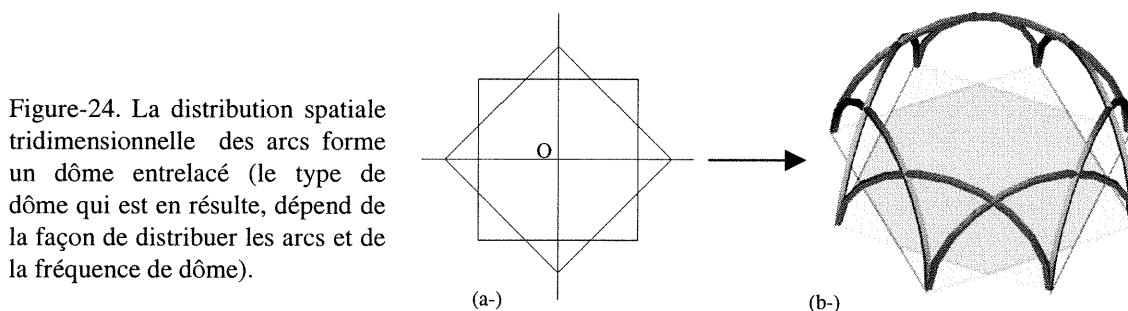


Figure-26. Tableaux mosaïques (Paccard. A, 1983)

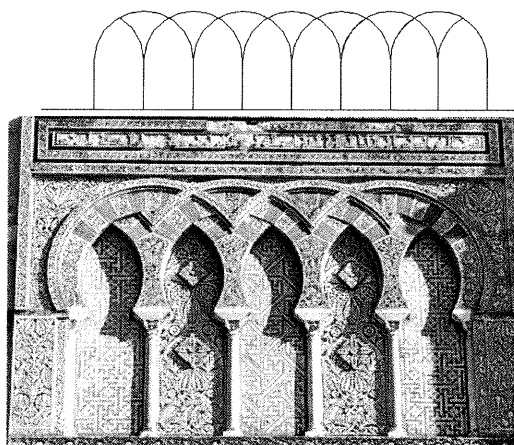


Figure-25. La distribution linéaire des arcs entrelacés. Par exemple l'entrée principale du palais *Alhamra*, Andalousie (Saudin. M, 1989).

2.1.C-Les relations topologiques entre les arcs et les dômes

On a déjà mentionné que l'intersection de deux demi-cylindres peut produire un dôme d'arête romain et, de la même façon, l'intersection d'un demi-cube avec une demi-sphère conduit à un dôme pendentif byzantin. Par contre, la construction de dômes entrelacés est essentiellement basée sur l'assemblage d'arcs. L'analyse topologique montre que ces trois dômes ont la même base constructive. Due à l'assemblage de quatre arcs distribués à même les côtés d'un carré et permettant d'y construire le dôme primitif «D.P» (figure-27). Par exemple un dôme primitif D.P et deux arcs diagonaux génèrent un dôme d'arête. Un dôme D.P et un anneau forment un dôme pendentif et l'assemblage de deux dômes D.P selon une règle géométrique du système entrelacé produit un dôme entrelacé. Cette étude analytique de la topologie des dômes permet de simplifier leurs règles d'assemblage (constructives) et d'unifier leurs sources géométriques et topologiques. En conclusion, les trois systèmes de construction de dômes romains, pendentifs et entrelacés peuvent avoir la même référence géométrique.

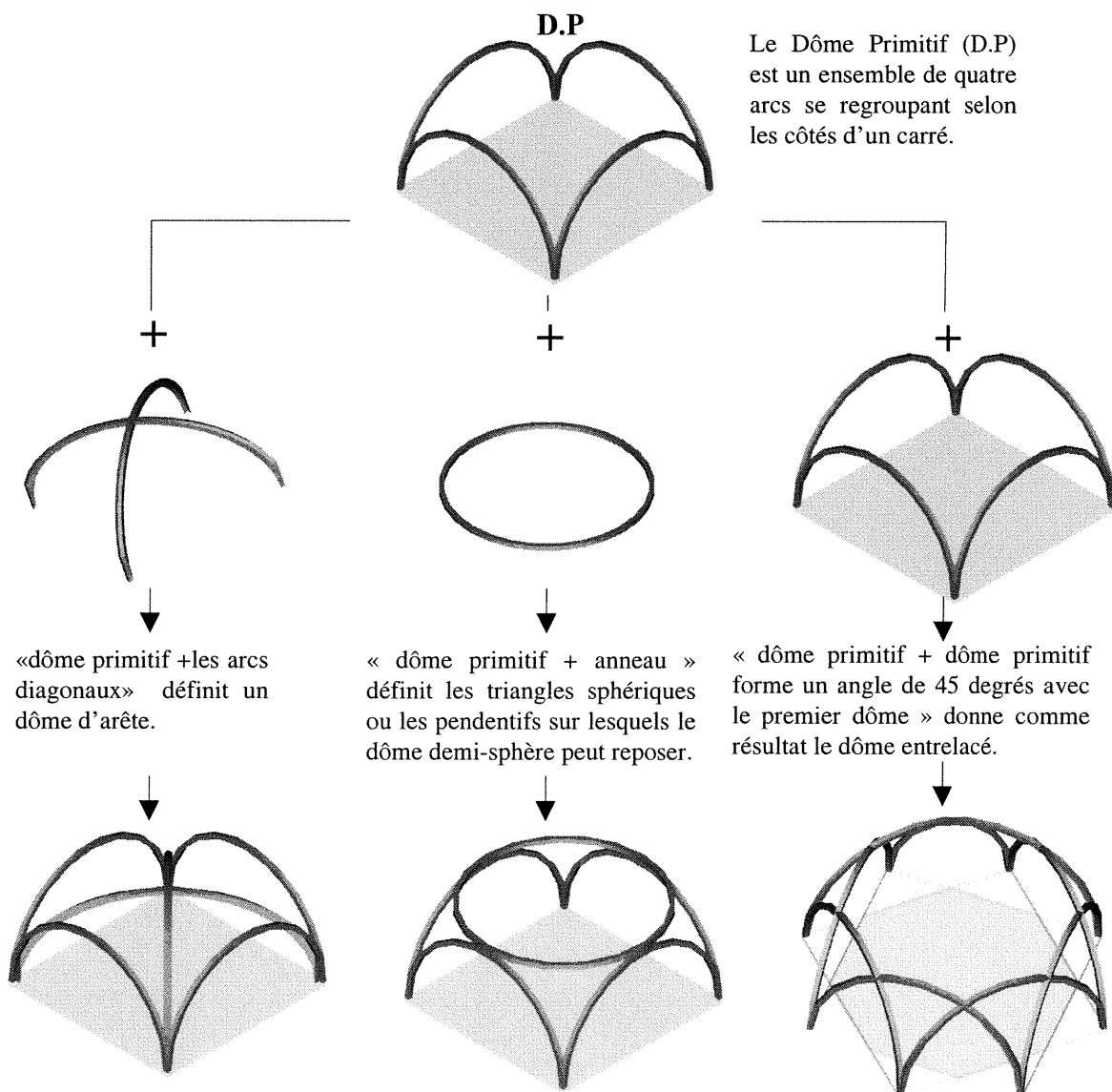


Figure-27. La topologie des dômes arêtes, pendentifs et entrelacés

2.2-Analyse des règles d'assemblage (constructives)

L'arc représente la primitive principale des dômes entrelacés. La manière de regrouper ou d'assembler cette primitive engendre divers types de dômes entrelacés. Il y a deux règles d'assemblage: celle des dômes entrelacés avec une fréquence constante=1 et celle des dômes entrelacés avec une fréquence variable= n .

2.2.A-Règle d'assemblage de dômes entrelacés avec une fréquence constante $n=1$

Les paramètres déterminant cette règle sont:

1-Les polygones réguliers de base: (l'hexagone, l'octogone, le décagone, etc..). Le polygone de base détermine le nombre de sommets servant de points de départ et d'arrivée des arcs. Le paramètre (np) définit le polygone et son nombre d'arrêts.

Par exemple: un octogone ($np = 8$), un hexagone ($np = 6$).

2-Le paramètre (n) est le nombre de sommets entre les points de départ et d'arrivée d'un arc. Dans cette règle, le paramètre (n) est toujours constant $n=1$. C'est-à-dire que l'arc AB, par exemple, relie les sommets A et B et il passe outre (saute) le sommet A' (figure-28). Quel que soit le nombre d'arcs utilisés dans la construction des dômes, la fréquence y est toujours constante, $n=1$.

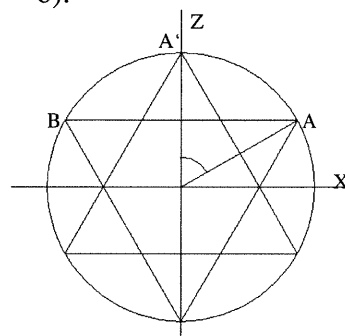


Figure-28. L'alternatif de sommets

Les exemples suivants expliquent la fréquence des dômes octogonaux et hexagonaux:

1-Le dôme entrelacé à base octogonale $np=8$. L'octogone est, ici, le résultat de la rotation du carré ABCD autour de son centre avec un angle $\text{Alfa}=\pi/4$. La forme que l'on obtient est un octogone (A,A',B,B',C,C',D,D') (figure-29-a).

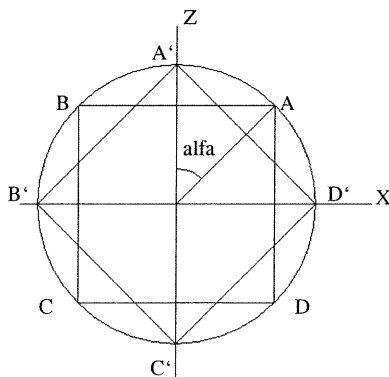
La relation entre l'angle de rotation Alfa et le nombre des sommets est $\text{Alfa}=2\pi/np$.

Dans le cas d'un octogone, $\text{Alfa}=2\pi/8$ $\text{Alfa}=\pi/4$

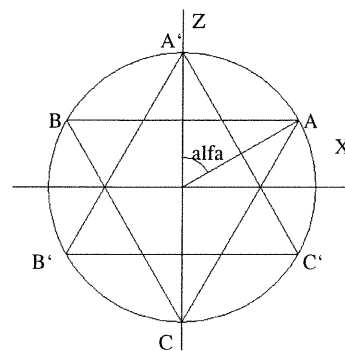
2-Le dôme entrelacé à base hexagonale $np=6$, est le résultat de rotation du triangle ABC autour de son centre en angle Alfa. $\text{Alfa}=2\pi/6$, $\text{Alfa}=\pi/3$. (figure-29-b)

$$\text{Alfa}=2\pi/np$$

Alfa étant l'angle de rotation du polygone de base autour de son centre.



(a). Cas d'un octogone $\text{Alfa}=2\pi/np = 2\pi/8=\pi/4$



(b). Cas d'un hexagone $\text{Alfa}=2\pi/np=2\pi/6=\pi/3$

Figure-29. La fréquence constante du dôme entrelacé

2.2.B-Règle d'assemblage de dômes entrelacés avec une fréquence variable n =variable

La fréquence des dômes faisant partie de cette règle dépend des paramètres « Na », « Te » et « n ».

(Na) est le nombre des arcs utilisés dans la construction du dôme.

(Te) est l'angle du positionnement de l'arc de base AB par rapport à l'axe XX.. L'arc AB est toujours parallèle à l'axe XX. L'angle entre l'axe XX et le point A (le point de départ de l'arc de base AB) est l'angle Te. Celui-ci peut être calculé en fonction de Na $Te = \pi / Na$ (figure-29-a).

(n) est la fréquence de dôme ou le nombre des sommets que l'arc passe outre (passe par-dessus) entre le point de départ et le point d'arrivée. Par exemple (figure-29-a) l'arc AB relie les sommets A et B et il passe outre le sommet (1). On peut dire que le dôme dont l'arc de base passe outre un sommet, a une fréquence $n=1$. Si l'arc passe outre deux sommets, le dôme a une fréquence $n=2$ (figure-29-b).

La relation entre les deux paramètres «Na» et «n»

Si le nombre d'arcs (Na) est connu, on peut calculer le paramètre (n)

la fréquence de dôme utilisant la relation $n = (Na - 4) / 2$ (1)

Si (n) est connu, on calcule Na $Na = (2 * n) + 4$ (2)

Ainsi, on peut calculer la fréquence (n) en fonction de (Na) en utilisant la relation (1).

Les trois dômes entrelacés de la figure-30 ont différentes fréquences. Par exemple:

Dôme hexagonal constitué de 6 arcs, $Na=6$, a une fréquence $n=(6-4)/2=1$ (figure-29-a).

Dôme octogonal constitué de 8 arcs, $Na=8$, a une fréquence $n=(8-4)/2=2$ (figure-29-b).

Dôme décagonal constitué de 10 arcs, $Na=10$, a une fréquence $n=(10-4)/2=3$ (figure-29-c).

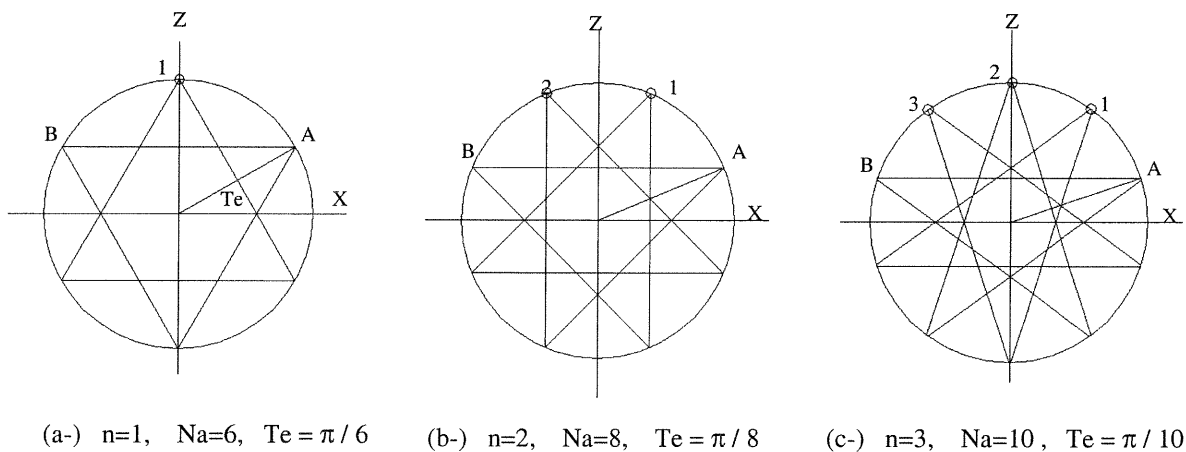


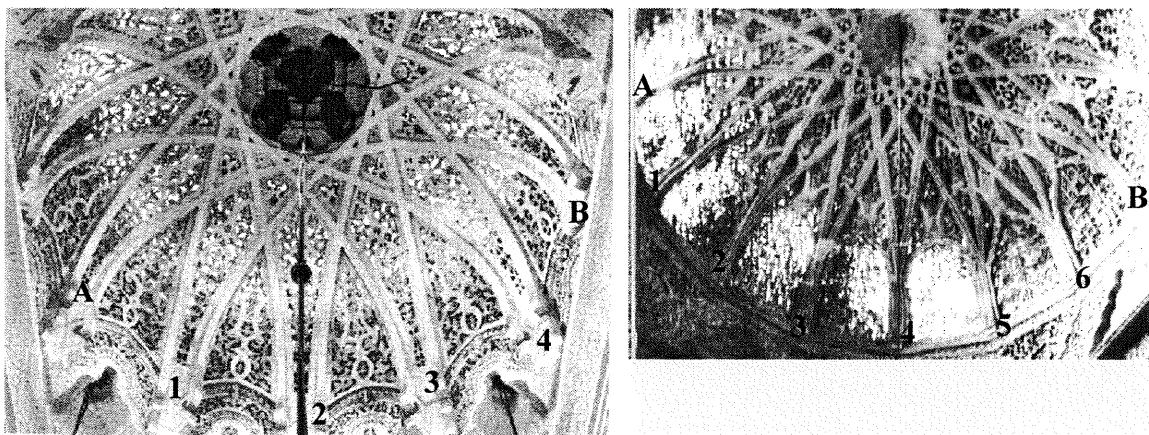
Figure-30. La fréquence variable d'un dôme entrelacé en fonction de Te et Na.

Le dôme entrelacé de la grande Mosquée de Fez, Maroc a une fréquence « $n=4$ » (figure-31-a). Les deux points AB représentent les points de départ et d'arrivée de l'arc AB. Celui-ci passe outre (passe par-dessus) les quatre sommets (1,2,3,4).

La relation précédente (2) $N_a = (2 \cdot n) + 4$ permet de calculer le nombre d'arcs utilisés dans la construction du dôme $N_a = (2 \cdot n) + 4$ $N_a = (2 \cdot 4) + 4 = 12$ arcs.

On peut aussi calculer l'angle T_e (T_e est l'angle entre l'extrémité A de l'arc A,B et l'axe XX'). De la relation (3) $T_e = \pi / N_a$ on calcule $T_e = \pi / 12$ $T_e = 15$ degrés.

Le paramètre N_a est le nombre d'arcs utilisés dans la construction d'un dôme entrelacé, (N_a) pourra être un des paramètres de la future structure algorithmique «programme».



(a)- Le dôme entrelacé de la grande Mosquée de Tlemcen, Algérie. La fréquence du dôme $n=4$ sommets, et $N_a = 12$ arcs et $T_e = \pi / 12$.

(b)- Le dôme entrelacé de la grande Mosquée de Fez, Maroc. La fréquence $n=6$ sommets, $N_a=16$ arcs, $T_e = \pi / 16$.

Figure-31. Exemples de dômes entrelacés avec une fréquence variable (M. Frishman, 1994, page-106).

2.2.C- Résumé de la règle d'assemblage (constructive) de dômes entrelacés

La règle d'assemblage de dômes entrelacés peut être résumée ainsi:

- 1-Les extrémités de chacun des deux arcs se rencontrent au sommet du polygone de base du dôme entrelacé.
- 2-Le nombre des arcs qui s'entrelacent ou s'entrecroisent est toujours un nombre pair.
- 3-Il faut au minimum six arcs pour assurer un entrecroisement ou un entrelacement respectant toutes les autres conditions d'assemblage.
- 4-Il faut que l'arc de base d'un dôme entrelacé passe par-dessus, au moins, un sommet, situé entre son point de départ et son point d'arrivée afin d'assurer un entrelacement.
- 5-La fréquence du dôme ($n=1$) est toujours constante, si le positionnement de l'arc de base du dôme est contrôlé par l'angle $\text{Alfa} = 2\pi / n_p$.
- 6-La fréquence du dôme (n) est variable, si le positionnement de l'arc de base du dôme est contrôlé par l'angle $T_e = \pi / N_a$.

2.3-Analyse géométrique des arcs

Le but de cette analyse est d'établir la base numérique qui peut être traduisible informatiquement en programme permettant de générer différents types d'arcs appartenant à la même famille géométrique. Cette analyse nous permettra de:

1-Caractériser les différentes configurations de l'arc de base utilisé dans la construction des dômes entrelacés. La caractérisation formelle de l'arc peut être intéressante lors du choix des paramètres qui permettront de contrôler la génération de la grande variété des nouvelles formes de dômes entrelacés. L'arc est la primitive principale dans ce type de dômes. La définition géométrique de cette primitive s'appuie sur l'utilisation d'une ou de plusieurs courbes géométriques. L'arc romain, par exemple, est représenté par un seul demi-cercle, tandis que l'arc gothique et l'arc islamique sont définis par deux arcs de cercle. D'autres types d'arcs connus à l'époque baroque et pendant la Renaissance sont définis par des courbes caténares. La courbe caténaire peut être tracée par une chaîne suspendue et fixée aux deux extrémités. Ce type de courbe est très proche d'une parabole.

2-Relever la règle géométrique qui contrôlent l'assemblage des dômes entrelacés impliquant tous les cas de configurations des arcs de base. Les équations et les données numériques résultant de cette analyse sont des représentations mathématiques de la règle d'assemblage des dômes entrelacés (voir le paragraphe précédant 2.2.C).

Ainsi, la modélisation des dômes entrelacés dépend, d'une part, de l'analyse constructive de leurs règles d'assemblage et d'autre part de l'analyse géométrique de leurs arcs de base. Ces derniers peuvent être classifiés en deux catégories de courbes:

Première catégorie: les courbes (les arcs) subissant des discontinuités surfaciques. Ce genre d'arc peuvent être engendré à partir de deux arcs de cercle. L'arc gothique, l'arc islamique et le cas général de génération d'arcs pointus font partie de ces courbes.

Deuxième catégorie: les courbes (les arcs) ont une continuité surfacique. Ce genre d'arcs peut être généré à partir d'un arc de cercle, d'une ellipse ou d'une parabole. Selon le type de courbe utilisé, on obtient trois formes d'arcs: une forme surhaussée, un demi-cercle et une forme surbaissée.

Ces deux dernières catégories représentent la base géométrique des arcs primitifs pouvant être utilisés dans la construction des deux principaux types des dômes entrelacés. Ainsi cette étude a-t-elle comme objectif l'analyse des deux types de dômes entrelacés suivants:

1-Les dômes entrelacés dont les arcs se définissent par une discontinuité surfacique et qui constituent le cas général. La règle islamique et la règle gothique seront analysées comme deux cas particuliers. Dans cette analyse, on recherchera comme résultat des données géométriques, des données numériques intégrées dans un algorithme pouvant générer les dômes. Ces données sont indépendantes du système informatique utilisé dans la modélisation.

2-Les dômes dont les arcs forment une surface continue. Dans cette analyse, on s'appuiera sur la capacité du système informatique utilisé et sur le résultat de l'analyse géométrique et graphique afin de générer un dôme entrelacé à base elliptique.

2.3.A- Analyse géométrique des arcs subissant des discontinuités surfaciques

L'arc de base de ce type de dômes subit des discontinuités surfaciques. On appelle ce type d'arc, arc pointu ou croisé. L'arc pointu est le résultat de l'intersection de deux arcs de cercle. Ainsi, deux courbes qui forment ce type d'arc font partie d'un cercle (figure-32-a). L'angle, qui définit l'arc, est l'angle (G_a). Selon la valeur de G_a , on obtient trois cas principaux:

Cas-1: $0 < G_a < \pi/2$, produit l'arc AB (la figure-32-b).

Cas-2: $G_a = \pi/2$, produit l'arc AC (la figure-32-c).

Cas-3: $\pi \geq G_a > \pi/2$, produit l'arc AD (la figure-32-d).

Les trois courbes géométriques (AC, AB et AD) constituent trois types d'arcs de base:

1-L'arc AB représente la moitié d'un arc pointu. Pour avoir un arc pointu complet, on construit l'autre moitié BA'. Les deux moitiés de l'arc pointu ABA' sont symétriques l'une de l'autre, par rapport à sa ligne centrale L (figure-32-b).

2-L'arc AC est un quart de cercle qui est défini par l'angle $G_a = \pi/2$. L'arc ACA' est un arc complet représenté par un demi-cercle. Dans cette étude, un arc en demi-cercle peut être considéré comme cas particulier d'un arc pointu (figure-32-c).

3-L'arc AD représente un arc brisé ou croisé. Cet arc est le troisième type d'arcs construits à partir des deux arcs de cercle et qui subissent des discontinuités surfaciques. Dans ce cas, la valeur numérique de l'angle G_a est située dans l'intervalle: ($\pi \geq G_a > \pi/2$).

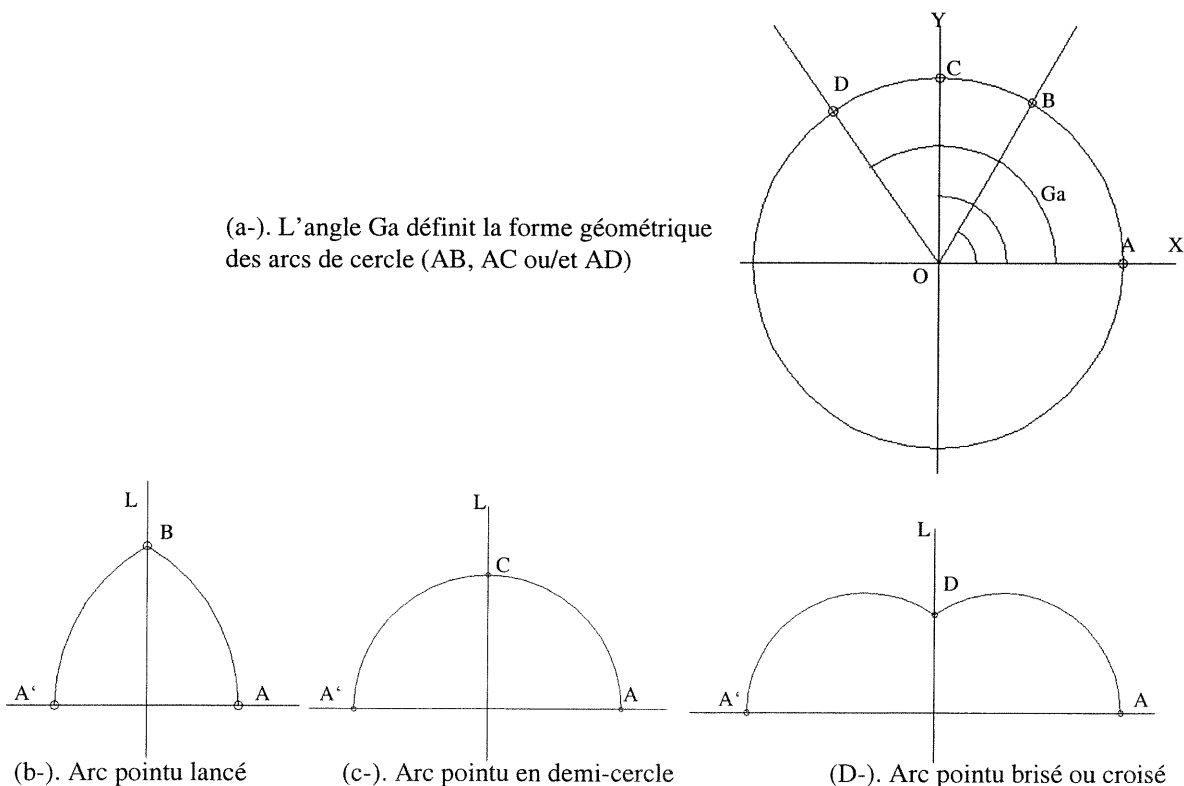


Figure-32. Les différentes configurations de l'arc pointu

Le principe géométrique de la construction d'un arc subissant des discontinuités surfaciques

Le but de cette analyse est de relever les données numériques qui permettent d'intégrer les trois cas ou les trois types d'arcs pointus dans un contexte géométrique et numérique global. Trois plans principaux participent à la construction de l'arc de base:

1-Le plan de référence horizontal XOZ qui est représenté par le cercle CEH et/ou la projection horizontale du dôme.

2-Le plan de référence vertical ZOY qui passe par le centre du dôme.

3-Le plan B, D_1, N sur lequel on trace l'arc de base. Ce plan est toujours parallèle au plan de référence XOY. Il se déplace sur l'axe ZZ' selon la distance OC. Cette distance est contrôlée par l'angle $Te = \pi/Na$, Na étant le nombre d'arcs utilisés dans la construction du dôme entrelacé.

Les trois plans précédents délimitent géométriquement la moitié de l'arc de base BD_1 . L'autre moitié AD_1 étant la symétrie de l'arc BD_1 , par rapport au plan de référence ZOY (figure-33). Les points C_1 , C et B sont reliés par des relations procédurales. Le point C_1 est le centre de l'arc BD_1N . Le point C est le centre de l'arc de base dans le cas où cet arc est représenté par un demi-cercle. Les points A et B sont les deux extrémités de l'arc de base AD_1B . Selon les positions des trois points C, C_1 et B l'un par rapport aux autres, on peut caractériser trois configurations d'arc de base:

1-La distance $BC_1 > BC$ produit un arc pointu lancé.

2-La distance $BC_1 = BC$ produit un arc demi-cercle.

3-La distance $BC_1 < BC$ produit un arc brisé ou croisé.

Le centre C_1 se déplace sur la ligne L en direction BN. La position de C_1 sur L change selon la valeur de l'angle G_a . Dans ce cas, la position de C_1 dépend de l'angle G_a et de la distance BC_1 .

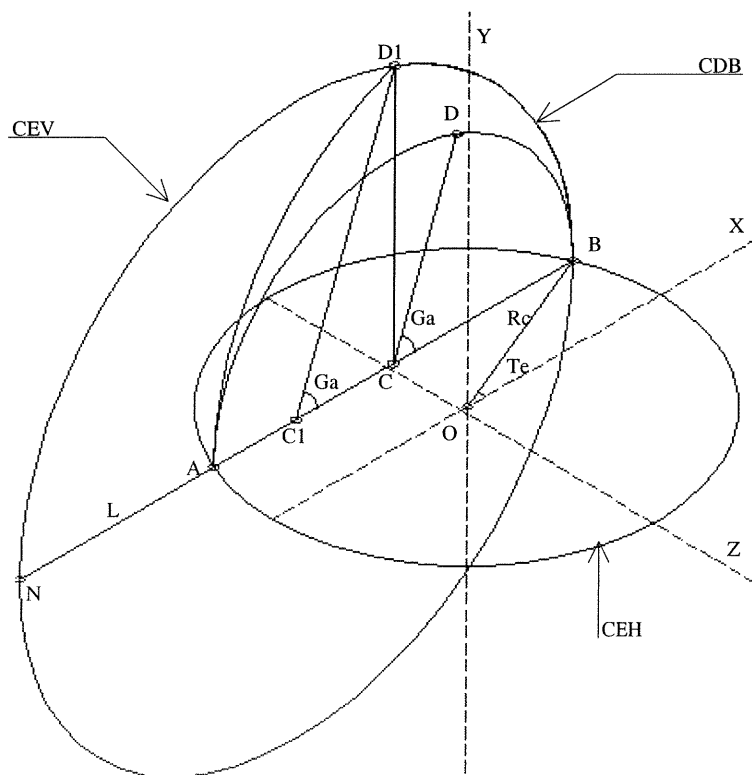


Figure-33. Les trois plans de construction de l'arc pointu:

CEV est le cercle ou le plan vertical sur lequel on trace l'arc de base.

CEH est le cercle ou le plan horizontal représentant la projection du dôme sur le plan de référence XOZ.

CDB est le cercle de base dont le diamètre AB est la projection de l'arc de base A,B,D1 sur le plan de référence XOZ.

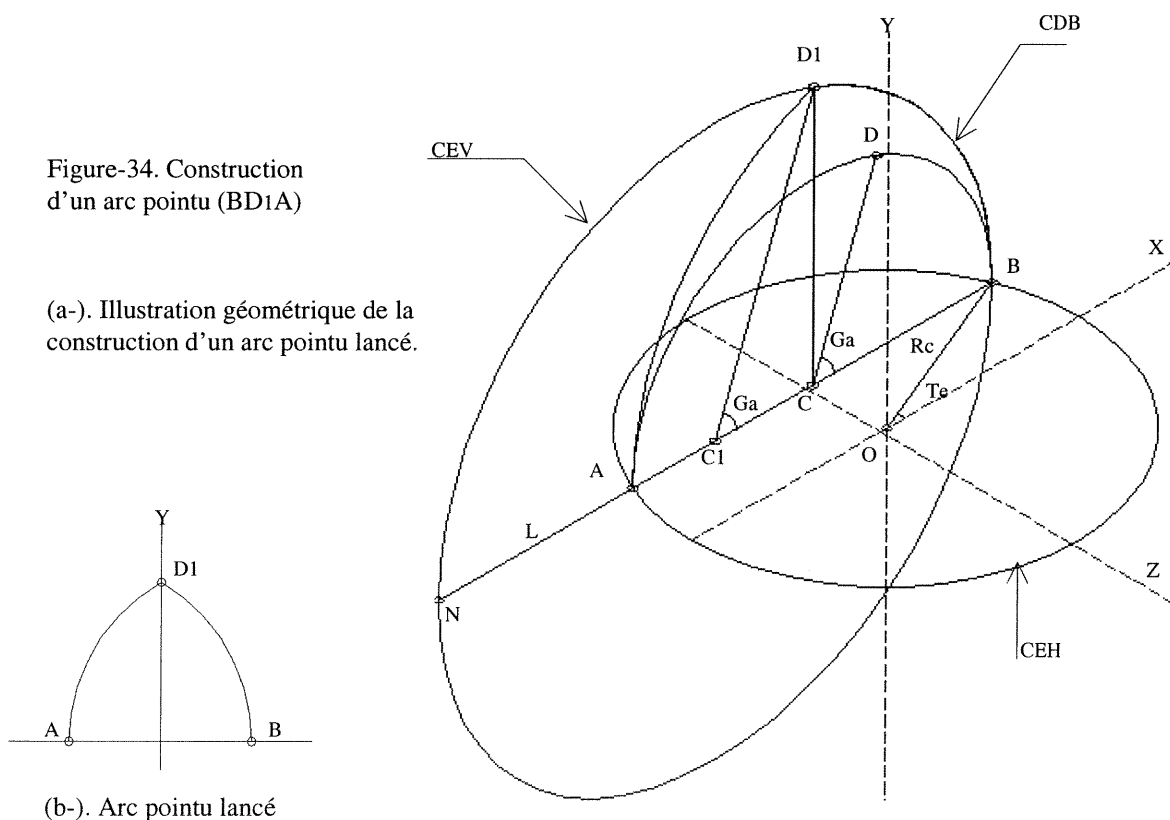
2.3.A.1-Étude des trois cas de la position du centre C_1 par rapport au centre C :

Cas-1: $0 < \alpha < \pi/2$ et $BC_1 > BC$

Le Cas-1 produit un arc pointu lancé. Tant que la valeur de l'angle α est dans l'intervalle $0 < \alpha < \pi/2$, la distance BC_1 reste supérieure à la distance BC . Le point D se déplace sur le demi-cercle BDA . La position du point D sur ce cercle dépend de l'angle α . En même temps, le point D_1 se déplace sur le demi-cercle BD_1N . La position du point D_1 sur ce cercle dépend aussi de l'angle α . En d'autres termes, les deux points D et D_1 sont contrôlés par le même angle α . Les deux rayons CD et C_1D_1 sont parallèles et le rayon C_1D_1 est plus grand que le rayon CD . Dans ce cas, l'arc de base BD_1 , que l'on obtient, est toujours inférieur à un quart de cercle.

L'angle $\alpha = \angle BCD = \angle BC_1D_1$. α contrôle alors la rotation des deux rayons CD et C_1D_1 qui sont toujours parallèles. Au fur et à mesure que l'angle α décroît, le rayon C_1D_1 croît et la distance qui sépare les deux centres C_1 et C devient, elle aussi, plus grande (figure-34-a)

En résumé, l'angle $\angle BCD$ et le rayon CD guident l'angle $\angle BC_1D_1$ et le rayon C_1D_1 . L'arc obtenu grâce à cette opération est l'arc BD_1A . Il est symétrique par rapport au plan ZOY (figure-34-b).



Cas-2: $Ga=\pi/2$ et $BC_1=BC$

Le Cas-2 produit un arc en demi-cercle ou un arc neutre. Cet arc, a l'apparence d'un demi-cercle, mais sa géométrie subit une rupture surfacique au point D.

Dans ce cas, la distance $BC_1=BC$, les deux centres C_1 et C sont alors superposés, ainsi que les points D et D_1 et les deux demis-cercles BDA et BD_1N .

L'angle $Ga=\pi/2$ représente le cas particulier de la génération d'arcs subissant des discontinuités surfaciques (figure-35-a-b). La forme de cet arc est continue, mais en réalité, elle constitue deux arcs de cercle ou deux quarts de cercle, chacun d'eux est délimité par un angle $Ga=\pi/2$.

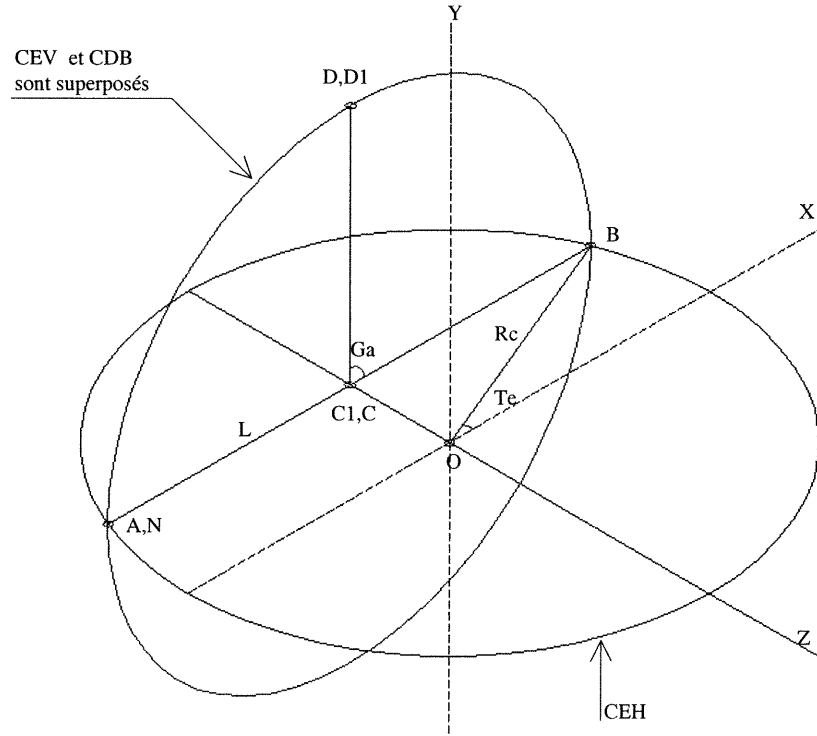
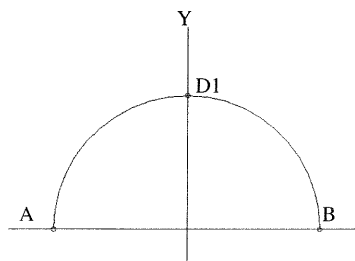


Figure-35. Construction d'un arc pointu en demi-cercle

(a-). Illustration géométrique d'un arc en demi-cercle ou un arc neutre.

(b-). Arc pointu en demi-cercle



Cas-3 : $\pi > Ga > \pi/2$ et $BC_1 < BC$

Le Cas-3 produit un arc brisé ou croisé. Il est exactement l'inverse du cas-1. Dans le cas-3, la valeur de l'angle G_a est dans l'intervalle $\pi \geq Ga > \pi/2$. Le centre C_1 se déplace sur la ligne L entre l'extrémité de l'arc B et le centre C . Ce qui veut dire que la distance BC_1 est inférieure à la distance BC . Le point D se déplace sur le demi-cercle BDA ce déplacement est contrôlé par l'angle G_a . En même temps, le point D_1 se déplace sur le demi-cercle BD_1N contrôlé par le même angle G_a . Les deux rayons CD et C_1D_1 sont parallèles; Ce qui veut dire que l'angle $G_a = \angle BC_1D_1$ varie en fonction de l'angle $G_a = \angle BCD$. Dans ce cas, l'arc BD_1 , que l'on obtient, est toujours supérieur à un quart de cercle (figure-36-a-b).

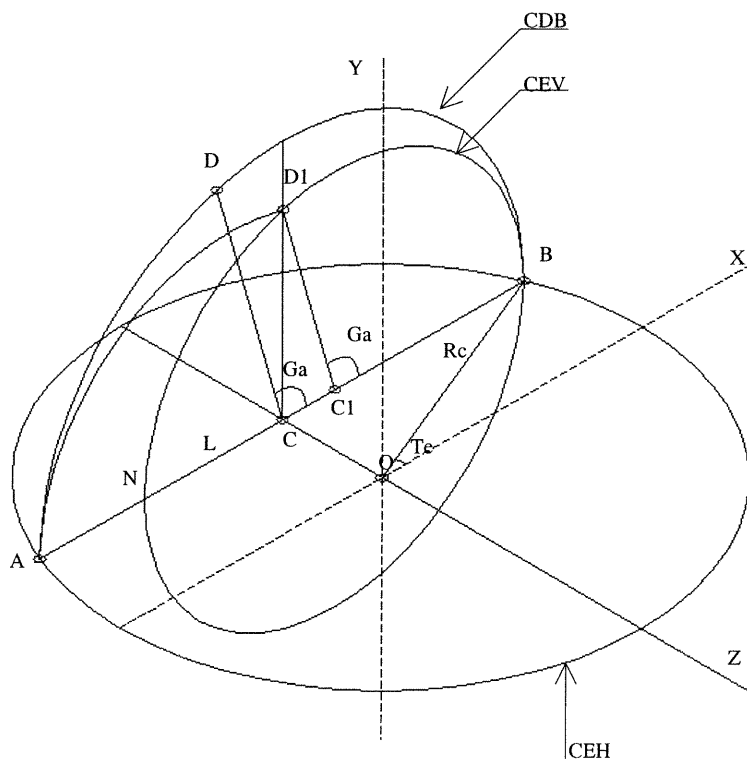
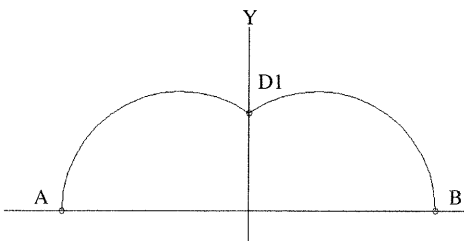


Figure-36. Construction d'un arc brisé

(a). Illustration géométrique de la construction d'un arc brisé ou croisé.

(b). Arc brisé ou croisé



2.3.A.2-Étapes de construction de l'arc de base subissant des discontinuités surfaciques

La construction de l'arc de base subissant des discontinuités surfaciques passe par trois étapes principales. Quelle que soit la forme géométrique de cet arc pointu, demi-cercle ou brisé, les étapes de construction sont toujours les mêmes. Les relations numériques que l'on obtiendra sont valables pour les trois types: arc pointu, demi-cercle et brisé ou croisé. Le but de cette opération est de construire l'arc de base.

Le principe de la construction

D'abord on trace le cercle (BDAE) dans un plan parallèle au plan de référence XOY et perpendiculaire au plan de référence XOZ (figure-37-I). La position de ce plan par rapport au plan de référence XOY est contrôlée par l'angle $T_e = \pi / Na$. Trois étapes sont nécessaires pour construire l'arc de base

Étape-1. «les relations verticales»: éliminer l'arc DAE en utilisant le plan de la construction verticale ZOY. On obtient alors l'arc DBE.

Étape-2. «les relations horizontales»: éliminer l'arc BE en utilisant le plan de la construction horizontale XOZ, on obtient alors l'arc BD.

Étape-3. «rétablir la géométrie de l'arc de base»: construire l'arc AD qui est symétrique à l'arc BD par rapport au plan ZOY. On obtient alors l'arc de base BDA.

Dans les paragraphes suivants, nous allons étudier en détails ces trois étapes, tout en reconstituant leurs données numériques.

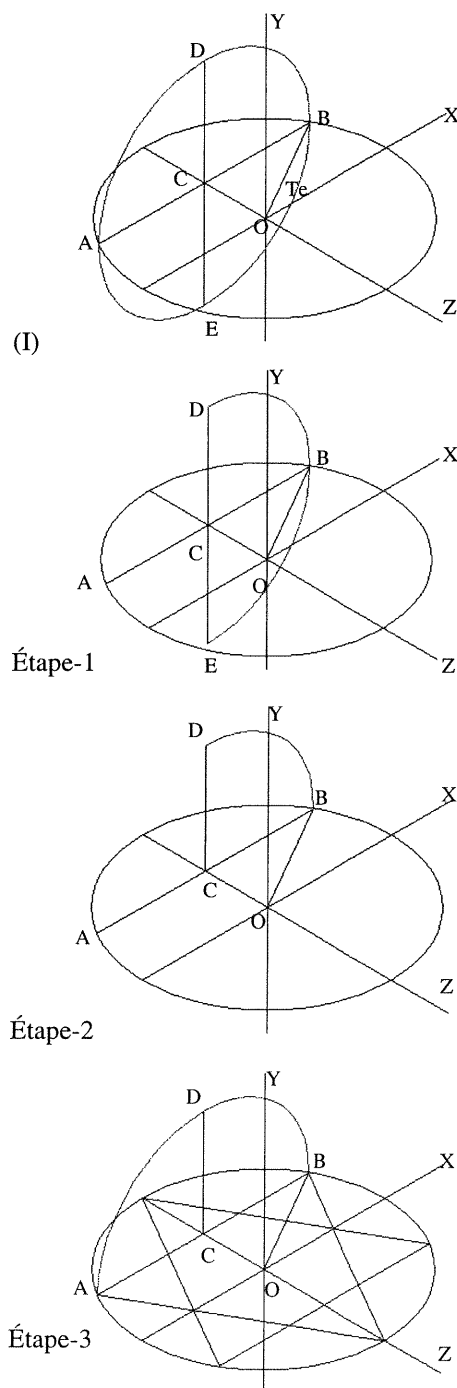


Figure-37. Les étapes de la construction de l'arc pointu.

2.3.A.2.I-Étape-1: Étudier les relations de la construction verticales de l'arc de base: Le plan de référence ZOY est le plan de la construction verticale. Celui-ci passe par le centre du dôme O. Le but de cette étape est de localiser l'arc BD_1N par rapport au plan de référence ZOY. Pour ce faire, on définit les paramètres nécessaires, on calcule les variables et enfin on établit les équations et les relations numériques qui s'intègrent dans le modèle numérique (figure-38).

Les données ou les paramètres:

(Rc) est le paramètre Rc est le rayon du dôme, $Rc=OB$.

(Na) est le paramètre Na est le nombre d'arcs utilisés dans la construction du dôme. Na nous permet de calculer (Te) où Te est l'angle formé par OB. Te est situé entre le rayon OB et l'axe XX' , (le point B est l'extrémité de l'axe de base BD_1A). ($Te=\pi/Na$, voir le paragraphe 2.2.B, chapitre-II).

(Ga) est l'angle qui définit verticalement la moitié de l'arc de base BD_1 .

(Rex)est la variable $Rex=AB/2$, AB étant la projection de l'arc de base sur le plan XOZ. Cette projection (AB) est toujours parallèle à l'axe XX' .

$Rex=Bc$ représente la projection de la moitié de l'arc de base sur le plan de référence XOZ ($Rex=CB=CD=CA$). La variable $Rex=Rc*\cos Te$.

(Rex_n)est la variable Rex_n représente le rayon du cercle CEV.

(E2c) est la variable E2c est la distance entre les deux centres C et C1.

Ces données représentent les paramètres et les variables qui contrôlent la construction verticale de l'arc de base.

L'objectif de l'étude de l'étape-1 est de calculer les variables « Rex_n et E2c »:

1-Le rayon Rex_n du cercle BD_1N : ($Rex_n=C_1B=C_1D_1$) Rex_n est le rayon du nouveau cercle CEV dont le centre est C1. Le rayon Rex_n est contrôlé par l'angle Ga. Au fur et à mesure que l'angle Ga décroît, le rayon Rex_n devient plus grand et le centre C1 s'éloigne de plus en plus du centre C sur la ligne L, en direction BN. (figure-38)

2-La distance $E2c=C_1C$; E2c est la distance qui sépare les deux centres C1 et C (figure-38).

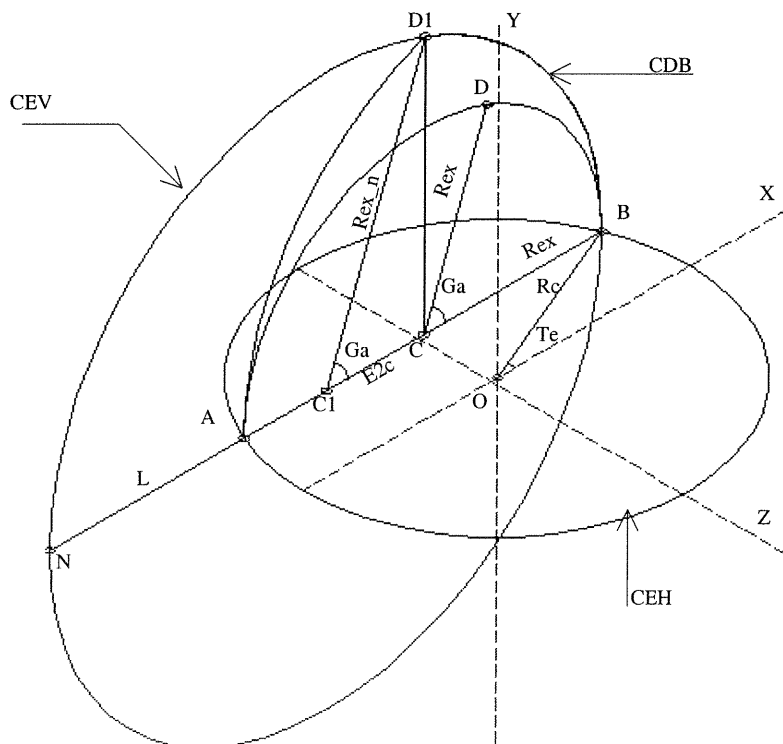


Figure-38. La construction verticale de l'arc pointu.

Le but de l'analyse de l'étape-2 est de localiser le plan de la construction horizontale PCH par rapport au plan de référence XOZ. Dans la figure-39 ci-dessous, ce plan est représenté par le cercle PCH.

Les données et les paramètres:

(Rc) est le rayon du dôme entrelacé où $Rc=OB$.

(Na) est le nombre d'arcs utilisés dans la construction du dôme.

(Ga) est l'angle qui définit verticalement l'arc BD_1 . Dans la figure-39 l'angle Ga est représenté par l'angle BC_1D_1 .

(Rex) est la projection de la moitié de l'arc de base BD_1 sur le plan de référence XOZ où $Rex=BC$.

Tous les paramètres ci-dessus sont déjà définis dans le paragraphe 2.3.A.2.I. Le nouveau paramètre dans cette étape est l'angle Be.

(Be) est l'angle qui définit horizontalement l'arc de base. Le plan de la construction horizontale PCH dépend de l'angle Be. Ce plan se déplace verticalement sur l'axe YY' tout en restant parallèle au plan de référence horizontale XOZ.

L'objectif de l'étude de l'étape-2 est de calculer la variable Y_{cut} .

$(Y_{cut})=KE$: la variable « Y_{cut} » représente la distance entre le plan de référence XOZ et le plan de la construction horizontale PCH. Y_{cut} dépend de l'angle Be. Celui-ci contrôle le déplacement du plan PCH sur l'axe YY' . L'angle Be prend sa valeur de l'intervalle $(-Ga < Be < Ga)$ (figure-41).

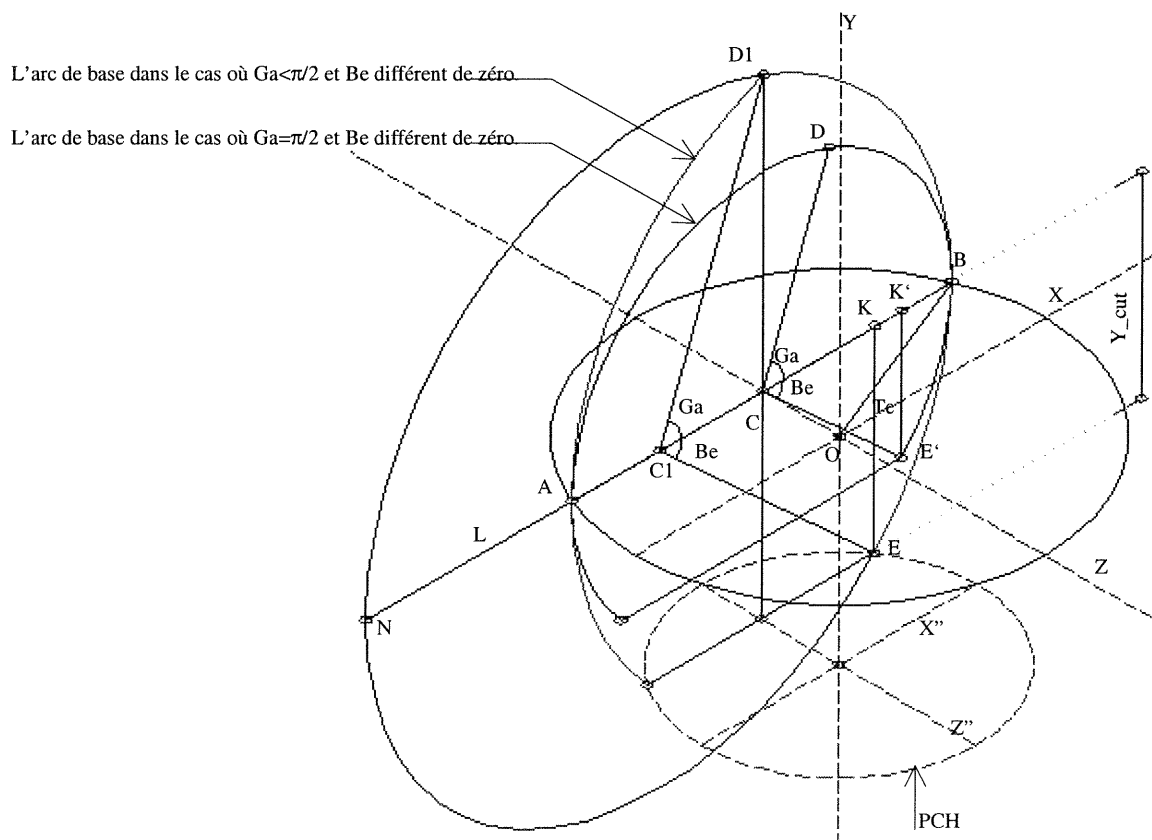


Figure-41. Illustration 3D montre la construction horizontale de l'arc pointu

Relations mathématiques de la construction horizontale de l'arc de base

Calcul de la variable «Y_cut» :

Si $Be=0 \rightarrow Y_{cut}=0$. Dans ce cas, les deux plans XOZ et PCH sont superposés autrement Y_{cut} prend une valeur dépendante de l'angle Be .

(Y_{Be}) est la hauteur du plan de la construction horizontale PCH par rapport au plan XOZ. Dans le cas où $Ga=\pi/2$, l'arc de base devient un demi-cercle, les centres C_1 et C sont superposés et les points E et E' sont superposés (figure-42).

Calculer Y_{cut} dans le cas général où Be n'est pas zéro:

On a $CB=CE'=Rex$,

$C_1B=C_1E'=Rex_n$ et $Y_{Be}=K'E'$

Du triangle $CE'K'$ on a

$\sin Be = K'E' / CE'$

on substitue à $K'E'$ et CE' leurs valeurs

$$\sin Be = Y_{Be} / Rex \quad (1)$$

À partir de la relation (1) on calcul Y_{Be}

$$Y_{Be} = Rex * \sin Be$$

Dans le triangle C_1EK , on a la relation

$$\sin Be = KE / C_1E \rightarrow$$

$$\sin Be = Y_{cut} / Rex_n \quad (2)$$

De la relation (2), on déduit:

$$Y_{cut} = Rex_n * \sin Be$$

Pour calculer Y_{cut} en fonction de Y_{Be} , On a les deux équations (1) et (2) qui sont égales, c'est-à-dire

$$Y_{Be} / Rex = Y_{cut} / Rex_n \quad (3)$$

La relation (3) nous permet de calculer (Y_{cut}) qui représente la distance entre le plan de construction horizontale PCH et le plan de référence XOZ.

$$Y_{cut} = Y_{Be} * Rex_n / Rex \quad II$$

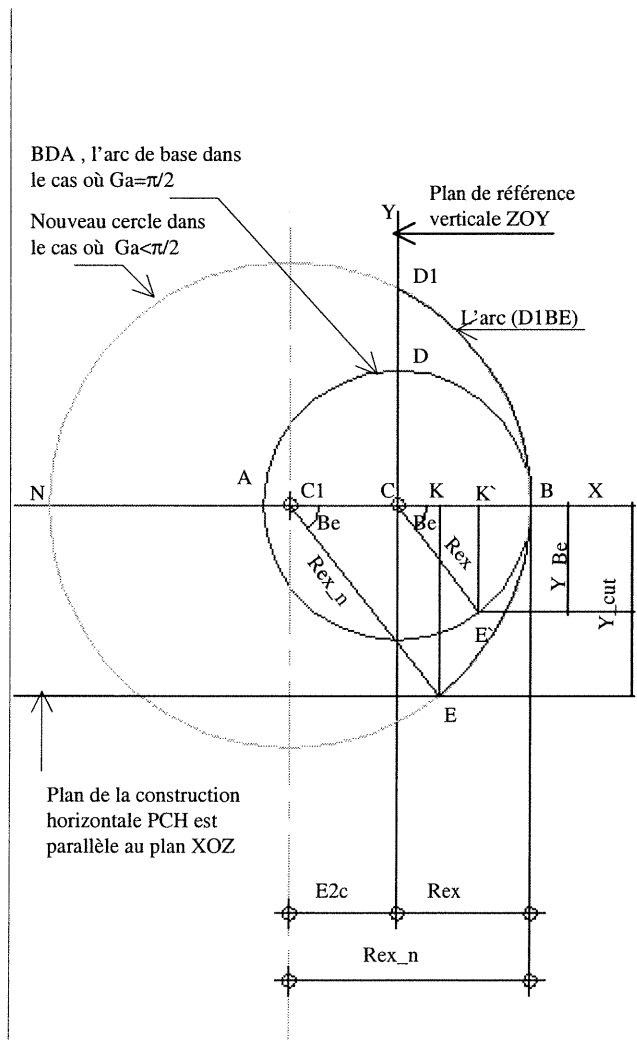


Figure-42. Illustration 2D montrant les paramètres et les variables de la construction horizontale de l'arc pointu

2.3.A.2.III-Étape-3: Rétablir la géométrie du dôme entrelacé et repositionner l'arc de base à la suite de l'opération de la construction horizontale

Au paragraphe 2.3.A.2.II, nous avons calculé la variable « Y_{cut} » qui représente la distance entre le plan de construction horizontale PCH et le plan de référence XOZ. Le dôme subit dès lors des changements dans sa géométrie suite au découpage de son arc de base. Ce changement ne pose aucun problème au niveau de la construction de l'arc de base, mais la difficulté apparaît lorsque cet arc est généré et utilisé dans la construction du dôme entrelacé. Il est important que les règles d'assemblage du dôme entrelacé soient respectées durant la construction de son arc de base. L'une de ces règles contrôle la position des arcs du dôme comme il est dit au paragraphe 2.2.C, chapitre-II: «Les extrémités de chacun des deux arcs se rencontrent au sommet du polygone de base du dôme entrelacé».

Le plan de construction horizontal PCH coupe l'arc de base suivant une ligne verticale Y_{cut} . Celle-ci est contrôlée par l'angle Be (figure-43).

Si l'angle $Be=0$, il n'y aura pas de changement dans la position de l'arc de base généré autour du centre du dôme. Si, au contraire, l'angle Be prend une valeur des intervalles $(-Ga < Be < 0)$ ou $(Ga > Be > 0)$, il faut alors repositionner l'arc de base, afin que la génération du dôme respecte la règle d'assemblage sus-mentionnée. La figure-43 illustre bien les trois configurations du plan PCH par rapport au plan de référence XOZ.

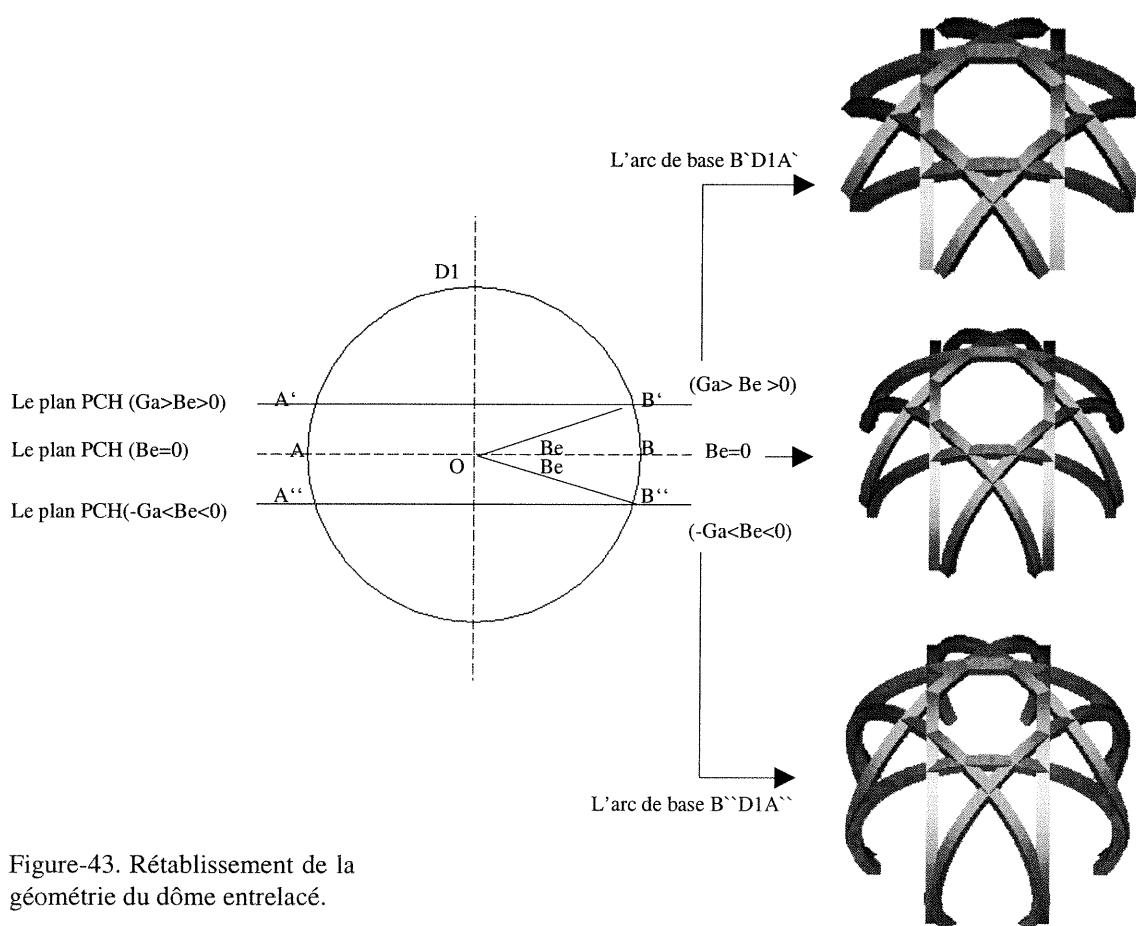


Figure-43. Rétablissement de la géométrie du dôme entrelacé.

Les données et les paramètres (figure-44)

Les paramètres et les variables utilisés dans cette analyse sont:

(Rc): le rayon du dôme. $Rc=OB$, il est situé dans plan XOZ.

(Na): le nombre d'arcs utilisés dans la construction du dôme. On utilise Na afin de calculer l'angle Te , (Te) étant l'angle formé par OB (le point B étant l'extrémité de l'arc de base) et l'axe XX' , ($Te=\pi/Na$).

(Ga): $Ga=BC_1D_1$, Ga est l'angle de construction verticale de la moitié de l'arc de base.

(Rex) est la projection de la moitié de l'arc de base EBD_1 sur le plan de référence XOZ, $Rex=BC$.

(Be): $Be=BC_1E$, Be est l'angle de la construction horizontale de l'arc de base. Le plan de la construction horizontale PCH dépend de l'angle Be. Ce plan se déplace sur l'axe YY' mais étant toujours parallèle au plan de référence XOZ.

(Dc): $Dc=OC$, Dc est la distance entre le centre du dôme et la face interne de l'arc de base, $Dc=Rc * \tan Te$.

Il y a aussi des variables qui ont été calculées au cours des étapes précédentes.

Du paragraphe 2.3.A.2.I, on a deux variables connues (Rex_n) et ($E2c$)

(Rex_n): « $Rex_n=C_1D_1$ » est le rayon de l'arc de base D_1BE où $Rex_n=Rex/(1-\cos Ga)$

($E2c$): « $E2c=C_1C$ » $E2c$ est la distance entre les deux centres C_1 et C $E2c=Rex_n-Rex$

Du paragraphe 2.3.A.2.II, on a la variable (Y_{cut})

(Y_{cut}) est la distance verticale entre le plan de référence XOZ et le plan de la construction horizontale PCH.

$Y_{cut}=Y_{Be} * Rex_n / Rex$

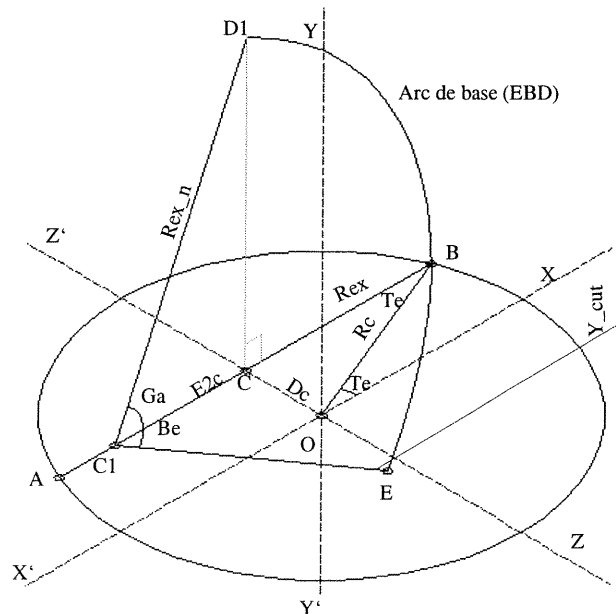


Figure-44. Les paramètres et les variables principalement utilisés dans le rétablissement de la géométrie du dôme entrelacé.

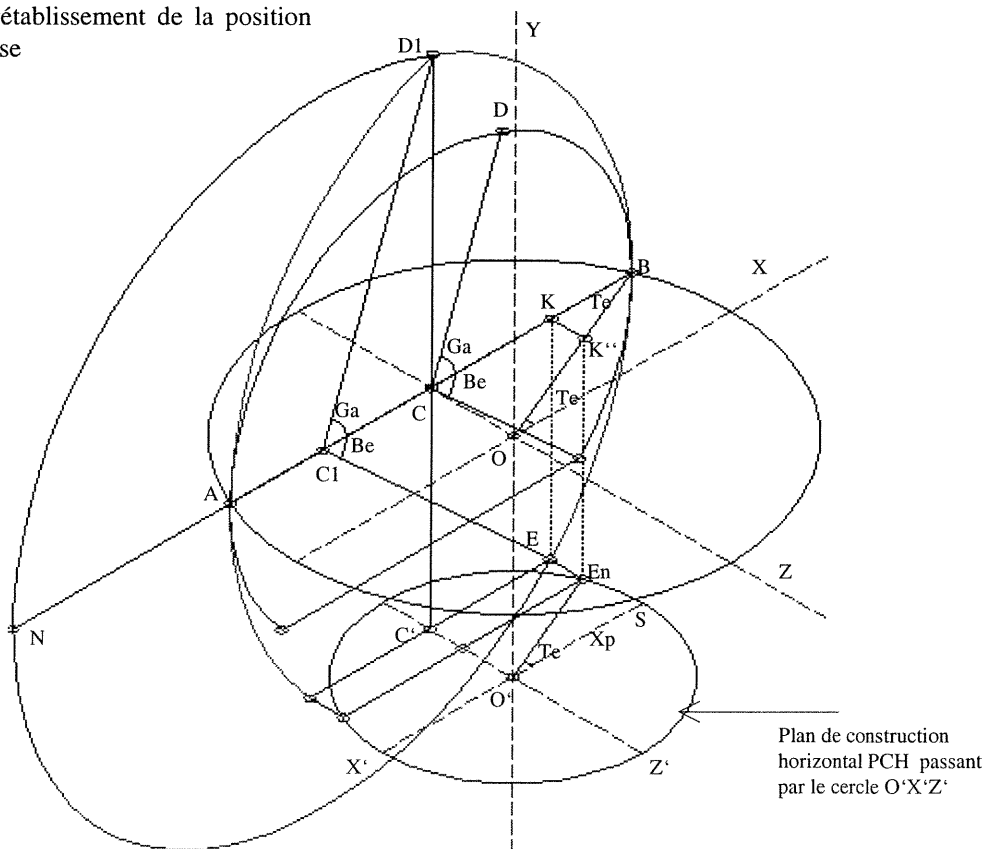
L'objectif de l'étude de l'étape-III est de calculer la variable (Zmove).

La variable Zmove représente la relation numérique entre la localisation du plan de construction horizontal PCH et la position de l'arc de base par rapport à l'axe X. Chaque fois que la distance entre le plan PCH et le plan de référence XOZ change, l'arc de base prend une nouvelle position en se déplaçant le long de l'axe Z entre les deux points O et C. Ce déplacement est représenté par la distance EEn ou par le symbole (Zmove). Le paramètre responsable de ce cette opération est l'angle Be (figure-45).

Le principe de cette analyse (voir la figure-45)

Nous avons déjà mentionné que l'angle Be du triangle C₁EK contrôle la localisation du plan de construction horizontal PCH par rapport au plan de référence XOZ. Le plan PCH est représenté par le plan ou le cercle X'O'Z' dont le centre est O'. Le triangle C₁EK est dans le plan AD₁B parallèle au plan de référence XOY. Dans le triangle droit C₁EK, l'angle KC₁E est représenté par l'angle Be, KE représente la distance verticale qui sépare les deux plans XOZ et PCH. Dans le cas où l'angle Be n'est pas égale à zéro, mais tire sa valeur de l'intervalle (-Ga < Be < Ga), l'arc de base subit une translation sur l'axe Z; autrement dit, le plan PCH change de position en se déplaçant sur l'axe Y selon l'angle Be et, dans le même temps, l'arc de base se déplace sur l'axe Z afin d'être dans la bonne position, lors de la génération du dôme entrelacé.

Figure-45. Illustration en 3D montrant le principe du rétablissement de la position de l'arc de base



Équations et relations numériques

Calcul (EEn) du déplacement de l'arc de base (figure-46):

$EEn = SE - Sen$; on a deux distances à calculer SE et Sen

1- Calcul de la distance (SE)

Dans triangle droit OCB, on a l'angle CBO égal à l'angle Te , ($Te = \pi/Na$, voir le paragraphe 2.2.B).

L'angle Te et l'angle OBC sont égaux puisqu'ils sont des angles alternes-internes. Dans le triangle OBC, on a:

$$\tan Te = OC/OB \rightarrow OC = OB * \tan Te$$

$CO = Dc$, où Dc est la distance entre le centre du dôme O et l'arc de base, $O, B = Rc$, Rc étant le rayon du dôme. On substitue à OB et CO leurs valeurs et l'on obtient la relation:

$$Dc = Rc * \tan Te \quad (1)$$

2-Calcul de la distance (SEn)

$$\text{Du triangle } O'SEn \text{ on a } Sen = O'S * \tan Te \quad (2)$$

On va calculer en premier O'S. Du triangle droit C_1EK , on a l'angle $KC_1E = Be$

$$\cos Be = C_1K/C_1E \rightarrow C_1K = C_1E * \cos Be \quad (3)$$

$C_1E = Rex_n$ qui est une variable déjà calculée. Désignons C_1K par X, la relation (3) devient $X = Rex * \cos(Be)$ (4)

$CK = C_1K - C_1C$ où $C_1C = E2c$; celle-ci est une variable déjà calculée.

$CK = X - E2c$; désignons CK par Xp

$$Xp = X - E2c \quad (5)$$

$$CK = C'E = O'S = Xp$$

Dans l'équation (2), remplaçons O'S par son symbole Xp

$$Sen = Xp * \tan Te$$

SEn prend comme symbole Zp

$$Zp = Xp * \tan Te \quad (6)$$

À partir des équations (1) et (6), nous calculons $EEn = ES - EnS$

EEn prend comme symbole ($Zmove$)

$$Zmove = Dc - Zp \quad (7)$$

Donc la relation qui calcule le déplacement de l'arc de base est:

$$Z_{move} = D_c - Z_p \quad \text{III}$$

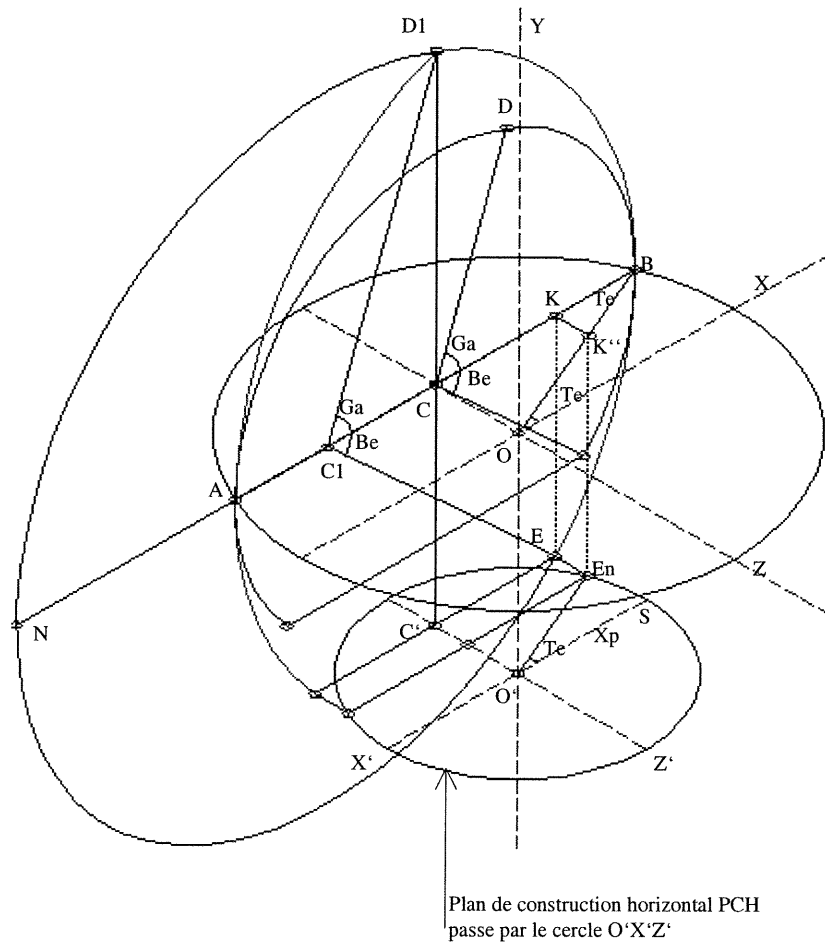
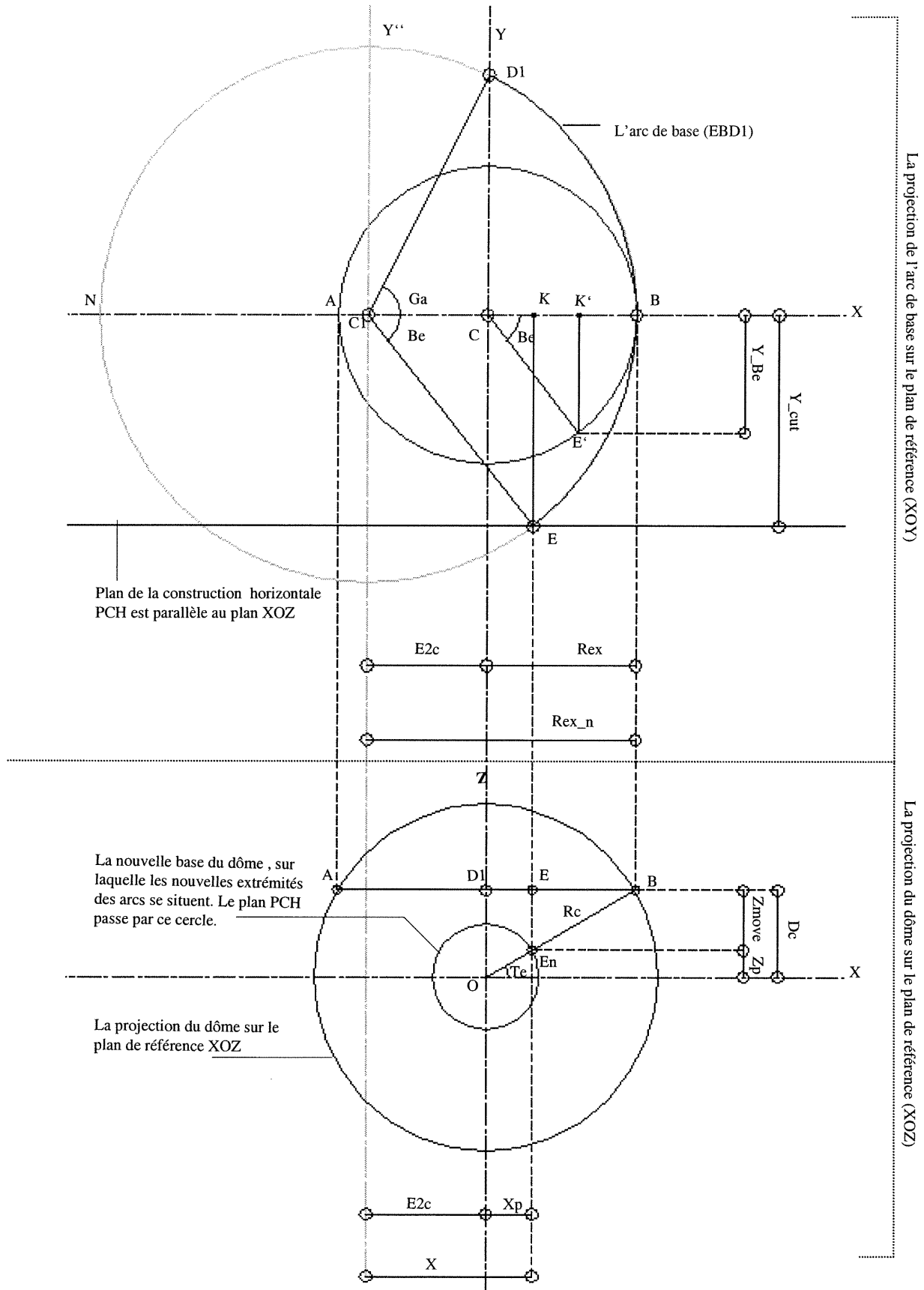


Figure-46. Illustration en 3D montrant le rétablissement de la position de l'arc de base

Figure-47- illustration 2D montrant le rétablissement de la géométrie de l'arc de base du dôme entrelacé



2.3.A.3.-Analyse géométrique des cas particuliers

Dans l'analyse géométrique précédente, on a considéré l'arc, d'un point de vue général sans caractériser cet objet ou sans regarder son style architectural. Pour que cet objet, c'est-à-dire l'arc, soit attaché à un style architectural particulier, nous choisirons deux styles d'arcs dont la forme géométrique est discontinue, à savoir l'arc gothique et l'arc islamique. Ces deux types d'arcs sont le résultat de l'intersection de deux arcs de cercle. Leur forme correspond à la structure géométrique d'un arc pointu dans le cas général de la génération de ce type d'arc.

2.3.A.3.I-Analyse géométrique de la règle gothique

Selon (Tiranti. A, 1957, p.30), la règle géométrique d'un arc gothique vient de la rotation de $\pi/4$ d'un carré autour de son centre. Soit ABEF, un carré de centre O, la rotation de ce carré autour de son centre O d'un angle de $\pi/4$, génère un autre carré A'B'E'F'. L'intersection de deux carrés produit un octogone dont les sommets sont 1,2,3,4,5,6,7 et 8. Le côté AB du carré ABEF représente la portée de l'arc gothique. C₁ et C₂ sont deux sommets consécutifs de l'octogone; ils représentent aussi les centres des deux cercles de rayons C₁B et C₂A. Les deux cercles se croisent au point D qui représente le sommet de l'arc gothique (figure-48). Le rayon du cercle C₁B est délimité par l'angle BOC₁ dont la valeur est $3\pi/8$, cet angle est la somme de deux angles BOO' et OO'C₁, donc $(BOO'=\pi/4) + (O'OC_1=\pi/8) = (BOC_1=3\pi/8)$.

L'objectif de cette analyse est d'établir les données numériques nous permettant de générer un arc gothique dans un contexte numérique général. On parle ici du cas général de la génération d'un arc pointu.

L'objectif de l'analyse de la règle gothique

est de calculer l'angle BC₁D qui définit la courbe géométrique de la moitié d'un arc gothique. L'angle BC₁D est représenté par l'angle Ga (figure-48). Celui-ci est déjà connu comme un paramètre dans le cas général de la génération d'un arc pointu. Le calcul de cet angle permettra d'insérer l'arc gothique à même la génération d'un arc pointu.

Les données sont:

(Rc) est le rayon du dôme entrelacé.
(Rex) est la projection de la moitié de l'arc de base sur le plan de référence XOZ. (O'B=Rex).

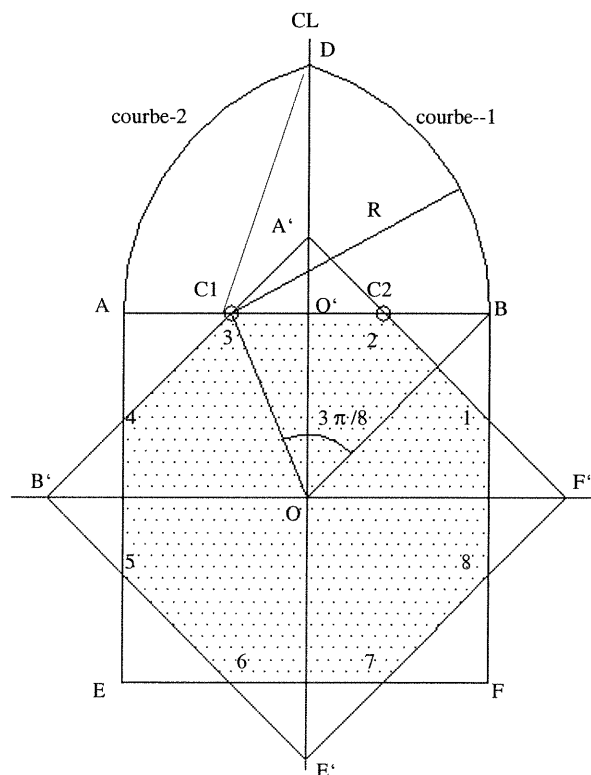


Figure-48. Données graphiques d'un arc gothique

Les données numériques de la règle d'un arc gothique

Calcul de l'angle «BC₁T = Ga»

(Ga) est l'angle qui définit la moitié de l'arc de base (figure-49).

C₁C₂ est le côté de l'octogone, il représente la distance entre les deux centres C₁ et C₂ des deux arcs du cercle qui forment l'arc gothique.

L'angle O'OC₁=π/8

OO' = O'B = Rex où Rex est la projection de l'arc BD sur l'axe XX'.

Tan O'OC₁ = C₁O' / OO' on substitue l'angle O'OC₁ par sa valeur

$$\tan \pi/8 = C_1O' / Rex \rightarrow C_1O' = Rex * \tan (\pi/8) \quad (1)$$

Calcul du rayon Rex-n de l'arc selon la règle gothique:

$$C_1B = BO' + C_1O' \quad \ll C_1B = C_1D = Rex_n \quad \text{et} \quad BO' = Rex \gg Rex_n = Rex + C_1O'$$

On substitue à C₁O' sa valeur de l'équation (1)

$$Rex_n = Rex + (Rex * \tan (\pi/8))$$

$$Rex_n = Rex (1 + \tan (\pi/8)) \quad (2)$$

Dans le triangle O'C₁T

les deux côtés C₁O' et C₁D sont connus,

on peut donc calculer l'angle Ga

$$\cos Ga = C_1O' / Rex_n$$

on substitue à C₁O' et Rex_n la valeur de leurs équations (1) et (2)

$$\cos Ga = (Rex * \tan (\pi/8)) / (Rex (1 + \tan (\pi/8)))$$

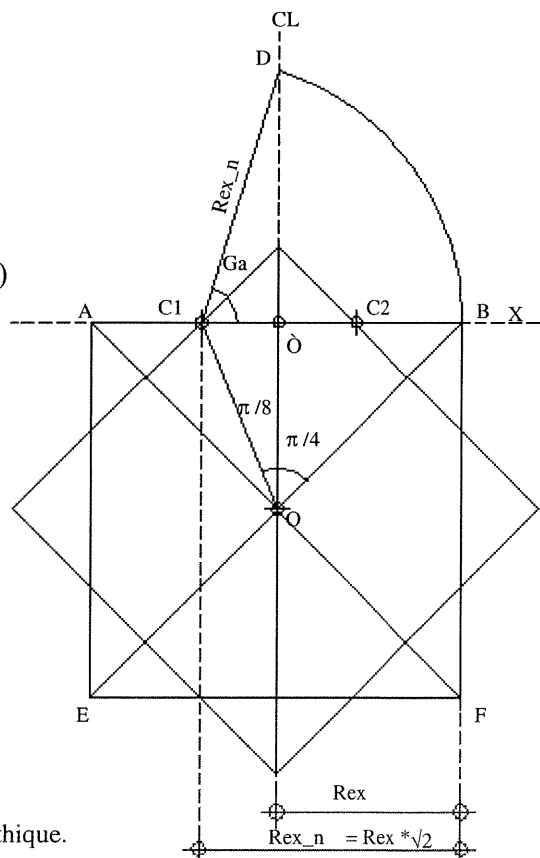
$$\cos Ga = (\tan (\pi/8)) / (1 + \tan (\pi/8)) \quad (3)$$

Équation IV

$$Ga = \arccos \left(\frac{\tan (\pi/8)}{1 + \tan (\pi/8)} \right)$$

Si on observe la valeur de l'angle Ga, on constate que Ga est un angle constant, cela signifie que la règle gothique n'implique qu'un changement d'échelle.

Figure-49. Illustration géométrique de la règle d'un arc gothique.



2.3.A.3.II-Analyse géométrique de la règle islamique

La règle islamique est basée sur la division de la base ou de la portée de l'arc par un nombre impair «n» (Paccard. A, 1983, p.345) et (Creswell, 1979, page-166). Les deux centres des courbes qui forment l'arc islamique sont les deux subdivisions de la portée les plus proches du plan de symétrie ZOY, qui est représenté par la ligne centrale OY.

Les données de la règle islamique sont:
(Rex) est la projection de la moitié de l'arc de base sur l'axe XX' .

(n) est le nombre de divisions, il est toujours impair.

(Du) est l'unité de division ou la distance entre deux divisions consécutives où $Du = (2 * Rex) / n$.

Les cas les plus répandus de la règle islamique impliquent une division de la portée en 3, 5 ou 7 (figure-50).

$$Du = 2 * Rex / 3$$

$$Du = 2 * Rex / 5$$

$$Du = 2 * Rex / 7$$

C_1 et C_2 sont les deux centres des deux courbes qui forment l'arc islamique. La distance entre ces deux centres est égale à une unité de division (Du).

L'objectif de l'analyse de la règle islamique est de calculer l'angle G_a qui définit la courbe géométrique de la moitié de l'arc islamique. L'angle « $KC_1D=G_a$ », l'angle G_a , ici, représente le lien entre le cas général de la génération d'un arc pointu et le cas particulier de la génération d'un arc islamique. Le calcul de cet angle permettra d'insérer la règle islamique dans le cas général de la génération d'un arc subissant des discontinuités surfaciques.

Les données et les paramètres sont :

(Rex): « $Rex=OB$ » est la projection de la moitié de l'arc de base sur l'axe XX' .

(n) est le nombre de divisions de la portée de l'arc islamique.

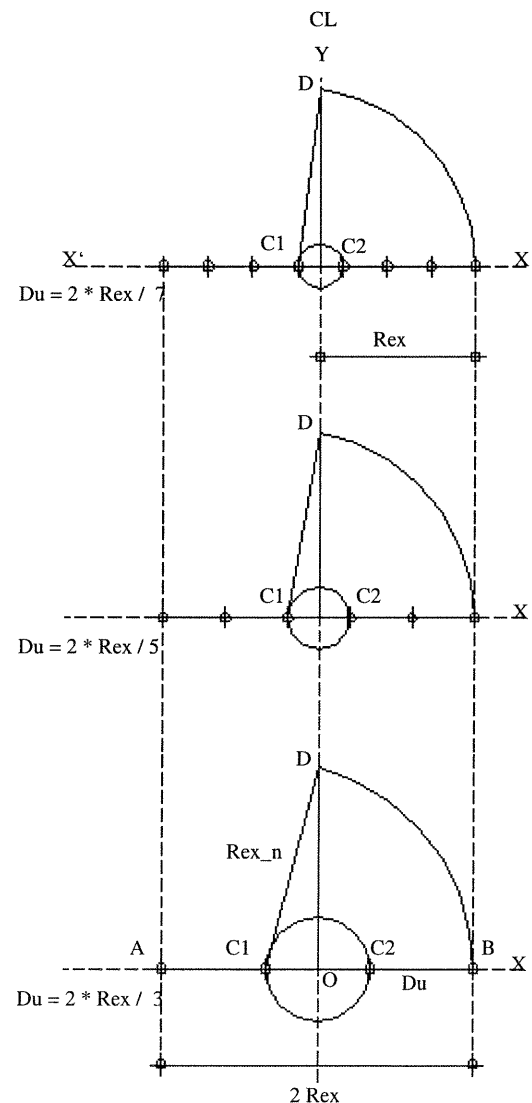


Figure-50. Les différentes divisions d'un arc islamique

Les données numériques de la règle d'un arc islamique:

Calcul du rayon R_{ex_n} : (figure-51)

(R_{ex_n}) est le rayon de la courbe BD selon la règle islamique.

(R_{ex}) est la demi-portée de l'arc islamique.

Pour un nombre de divisions (n), l'unité de division $Du=2 * R_{ex} / n$

On a $C_1B= C_1O + OB$ où

$C_1B=R_{ex_n}$, $C_1O=R_{ex} / n$ et $OB=R_{ex}$ d'où

$R_{ex_n}= (R_{ex} / n) + R_{ex}$

$R_{ex_n}=(R_{ex}+ (n * R_{ex})) / n$

$R_{ex_n}=(R_{ex} * (n + 1)) / n$

$$R_{ex_n}=(R_{ex} * (n + 1)) / n \quad (1)$$

Calcul de l'angle G_a :

Le triangle O,C_1,D a deux côtés connus:

$C_1O=R_{ex}/n$ et

$C_1D=R_{ex_n}$

$\cos G_a=C_1O / C_1D$

on remplace C_1O et C_1D par leurs valeurs

$\cos G_a=(R_{ex}/n) / R_{ex_n}$

On remplace R_{ex_n} par sa valeur contenue dans l'équation (1)

$\cos G_a=(R_{ex} / n) / ((R_{ex} * (n + 1)) / n)$

$\cos G_a=(R_{ex} / R_{ex}) * (1 / (n+1))$

$\cos G_a=1 / (n+1)$

$$G_a = \arccos (1 / (n + 1)) \quad \text{Équation-V}$$

Cette relation permet de générer un arc selon la règle islamique dans le cadre du cas général de la génération d'un arc pointu.

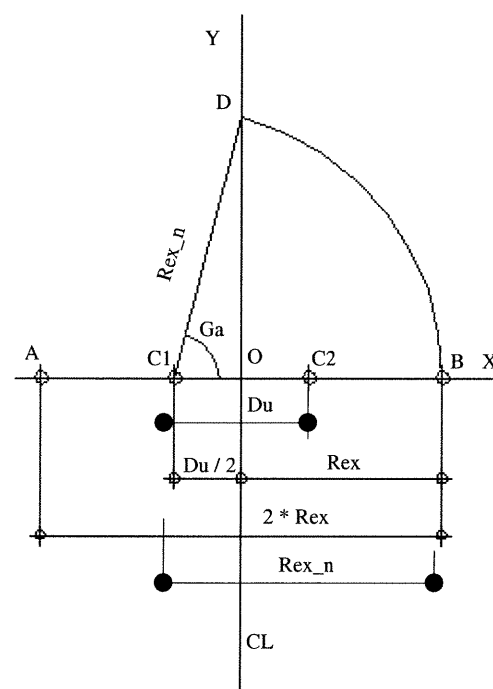


Figure-51. Illustration géométrique de la règle d'un arc islamique

2.3.B-Analyse géométrique (graphique) d'un dôme entrelacé dont l'arc de base a une continuité surfacique (elliptique):

Dans cette étude, nous cherchons à construire un arc continu dont la forme peut être un demi-cercle ou une demi-ellipse surhaussée ou surbaissée.

Au paragraphe précédent 2.3.A, le but de l'analyse géométrique était de relever les données numériques, les équations qui contrôlent la génération de l'arc de base, les relations géométriques et enfin d'adopter ces données comme solution pour générer les différents types d'arcs pointus.

Dans le cas d'un arc continu, l'analyse géométrique ou graphique est basé sur les données graphiques du tracé d'un arc. On établit un champ d'informations, de descriptions graphiques afin de résoudre le problème de la génération d'un arc continu. Dans ce genre d'analyse, on s'appuie sur la capacité du système informatique de générer ce type d'arc à partir de données graphiques. Souvent on utilise cette méthode d'analyse afin d'éviter un long calcul numérique et géométrique. L'idée de l'analyse graphique ou géométrique est basée sur l'utilisation de la géométrie analytique, où l'on se sert d'une courbe dont les propriétés géométriques sont connues. Il suffit de transférer ces propriétés au système informatique pour générer l'arc approprié.

En résumé, en ce qui concerne la géométrie des arcs pointus discontinus, on cherche la solution dans les relations numériques. Par contre, dans la géométrie des arcs continus, la solution choisie est de type géométrique ou graphique.

La méthode géométrique ou graphique de tracer un arc continu est simple et, de plus, elle est facile à transférer au système informatique. Mais, il n'y a pas beaucoup de logiciels DAO ou de CAO permettant de traiter des données à base de descriptions graphiques. Prenons, par exemple, le problème qui consiste à définir le point d'intersection de deux lignes non parallèles et situées dans le même plan. Or, il y a peu de logiciels DAO ou CAO, qui soient capables de traiter ce genre d'informations descriptives (les données graphiques) et de fournir un résultat numérique. Ce qui veut dire que le choix de la méthode graphique dépend du logiciel utilisé dans la modélisation. C'est pour cette raison que nous devons avoir une idée claire et précise sur le système informatique que nous allons utiliser pour la construction de l'arc, avant d'établir notre solution géométrique, graphique ou numérique.

Par exemple: la méthode de tracé d'un arc elliptique s'appuie sur une structure géométrique qui est constituée de lignes et de points représentant l'intersection de ces lignes. Dans ce cas, il faut être sûr que le système informatique adopté permet de diriger ce genre d'opération. On peut dire qu'il y a une relation directe entre le choix des outils informatiques et la méthode de construction d'arcs continus.

Méthode de tracé d'un arc elliptique

On peut appeler cette façon de tracer un arc «méthode appliquée ou modèle appliqué ». Le but de cette analyse est d'établir le champ d'informations comprenant des données graphiques et descriptives de la méthode de tracé d'un arc demi-ellipse. Ces données peuvent être transférées au système informatique qui, à son tour, génère l'arc de base.

La méthode descriptive de tracé d'un arc demi-ellipse:

La méthode la plus connue est de tracer deux cercles concentriques, l'un représente le grand axe et l'autre le petit axe de l'ellipse.

On trace un rayon quelconque du centre C soit CB (Figure-52-a).

Soit B le point d'intersection du rayon CB avec le cercle (I).

Soit A le point d'intersection du rayon CB avec le cercle (II).

On trace une ligne BE parallèle au petit axe PP, à partir du point B.

On trace une ligne AE parallèle au grand axe GG, à partir du point A.

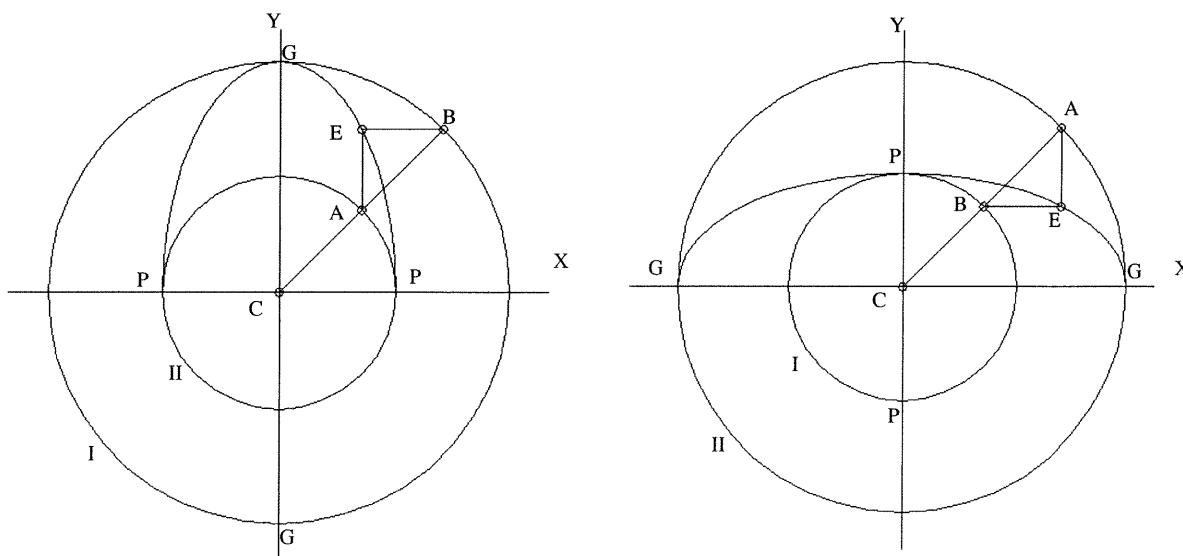
L'intersection des deux lignes est le point E, celui-ci fait partie de l'ellipse PGP. Selon la position des axes PP et GG de l'ellipse PGP, on peut tracer trois types d'arcs elliptiques.

1-Le petit axe PP est sur l'axe XX' , le grand axe GG est sur l'axe YY' et le rayon $CB > CA$. On obtient un arc elliptique surhaussé (figure-52-a)

2-Le petit axe PP est sur l'axe YY' , le grand axe GG est sur l'axe XX' et le rayon $CB < CA$. On obtient un arc elliptique surbaissé (figure-52-b)

3-Les axes PP et GG sont égaux, les points A, B et E sont superposés et le rayon CB est égal au rayon CA. L'arc elliptique devient un demi-cercle et les deux cercles I et II sont superposés.

Pour tracer un arc elliptique, il suffit de transmettre cette logique ou cette méthode de description du tracé d'un arc à un système informatique capable de manipuler ce genre d'informations.



(a-). Arc elliptique surhaussé

(b-). Arc elliptique surbaissé

Figure-52. La méthode de tracé d'un arc demi-ellipse

3-Étude contextuelle

Le but principal de cette étude est de dégager une stratégie ou une compréhension globale permettant de mettre en relation toutes les données de l'analyse d'un dôme entrelacé. C'est-à-dire établir un schéma des étapes du processus de génération d'un dôme entrelacé (figure-53).

L'analyse géométrique fournit toutes les variables, paramètres, équations ainsi que l'ensemble de la solution géométrique; ce qui a pour but de générer l'arc de base.

Les données constructives et la règle d'assemblage complètent la tâche de la génération du dôme en regroupant les arcs générés selon la règle géométrique adoptée.

Les résultats de l'analyse historique interviennent dans les dernières étapes de la modélisation et servent à caractériser l'arc de base et le dôme entrelacé généré. Cette caractérisation permet de produire un dôme entrelacé, qu'il soit islamique, gothique ou appartenant à d'autres styles architecturaux.

La figure-53 montre le schéma des relations entre les trois étapes de la génération d'un dôme entrelacé. Ces relations étant séquentielles nous pouvons donc obtenir un résultat à partir de l'étape-1 ou l'étape 1 + 2 ou l'étape 1 + 2 + 3.

Dans le cadre de l'étude contextuelle, nous définirons les paramètres du programme, établirons le modèle logique et enfin installerons le modèle mathématique (numérique).

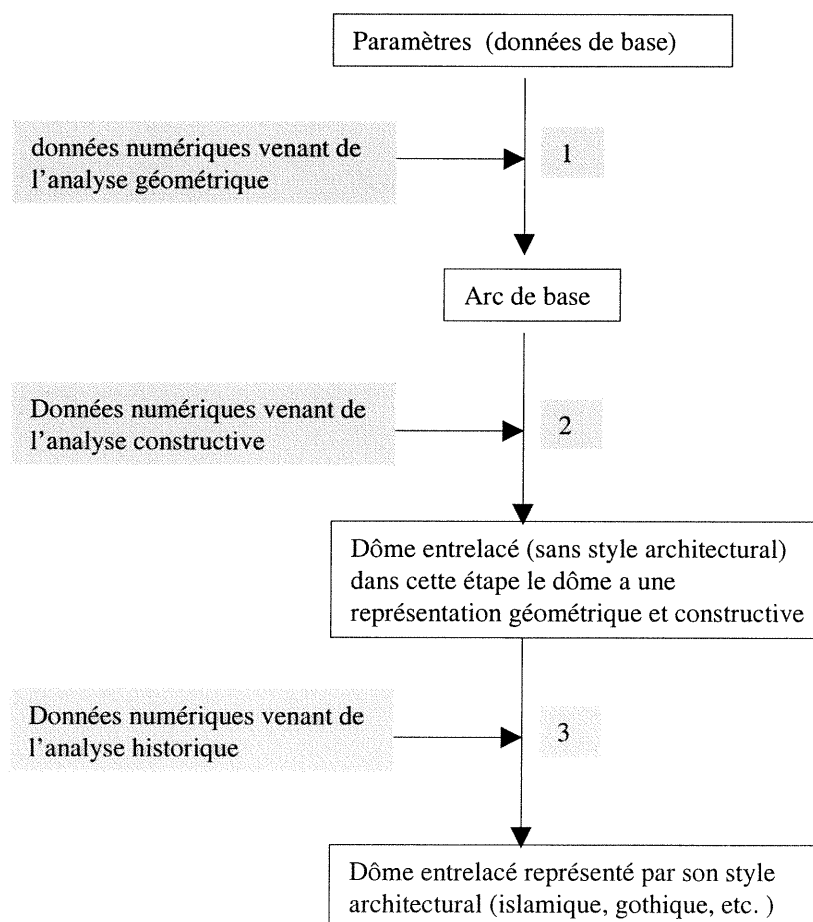


Figure-53. Schéma d'étude contextuelle

3.1-Les paramètres et les données

L'analyse géométrique et constructive nous a permis de découvrir les clés principales des données numériques que sont les paramètres (figure-54).

Un paramètre est une variable qui influence d'autres valeurs dans une solution géométrique et qui peut prendre une série de valeurs.

Les principaux paramètres sont :

1-(Rc) est le rayon de dôme entrelacé.

2-(Na) est le nombre d'arcs à générer dans le dôme entrelacé. Le paramètre Na donne à l'angle (Te) sa valeur ($Te = \pi/Na$), celui-ci représente l'angle de position de l'arc de base par rapport à l'axe XX' (voir paragraphe 2.2.B).

2-(Ga) ou (Gama) est l'angle qui définit la construction de la moitié de l'arc de base. L'angle Ga existe dans un plan qui passe par la droite AB et est parallèle au plan XOY.

3-(Be) est l'angle qui contrôle le déplacement du plan de construction horizontal de l'arc de base. Les angles Be et Ga existent dans le plan de construction de l'arc de base (voir paragraphe 2.3.A.2.II) .

4-(Nis) est le paramètre d'initialisation de la règle islamique.

5-(n) est le nombre de divisions de l'arc islamique

6-(Ngo) est le paramètre d'initialisation de la règle gothique .

Les deux paramètres Nis et Ngo prennent la valeur 1 ou 0. Leur rôle est de filtrer les informations concernant les règles géométriques islamiques, gothiques ou d'autres règles pouvant être intégrées au choix du style architectural de l'arc de base.

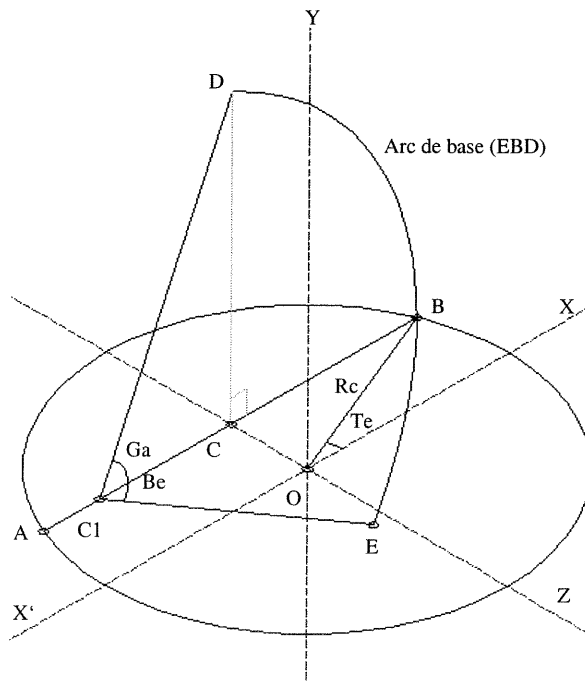


Figure-54. Les paramètres de la construction de l'arc de base et le dôme entrelacé (Rc, Na, Ga, Be, Nis, n, Ngo)

3.2-Établissement un modèle logique

Un modèle logique résume le lien existant entre les différents types d'informations: géométriques, constructifs et historiques. Toutes ces informations peuvent être représentées par des données numériques. En fait, le modèle logique est la manière d'organiser la succession des décisions et des actions pour obtenir le résultat souhaité. Il faut faire la différence entre les informations et/ou les relations qui sont essentielles dans notre structure logique. Elles sont indispensables à la formation du noyau principal du programme. D'autres informations secondaires élargissent les tâches et les complètent. Par exemple: les données du bloc (A) (figure-55) sont complémentaires par rapport au reste du programme. Il y a aussi des informations numériques qui ne sont pas acceptées dans le cadre de la structure algorithmique; dans ce cas, le programme provoque une erreur lorsque les paramètres sont analysés. Si, par exemple, un paramètre n'accepte que les valeurs 1 ou 0, et que l'on donne à ce paramètre la valeur 2, le système provoque une erreur; Cela signifie que les valeurs numériques ne correspondent pas aux données acceptées par le programme.

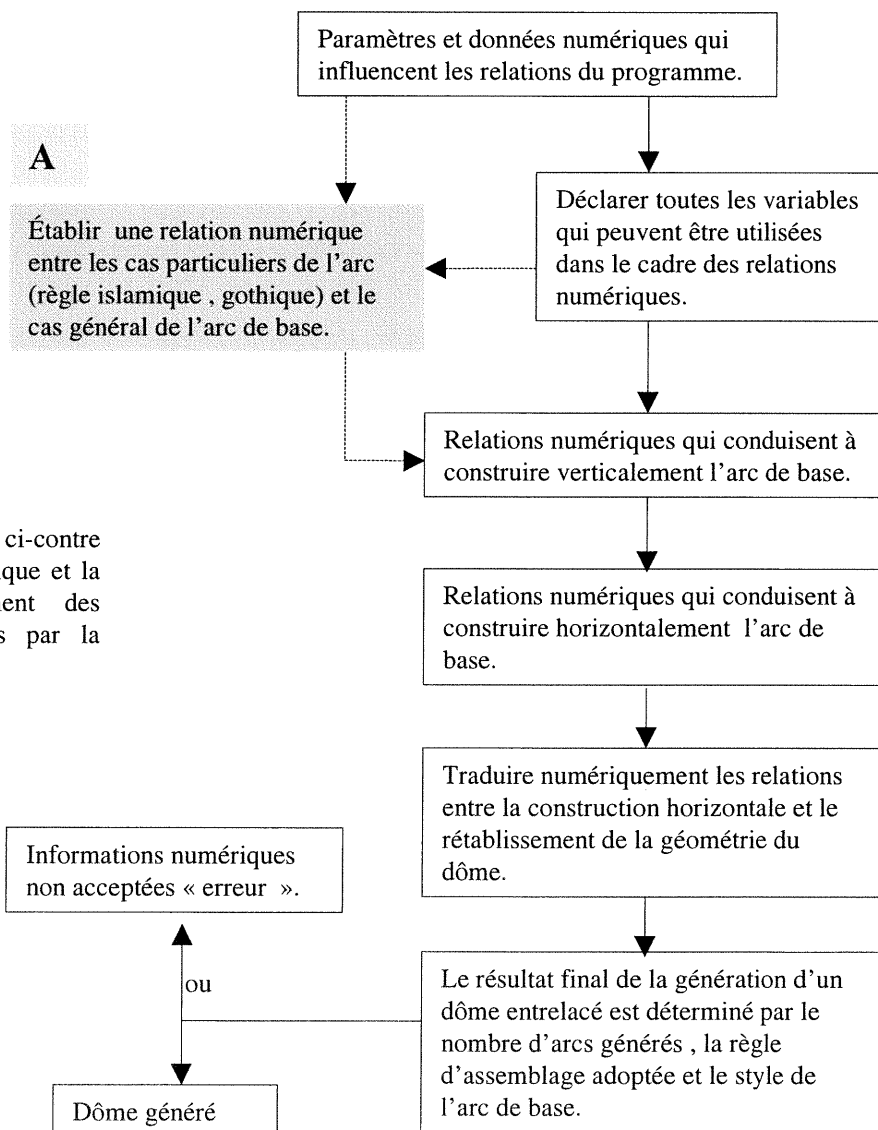


Figure-55. Le schéma ci-contre montre le modèle logique et la manière du traitement des informations obtenues par la partie analytique.

3.3-Établissement un modèle mathématique (numérique)

un modèle numérique est un procédé organisé dans un contexte logique visant à fournir une solution dimensionnelle dans un cadre formel et scientifique.

Le but d'un modèle numérique est de mettre en relation toutes les données numériques qui sont le résultat des analyses précédentes: analyse géométrique, analyse de la règle d'assemblage et analyse historique. À partir du schéma ci-dessous, on remarque que les données ou les paramètres peuvent être utilisés dans le calcul des variables et filtrés par des expressions conditionnelles et même entrer directement dans le corps du programme. Ainsi les paramètres ont un accès direct ou indirect au corps de la structure algorithmique.

Le résultat retenu peut prendre différents aspects. On peut avoir un objet généré, un message d'erreurs ou d'informations numériques réintégré de nouveau dans le calcul (figure-56). Il est important que le résultat retenu par le programme soit clair et compréhensible. Par exemple, un message d'erreur retenu par le programme doit en indiquer la cause et guider l'utilisateur vers les corrections possibles.

Enfin, on traduit le schéma ci-dessous en modèle numérique formant la principale base numérique du futur programme, voir la figure-57, page-63.

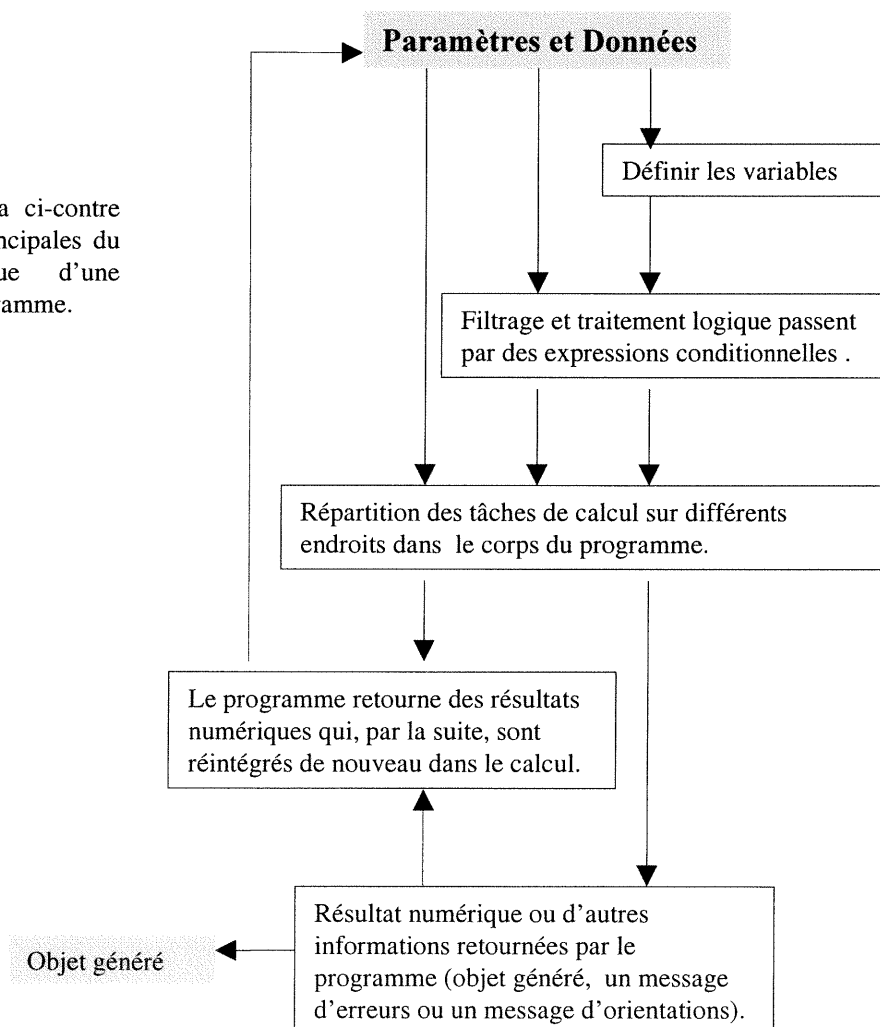


Figure-56. Le schéma ci-contre montre les étapes principales du traitement numérique d'une fonction ou d'un programme.

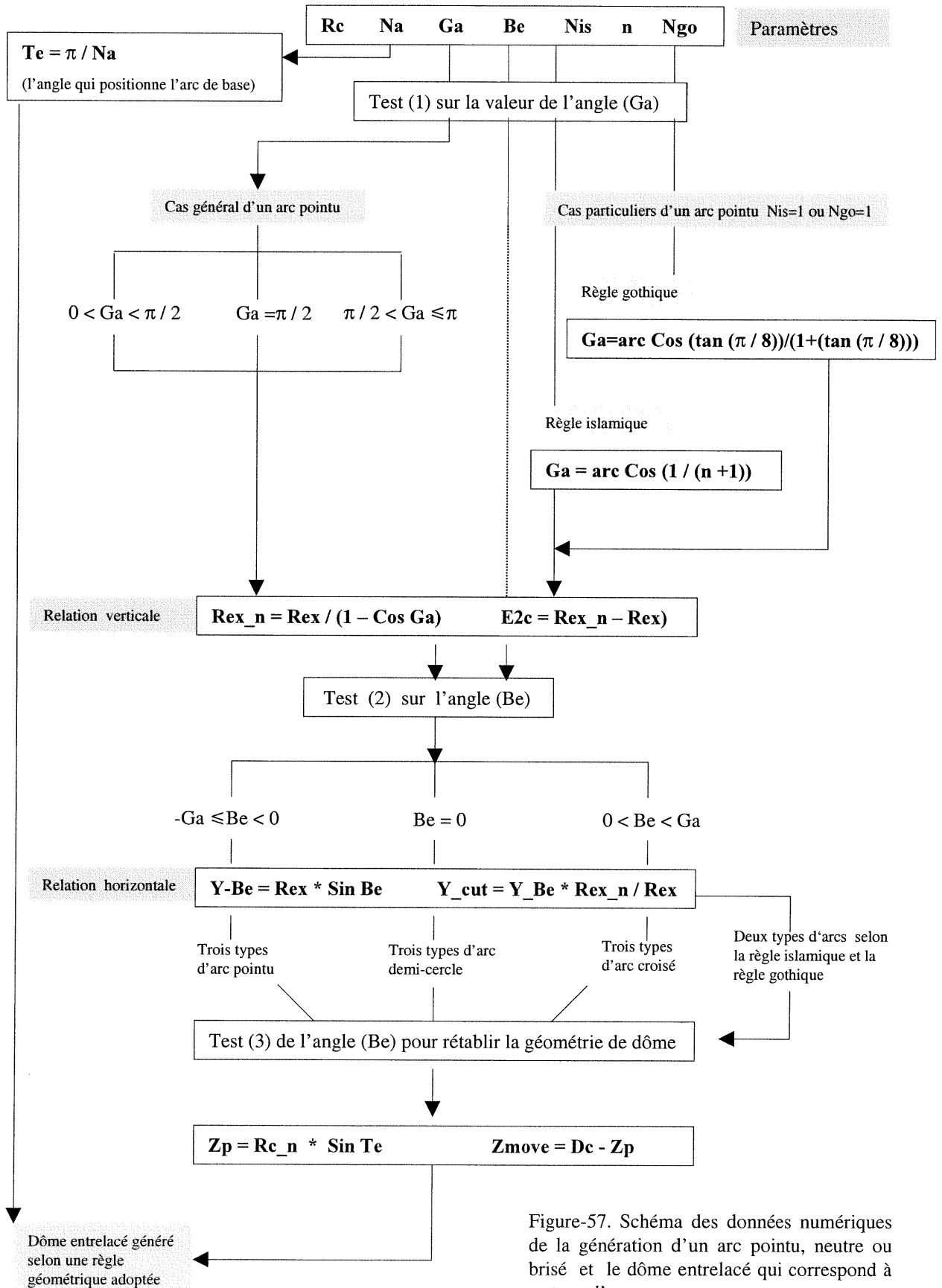


Figure-57. Schéma des données numériques de la génération d'un arc pointu, neutre ou brisé et le dôme entrelacé qui correspond à ce type d'arc .

Conclusion du chapitre-II

Les dômes sont les éléments les plus sophistiqués en architecture. Leur création a conduit une nouvelle compréhension de la délimitation de l'espace. La nouvelle technologie et le développement des théories du calcul des dômes ont permis de résoudre plusieurs problèmes d'organisation de l'espace ainsi que la création de nouvelles formes architecturales. L'importance architecturale des dômes et des arcs nous a encouragé à les adopter comme champ d'étude et/ou domaine d'application de cette recherche. En fait, la concrétisation de la recherche implique l'évaluation des principaux types de dômes qui, à leur tour, seront soumis à des critères géométriques, historiques et structuraux; ces derniers nous Permettront de préciser notre choix et d'adopter les dômes entrelacés comme cas d'étude particulier d'étude à modéliser.

L'utilisation de l'informatique, à des fins de modélisation des processus de construction et/ou modélisation des dômes entrelacés et de génération de nouvelles formes géométriques appropriées à ce type de dômes, exige une grand nombre opérations d'analyse. Ces analyses nous permettront de découvrir tous les aspects historiques, structuraux et géométriques des dômes entrelacés et de définir leurs données mathématiques dans le cadre d'un modèle numérique.

Sur le plan historique, nous avons fait ressortir les liens qui existent entre les différentes expériences de constructions des dômes. En fait les dômes entrelacés ne sont pas isolés des démarches de développements architecturaux et géométriques par rapport à d'autres types de dômes. Comme nous l'avons vu, l'expérience de la construction des voûtes romaines et byzantines a influencé la création du système entrelacé. Nous savons aussi que les trois principaux systèmes de constructions des dômes (entrelacé, pendentif et d'arête) ont été développés afin de résoudre le problème qui est de couvrir un plan carré ou polygonal.

À la différence d'autres systèmes de construction de dômes, le système entrelacé se distingue par une règle de construction, basée sur l'idée de l'assemblage d'arcs regroupant les sommets du polygone de base du dôme. La base du dôme peut être, par exemple, un hexagone, un pentagone, un décagone, etc.). Cette caractéristique a assuré une grande flexibilité lors de la génération des dômes. On peut avoir une large variété de dômes entrelacés générés selon leurs bases polygonales et selon le mode d'assemblage de leurs arcs de base.

La forme géométrique de l'arc de base des dômes entrelacés joue un rôle essentiel dans la définition de son style, par exemple, des dômes entrelacés pointus, ellipsoïdaux, gothiques, islamiques, etc. La géométrie nous permet de caractériser ce type de dômes à partir de deux catégories de surfaces: la catégorie dite des surfaces discontinues et catégorie dite des surfaces continues. La première est analysée de façon mathématique et la deuxième de façon graphique. Cette subdivision établit une base de données numériques (modèle numérique) et une autre graphiques (modèle graphique appliqué), permettant de communiquer les formes géométriques de dômes et de les préparer pour la prochaine phase de modélisation, soit la représentation informatique.

CHAPITRE - III

CHOIX DES OUTILS DE TRAVAIL ET STRUCTURE INFORMATIQUE

CHAPITRE - III

CHOIX DES OUTILS DE TRAVAIL

STRUCTURE INFORMATIQUE

1-Le choix des outils de travail

1.1-Critères de choix et motivations

Selon l'objectif de cette recherche, la modélisation, la simulation et la génération des nouvelles formes de dômes entrelacés demeurent notre principale préoccupation. Le choix des outils de travail peut être basé sur cet objectif. Cette étude vise à choisir le système informatique le plus performant capable de nous aidant à réaliser cette recherche. Nous allons préciser tout d'abord les besoins et les tâches que le système doit satisfaire. Les analyses historiques, constructives et géométriques étant réalisées dans le cadre du chapitre-II, nous guideront dans le choix des outils les mieux adaptés et les plus convenables face au traitement des données (géométriques, numériques et graphiques), à la modélisation des dômes entrelacés et enfin à la génération de nouvelles formes d'arcs de base et des dômes entrelacés. Cinq critères peuvent être annoncés:

Premier critère: le système accepte le traitement des données numériques:

Dans le cadre du chapitre-II paragraphe 2.3.A, on remarque que le résultat de l'analyse géométrique d'un dôme entrelacé dont l'arc de base subit des discontinuités surfaciques, est représenté par des équations et des relations numériques. C'est-à-dire que les données de base sont numériques. La tâche principale du système informatique choisi sera d'accepter ce genre d'informations numériques, de les manipuler et de retourner le résultat de cette manipulation.

Deuxième critère: le système accepte le traitement des données graphiques:

Dans le même chapitre-II, le paragraphe 2.3.B montre un autre type d'analyse: celui de l'analyse géométrique ou graphique d'un dôme entrelacé dont l'arc de base a une continuité surfacique. Le résultat de cette analyse est représenté par la méthode graphique du tracé de l'arc de base. On a appelé cette méthode «modèle appliqué». Dans cette sorte d'analyse les données numériques ne jouent pas un rôle majeur. En effet, on utilise la méthode descriptive pour représenter graphiquement l'arc de base. Le rôle du système informatique choisi est de traduire les données graphiques en représentations géométriques. Par cette méthode on décrit les étapes conduisant à la création d'un objet géométrique. L'ensemble de ces démarches est représenté par une méthode impliquant la logique et les processus de construction de l'objet.

Par exemple: la figure-58 montre la méthode graphique pour tracer un arc elliptique. «On demande au système informatique de définir l'intersection du rayon CB et des deux cercles dont les diamètres représentent le petit et le grand axe de l'ellipse, et par la suite, de retourner le point de l'intersection de la ligne horizontale qui passe par le point B et de la ligne verticale qui passe par le point A». En transmettant cette description avec les coordonnées des points C et B au système informatique, celui-ci retournera ou définira le point E qui est un des points de l'ellipse .

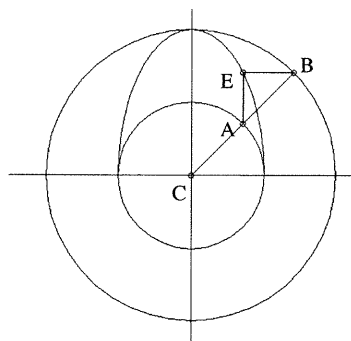


Figure-58. Méthode de tracé d'une ellipse

Troisième critère: le système basé sur la programmation:

Lorsqu'on modélise un objet le but n'est pas uniquement d'obtenir une image ou de réaliser une simulation, mais plutôt de décrire et de transmettre l'intelligence de cet objet (le développement historique de sa forme géométrique, son style architectural, sa règle constructive et/ou d'assemblage et sa référence géométrique). Il faut que cette intelligence soit organisée à l'aide d'un langage informatique, lisible par le système informatique adopté. Il faut que ce dernier soit un des moyens du transfert du « savoir-faire » et d'assurer une mise à jour facile de l'objet.

Quatrième critère: la base géométrique du système:

La génération des nouvelles formes s'appuie sur la capacité des outils informatiques utilisés. Cette capacité dépend du langage de la programmation et du niveau de la géométrie sur laquelle le système est basé. Il est important que le système géométrique des outils informatiques choisis permette une transformation géométrique entre les différents types de surfaces (parabolique, ellipsoïdale et hyperboloïde). Par exemple: un cylindre se transforme en ellipsoïde ou hyperboloïde. En architecture, une colonne peut prendre une forme cylindrique, ellipsoïdale tronquée ou hyperbolique. Ces possibilités réduisent énormément le nombre de primitives, utilisés dans la modélisation et, de ce fait, simplifient la structure du programme.

Cinquième critère: la manipulation algorithmique:

Lorsqu'on modélise un objet compliqué, plusieurs fonctions ou algorithmes doivent pouvoir être définis. Le système informatique assure les relations interactives entre les différentes fonctions utilisées dans le programme. L'appel conditionnel à des fonctions et/ou à des algorithmes périphériques est une préoccupation majeure dans les tâches de programmation. Pour créer et de gérer un réseau de fonctions, nous avons besoin d'un système informatique et d'un langage de programmation suffisamment développés permettant le contrôle du programme.

L'objectif de cette étude n'a pas de comparer les systèmes informatiques mais plutôt d'annoncer des critères de choix d'un système répondant le mieux à nos besoins de modélisation. On peut donc résumer les critères du choix du système informatique par les points suivants:

- 1-Le système informatique doit être en mesure de manipuler des relations, des équations et des données numériques et géométriques.
- 2-Le système doit pouvoir traduire la représentation descriptive d'un objet en données numériques et/ou géométriques.
- 3-Le système utilise la programmation comme moyen principal pour modéliser un objet. C'est le seul moyen qui permet de décrire et de transmettre les processus de la construction d'un objet.
- 4-Le langage de la programmation doit permettre de manipuler des données numériques et/ou géométriques et de donner des instructions à l'ordinateur dont la fin est d'organiser ces manipulations.
- 5-Le système peut gérer les fonctions et les algorithmes d'une manière interactive et assurer la fiabilité du partage des tâches entre les différentes parties du programme.

1.2-La base d'outils informatique (système géométrique et langage informatique)

Les cinq critères sus-mentionnés sont notre principal guide pour choisir le système informatique adéquat. On a déjà rappelé, (paragraphe-5, chapitre-I), que les processus de modélisation d'un objet peuvent être réalisés à l'aide d'outils informatiques basées, d'un part, sur un système géométrique et d'autre part sur un langage de programmation.

Un système géométrique: selon (T. Tidafi, 1996, p.255) «le système géométrique est un ensemble d'équations mathématiques permettant de définir et de transformer une forme géométrique». La différence entre un système géométrique et un autre réside dans le niveau de géométrie sur laquelle chacun de ces systèmes s'appuie et dans les concepts mathématiques considérés pour définir leurs primitives géométriques. Cela veut dire qu'un système géométrique doit être évalué à partir de ses concepts mathématiques et du niveau de sa géométrie. Les deux systèmes les plus répandus dans le monde des logiciels DAO et CAO sont le système CSG (*Constructive Solid Geometry*) et système B-Rep (*Boundary Representation*).

Le système CSG: la construction d'un objet à l'aide du système CSG s'appuie sur l'utilisation des primitives élémentaires de la géométrie euclidienne, qui peuvent être assemblées au moyen d'opérateurs booléens comme l'union, l'intersection ou la différence. Les résultantes de ces constructions sont organisées en arbre où les nœuds de l'arbre correspondent à des opérateurs et où le bout des ramifications comportent les primitives géométriques considérés (T. Tidafi, 1996). La représentation mathématique par des équations implicites des primitives élémentaires du système CSG ne permet aucune déformation. Le système CSG définit l'objet d'une manière constructive à partir de primitives géométriques et d'opérations de constructions. Selon (J. F. Rotgé, 1997) le modèle mathématique n'est nécessairement pas le reflet du processus de construction de l'objet. En fait, le système CSG est un système rigide et limité par des primitives volumiques prédéfinis; ce qui ne convient ni à la déformation géométrique de la forme de l'objet ni à la génération de nouvelles formes, à partir d'une forme de base.

Le système B-Rep est un système surfacique basé sur la définition d'une forme à partir de ses frontières, c'est-à-dire ses sommets, arêtes et surfaces. Ce système est plus flexible que le système CSG, parce qu'il permet la déformation surfacique de l'objet modélisé et les transformations géométriques, (telles que rotation, translation et déplacement), de ses éléments topologiques, à savoir sommets, arêtes et surfaces). La géométrie du système B-Rep repose essentiellement sur le principe de la géométrie topologique. Selon (Fleury. M, 1986, p.14) la topologie est une branche des mathématiques qui a généralisé la notion de déformation élastique d'une surface. Ce qui permet d'appliquer une transformation géométrique à un objet sans en modifier les propriétés topologiques. Ceci explique la base géométrique du système B-Rep qui est inspirée d'un très haut niveau géométrique, c'est la géométrie topologique. Mais le problème demeure dans le concept mathématique de ses formes primitives étant basés sur la géométrie euclidienne. Ce qui ne permet pas une transformation ou une déformation géométrique des différents types surfaciques (parabolique, ellipsoïdale et hyperboloïde).

En résumé, les deux systèmes précédents CSG et B-Rep ne répondent pas exactement au problème de modélisation posé au début de la recherche, à savoir la modélisation d'un objet dans un environnement dynamique, la génération des nouvelles formes géométriques ayant différents types surfaciques, et l'insertion de ces dernières dans un autre contexte architectural. En effet les deux systèmes CSG et B-Rep utilisent un concept mathématique et une géométrie euclidienne dans la construction de leurs primitives de base. Ce concept ne permet pas le passage d'un type surfacique à un autre. Par conséquent la solution se trouve dans des systèmes dont la base géométrique permet des déformations et des transformations géométriques entre les différents types de surfaces et de généraliser le concept mathématique géométrique de primitives de base. Le système SGDL (*Solid Geometry design Logic*) fait partie de ces descriptions informatiques. Ce système, développé par J. F. Rotgé, propose un cadre géométrique issu de la géométrie projective et rend possible une représentation unique pour l'ensemble des primitives du système SGDL.

Un langage de programmation est un moyen ou un outil qui permet de traduire des informations en code informatique et de donner des instructions à l'ordinateur afin de les traiter. Dans le cadre de cette étude, nous nous proposons d'établir un langage de programmation capable d'exprimer et de réaliser des processus ou des actions (T. Tidafi, 1996, p.255). Notre objectif est d'établir un programme ou une structure informatique pouvant générer une forme géométrique dynamique, c'est-à-dire une forme qui s'évalue dans une unité espace-temps. L'utilisation d'un langage informatique dans la modélisation du processus de construction d'un objet permet de contrôler l'évolution de l'objet modélisé, d'analyser les résultats d'une manière précise et de faciliter la mise à jour des informations du programme.

Plusieurs types de langages de programmation sont envisagés. Les langages impératifs^[1] transmettent une liste d'instructions à une machine qui doit les exécuter dans un ordre bien défini. Les langages orientés objets^[2] visent la manipulation d'objets informatiques ainsi que leurs caractéristiques. Les langages fonctionnels ont été mis au point pour permettre à l'ordinateur de traiter des méthodes afin de résoudre des problèmes (T. Tidafi, 1996, p.256).

Il est clair que les langages fonctionnels répondent mieux à notre préoccupation la modélisation dynamique du processus de construction de l'objet (dômes entrelacés). Ces langages peuvent à la fois traiter des problèmes combinatoires et décrire des processus et/ou des actions. La programmation fonctionnelle se caractérise par une sémantique exceptionnellement claire et simple. Sa syntaxe est basée sur un très petit nombre de règles, sans restriction sur la manière de former ou de composer des expressions. Elle est mise en application grâce au langage *Scheme*, (*Scheme* est un dialecte du Lisp). L'importance du langage *Scheme* tient à sa capacité d'effectuer des calculs aussi bien sur les valeurs que sur les fonctions, grâce à une formulation homogène, appelée lambda-calcul.

En résumé, l'adoption de SGDL en tant que système géométrique, et de *Scheme*, en tant que langage de programmation, nous permettra d'atteindre l'objectif recherché, qui est de constituer un programme, capable de modéliser l'objet et de créer de nouvelles formes. Dans ce cas le système SGDL parvient à définir les formes géométriques des éléments de l'objet à modéliser, et le langage *Scheme* à coder ces formes géométriques et à donner à l'ordinateur les instructions nécessaires pour les traiter.

Selon j. F. Rotgé la programmation volumique d'une forme est basée sur trois schémas de programmation: la programmation arborescente, la programmation itérative et récursive. Dans la programmation arborescente un volume est défini par une arborescence spécifiant le mode d'assemblage des volumes élémentaires, tandis que la programmation itérative et récursive décrit des mécanisme répétitifs. La récursivité est un modèle de calcul qui permet à une fonction de faire appel à sa propre définition, pendant qu'il produit un résultat. Le système SGDL utilise la syntaxe du langage *Scheme*, ses caractéristiques de programmation (l'arborescente, la récursivité) et des fonctions qui, ajoutées à celles de SGDL permettront de décrire dynamiquement la façon de construire l'objet 3D, plutôt que de spécifier statiquement des coordonnées ou des instructions séquentielles.

[1]-Les langages impératifs: par exemples C, BASIC, Pascal et FORTRAN, selon (P. Aitke, 1995; P.650) sont des langages impératifs ou procéduraux. Dans ces langages le programme est entièrement construit autour des codes (fonctions) et des datas (structures). Les fonctions et les structures ne sont pas connectées. Une fonction peut réaliser des opérations sur plusieurs types de structures tout en gardant son indépendance des structures opérées.

[2]-Orienté-objet: par exemple C++; dans ce langage, le programme est construit autour des datas (structures) et des codes (fonctions). Ceux-ci sont regroupés dans le cadre d'un seul objet (boîte noire) ce qui permet d'encapsuler et de manipuler les datas et les codes d'une manière combinatoire (P. Aitke, 1995; P.650).

2-Survol de la technologie SGDL

SGDL-system est basé sur un système de modélisation logique des objets volumiques, appelé "Logique des Formes". L'utilisateur combine les primitives géométriques décrites à l'aide d'opérateurs volumiques permettant de spécifier à la fois la nature des opérations, comme, par exemple, l'ajout ou le prélèvement de matière, et leur ordonnancement dans le temps. La description d'un objet volumique prend la forme d'un arbre volumique constitué d'opérateurs comme union, soustraction et intersection et d'opérandes qui sont, soit des primitives élémentaires (une sphère, un cylindre par exemple), soit des structures volumiques plus complexes.

Sur le plan géométrique, J. F. Rotgé propose une primitive volumique unique définie en géométrie projective. Il s'agit d'une quadrique projective générale qui regroupe toutes les primitives de la famille des quadriques en géométrie euclidienne. De là, la possibilité de représenter les trois types de surfaces (parabolique, ellipsoïdal et hyperboloïde) dans une surface unique quadrique.

Le système de coordonnées de SGDL s'appuie sur les coordonnées homogènes. Ce système contient quatre composantes (x, y, z, k) . Dans le cas où la quatrième composante k est égale à 1 $(x, y, z, 1)$, nous nous trouvons dans l'espace géométrique euclidien. Or, les coordonnées euclidiennes ne sont représentées que par trois composantes (x, y, z) , mais en ajoutant une nouvelle composante dont la valeur est égale à 1, cela n'a aucun effet sur ces coordonnées; ce qui veut dire que les deux systèmes de coordonnées $(x, y, z, 1)$ et (x, y, z) sont équivalentes. Mais, si la quatrième coordonnée (k) prend la valeur (0) , nous sommes alors dans la géométrie affine. Ce système de coordonnées permet de gérer un point à l'infini.

La densité est un autre point fondamental qui caractérise le système SGDL. La densité est un système de codage de l'espace (objet, volume) hétérogène. Ce système est contrôlé par des modifications accordées à la description des processus algorithmiques du programme, autrement dit chaque région de l'espace a une valeur numérique et le mécanisme arithmétique des attributs de densité contrôle le codage de cette répartition spatiale.

L'importance de SGDL se trouve dans la manière de traiter l'information. Il peut traiter des données numériques et des équations mathématiques. Les données géométriques et/ou graphiques peuvent être aussi traitées par le système SGDL qui retourne les résultats en format numérique. Cela signifie que SGDL est capable d'interpréter des méthodes (par exemple la méthode de tracer une ellipse) et de représenter, sous une forme numérique, les informations qui en résultent afin qu'elles soient intégrées, de nouveau, aux calculs. En résumé, le système SGDL permet aux utilisateurs, ayant les mêmes types d'informations représentées par des méthodes différentes, de traiter ces informations et d'obtenir des résultats identiques sur l'écran de l'ordinateur. C'est ce qui explique la fiabilité de ce système et son ouverture à différentes disciplines.

Sur le plan mathématique, le système SGDL se développe sur de nouvelles formules implicites et paramétriques d'une surface algébrique projective. Cette surface est le résultat de contraintes projectives contrôlées par des points de passage, des points de contact et des points de pôle ou d'inflexion (figure-59-b). La description mathématique de cette surface est contrôlée par des contraintes projectives en m degrés et en n dimensions. Par exemple, pour une équation implicite de deuxième degré $m=2$, définie dans un espace tridimensionnel $n=3$, le nombre de coefficients de l'équation est égale à 10. Dans le cas d'une quadrique (10 coefficients-1=9), neuf points en position générale dans un espace tridimensionnel définissent une quadrique. Celle-ci se distingue géométriquement par deux sections coniques (figure-59-a). Ainsi, l'équation implicite d'une quadrique est un polynôme de deuxième degré en 10 coefficients. Ceux-ci dépendent de six points de contrôle se déplaçant librement dans un espace projectif, c'est-à-dire dans un espace où la géométrie projective contrôle les positions de points les uns par rapport aux autres. Le mouvement de ces points permet d'obtenir toute la gamme des quadriques: parabolique, ellipsoïde et hyperboloïde (*SGDL technology. S. A, chapitre-VI, 1995*).

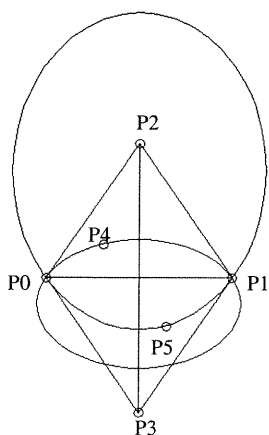
On peut dire que toutes les primitives du système SGDL sont représentées géométriquement par une quadrique dont les six points de contrôle sont:

P0 , P1: deux points de contact.

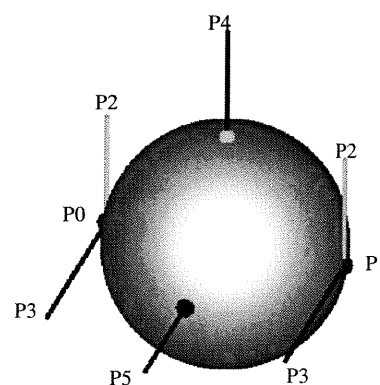
P2 , P3: deux points de pôle.

P4 , P5: deux points de passage.

Le déplacement de ces points dans l'espace conduit au changement de la forme géométrique de la quadrique. Par exemple: si on donne la valeur 0 à la quatrième coordonnée du point P4:(x, y, z, 0), ce dernier va à l'infini et la quadrique se transforme en cylindre.



(a-). Les deux sections coniques qui génèrent une surface quadrique (selon *SGDL technology, 1995*)



(b-). Points de contrôle d'une quadrique

Figure-59. Représentation géométrique d'une quadrique

3-Structure informatique

À cette étape, les informations numériques et leur traitement seront traduits en fonctions, formant le corps du programme. Pour s'assurer de la validité du programme, on effectue des tests, comme ceux qui portent sur des paramètres, des simulations, de l'interactivité et sur le partage des tâches entre les différentes fonctions du programme. Le but de ces tests est d'assurer le fonctionnement du programme. La tâche la plus importante dans la structure algorithmique ou informatique est la combinaison de fonctions et le partage de tâches (figure-60). Les fonctions ou les algorithmes représentent une base de données ou une bibliothèque numérique de l'ensemble du programme. Le résultat retourné par le programme dépend des fonctions appelées. Cette méthode permet de générer un nombre infini d'objets variables, en fonction de leurs paramètres et de leurs sources numériques.

Le schéma ci-dessous montre la structure informatique du programme. Le modèle numérique et graphique sont les points de départ. Le langage de la programmation et le système informatique forment l'étape intermédiaire, pour finalement confirmer la validité du programme.

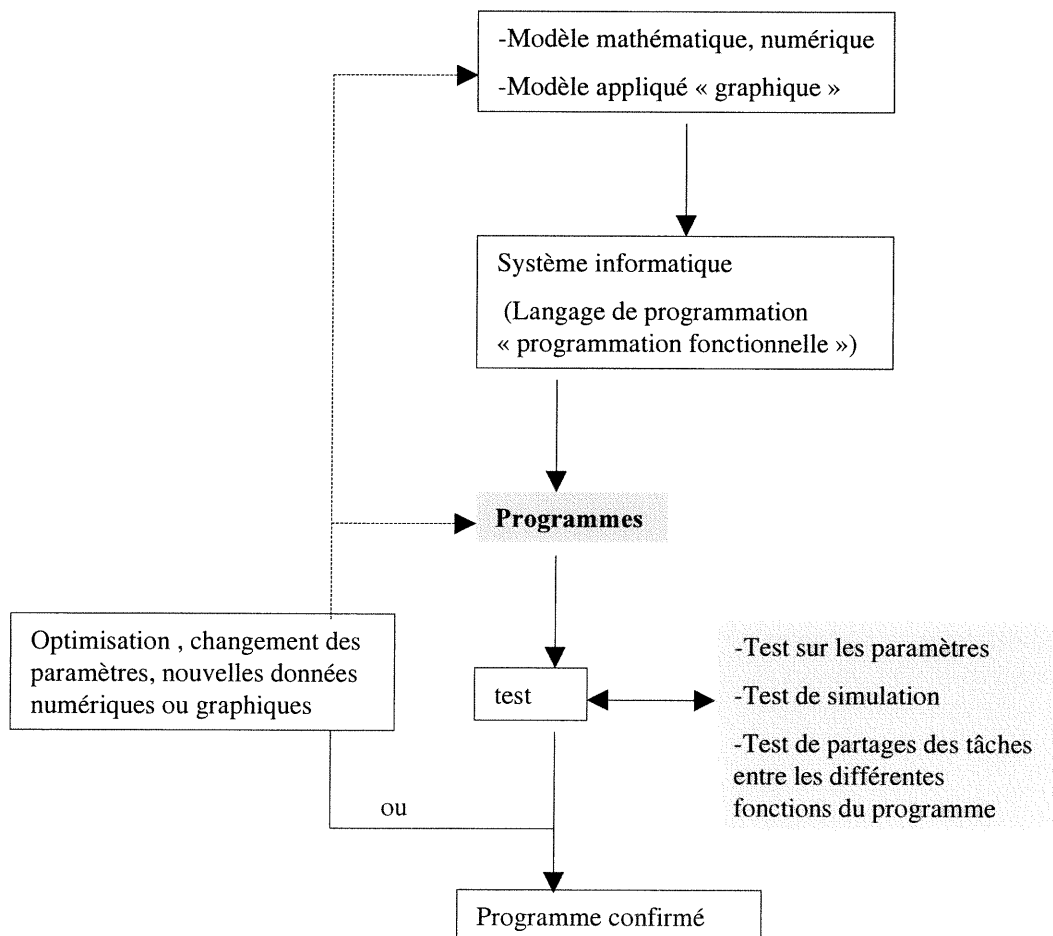


Figure-60. Schéma de la structure informatique

I-Le premier programme

Nom du programme: « Prog_pointu »

L'ensemble de huit fonctions (groupe-I) forme le programme «prog_pointu» qui génère un dôme entrelacé dont l'arc de base subit des discontinuités surfaciques: arc pointu, demi-cercle ou brisé. (pour les codes des fonctions voir l'annexe-A)

II-Le deuxième programme

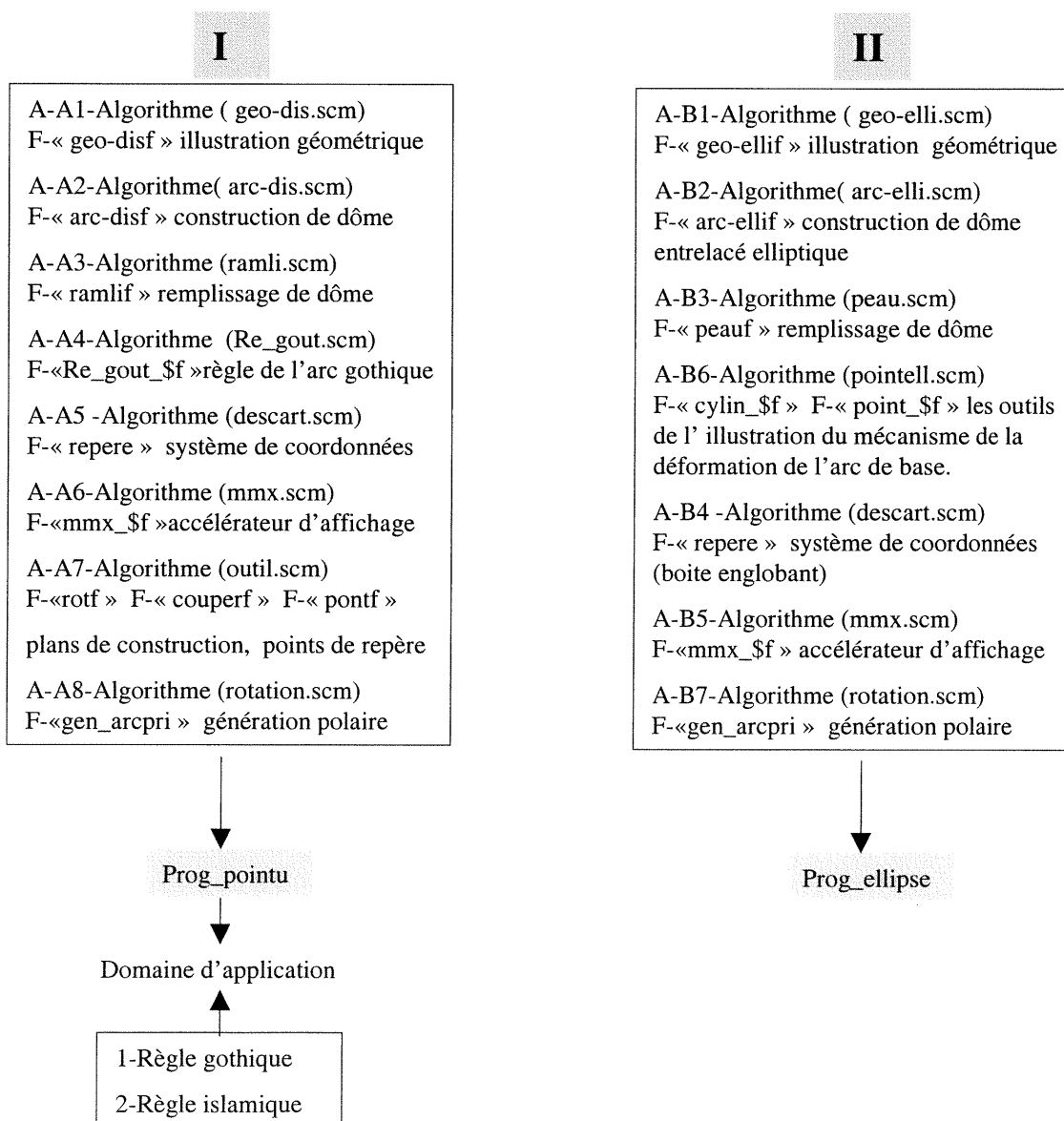
Nom du programme: « Prog_ellipse »

L'ensemble de sept fonctions (groupe-II) forme le programme «prog_ellipse» qui génère un dôme entrelacé dont l'arc de base est continu, ne subissant pas des discontinuités surfaciques: arc elliptique. (pour les codes des fonctions voir l'annexe-B)

Si l'algorithme consiste en une ou plusieurs fonctions, on écrit

A-«nom de l'algorithme»

F-« nom des fonctions »



Conclusion du chapitre-III

L'adoption des outils informatiques dans la modélisation d'un objet ou d'une forme géométrique passe par des recherches permettant d'évaluer différents types de systèmes informatiques. Ces recherches sont orientées vers des descriptions informatiques permettant la manipulation de la connaissance et des données mathématiques, numériques et graphiques. Les systèmes informatiques programmables font partie de ces descriptions. Ces systèmes transmettent les instructions à l'ordinateur à partir des codes décrits et organisés par des langages de programmations. Or, chaque système a une méthode spécifique de manipulation des données et/ou des informations et a aussi un niveau de capacité à traiter des problèmes compliqués et/ou complexes. Certaines tâches de représentations sont exprimées comme des problèmes complexes surtout lors de l'organisation d'actions. Dans le cadre de cette recherche, la complexité de la modélisation d'un objet dans un environnement dynamique demeure dans l'organisation d'actions. Ceci-dit dans le cadre d'une structure logique performante et dans la mise en relation des différentes étapes de représentations de l'objet modélisé.

La modélisation d'actions ou de processus implique un système informatique basé d'une part sur un système géométrique et d'autre part sur un langage de programmation. Le système géométrique est évalué à partir du niveau de géométrie adopté dans son concept mathématique. Un système basé sur une très haute géométrie (la géométrie projective) permet l'unification du concept mathématique de ses primitives de base et la liberté de transformations géométriques. L'importance du langage de programmation vient de sa capacité de traiter des méthodes pour la résolution de problèmes et de représenter des processus de construction de l'objet. Les langages fonctionnels permettent de traiter ce genre de problèmes et d'assurer la transparence des codes afin qu'ils soient représentables et compréhensibles.

Donc le système informatique adopté devrait satisfaire à certains critères imposés par l'objectif de la recherche. Soit le traitement des différents types d'informations, la généralité de son système géométrique et la capacité de son langage de programmation à traiter des données combinatoires et des actions. Le système SGDL avec son concept géométrique (géométrie projective) et son langage de programmation *Scheme* satisfait aux critères et aux besoins mentionnés ci-dessus. Comme on l'a vu, SGDL permet d'établir des programmes structurés à partir des fonction pouvant se partager des tâches selon des relations interactives. Ces fonctions sont représentés informatiquement par des codes permettant de générer l'objet, de lire l'histoire de la construction et enfin de participer au transfert du savoir-faire.

CHAPITRE -IV

EXPÉRIMENTATION

VALIDITÉ ET

DOMAINE D'APPLICATIONS

CHAPITRE -IV

EXPÉRIMENTATION

1-Expérimentation et domaines d'application

Dans le cadre de cette recherche, l'expérimentation vise à générer et gouverner une maquette numérique représentant toutes les informations et les données géométriques, structurales et architecturales de l'objet modélisé (un dôme entrelacé). On montrera que le modèle virtuel ou la maquette numérique générée contient les connaissances de l'objet source et que ces connaissances peuvent être contrôlées et mises à jour en ne changeant que des paramètres définis dans le cadre du programme examiné. Le but de cette expérimentation est de s'assurer de la validité de l'hypothèse mentionnée au début de la recherche et de la réalisation des objectifs. Les deux programmes «prog_pointu» et «prog_ellipse» établis dans le cadre du chapitre-III doivent permettre de modéliser un dôme entrelacé dont l'arc de base est pointu et/ou elliptique, de générer une multitude infinie de formes géométriques d'arcs de base et enfin de généraliser la génération de ces arcs sur des cas particuliers: par exemple l'arc pointu gothique et l'arc pointu islamique. Les deux programmes doivent aussi permettre d'exporter le dôme généré en vue de l'insérer dans un autre contexte architectural. Donc, notre principale occupation dans le cadre de cette expérimentation est la vérification de la capacité des deux programmes à réaliser leurs tâches mentionnées ci-dessus. Pour ce faire, on subdivisera l'expérimentation en deux étapes principales. Chacun des programmes «prog_pointu» «prog_ellipse» sera examiné selon les étapes ci-dessous.

Première étape: le cas général de la génération d'un arc pointu et/ou elliptique peut être vérifié à partir d'un test réalisé sur les programmes «prog_pointu» et «prog_ellipse». Ce test a pour but de s'assurer que la série des arcs générés par ces programmes correspond à la description géométrique d'un arc pointu et/ou elliptique sans regard à leurs références architecturales. Du point de vue géométrique, un arc pointu est le résultat de l'intersection de deux arcs de cercle. Le test effectué au cours de cette étape est purement géométrique, car il s'agit, en fait, de faire un test de validité interne sur les deux programmes. On réalise aussi un test sur le partage des tâches réalisées par la combinaison des différentes fonctions des programmes. Ce test permettra de vérifier le fonctionnement des relations interactives entre les différentes fonctions des programmes: de démontrer la combinaison des fonctions permettant de générer un arc pointu ou elliptique avec ou sans illustration géométrique et de démontrer les dômes entrelacés résultants de cette combinaison.

Deuxième étape: on cherche la validité des deux programmes dans d'autres domaines d'applications. En architecture, par exemple, un arc, dont la forme géométrique est pointue, peut être représenté architecturalement par un arc gothique, islamique, lancé ou d'autres styles d'arcs, dont la forme géométrique est le résultat de l'intersection de deux arcs de cercle. Autrement dit, le programme devrait générer tous les styles d'arcs pointus compris dans la description géométrique d'un arc pointu. Pour réaliser ce test, nous examinerons le programme «prog_pointu» sur la génération d'un arc islamique et d'un arc gothique. Quant au programme «prog_ellipse», nous le testerons sur la génération des arcs architecturaux elliptiques surhaussés et/ou surbaissés. Le test final portera sur l'exportation des divers dômes entrelacés générés par l'un des deux programmes en vue de les insérer dans un contexte architectural.

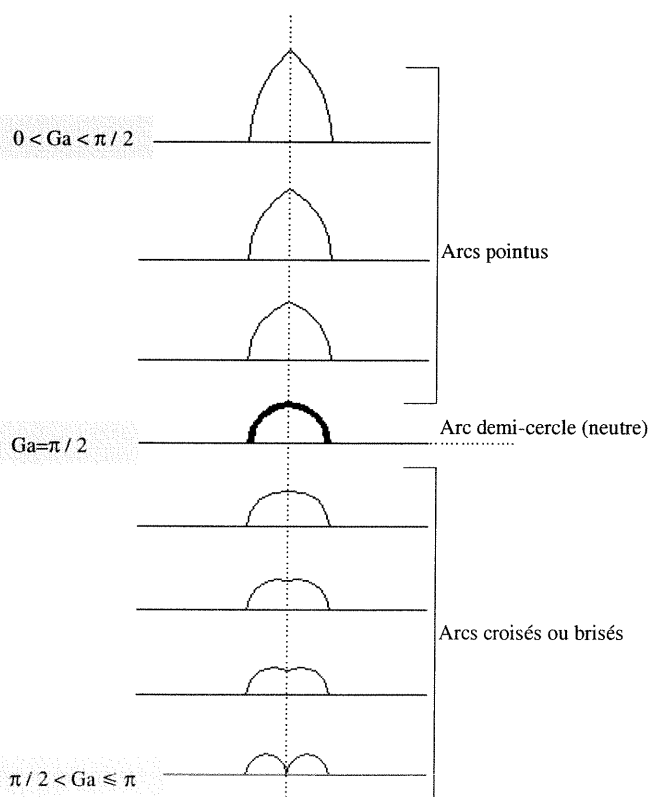
En résumé, la première étape aura pour tâche de valider les deux programmes sur le cas général de la génération d'un arc pointu et/ou elliptique; et de ce fait, les arcs générés seront validés géométriquement. Dans la deuxième étape, nous nous occuperons à vérifier la validité des deux programmes dans le domaine de l'architecture; les arcs générés pourront alors porter un style architectural.

2-Expérimentation et test de validité du programme «prog_pointu»

Validité interne «prog_pointu»: Ce test aura pour but de vérifier que le programme «prog_pointu» est capable de générer toutes les formes géométriques d'un arc pointu résultant de l'intersection des deux arcs de cercle (figure-61).

Le paramètre G_a est l'angle de construction verticale de la moitié d'un arc pointu (voir paragraphe-2.3.A.2.I, chapitre-II). G_a contrôle la forme géométrique des arcs pointus et brisés enregistrés par le programme «prog_pointu». Notre préoccupation est d'alimenter le paramètre G_a par différentes valeurs de 0 à π , d'inventorier les formes possibles enregistrées par le programme et enfin de vérifier si ces formes correspondent à la description géométrique d'un arc pointu. G_a prend des valeurs de l'intervalle ($0 < G_a \leq \pi$).

Figure-61. L'intervalle de l'angle G_a et les principales formes géométriques d'un arc pointu enregistrées par le programme «prog_pointu».



On peut générer un nombre infini d'arcs de base dont l'angle G_a fait partie de l'intervalle $(0 < G_a \leq \pi)$. (Voir annexe-A, A-A1-«geo_dis.scm» F-«geo_disf»)

À partir de ces arcs, le programme peut générer différents types de dômes entrelacés dont le nombre d'arcs utilisés dans la construction est contrôlé par le paramètre N_a . (Voir annexe-A, A-A2-«arc_dis.scm» F-«arc_disf»)

En fait, le style de dômes générés dépend du style de l'arc de base. Il y a quatre types principaux d'arcs générés selon des différentes valeurs de l'angle G_a :

a. $(0 < G_a < \pi/2)$: l'arc généré est un arc pointu. (figure-62-a)

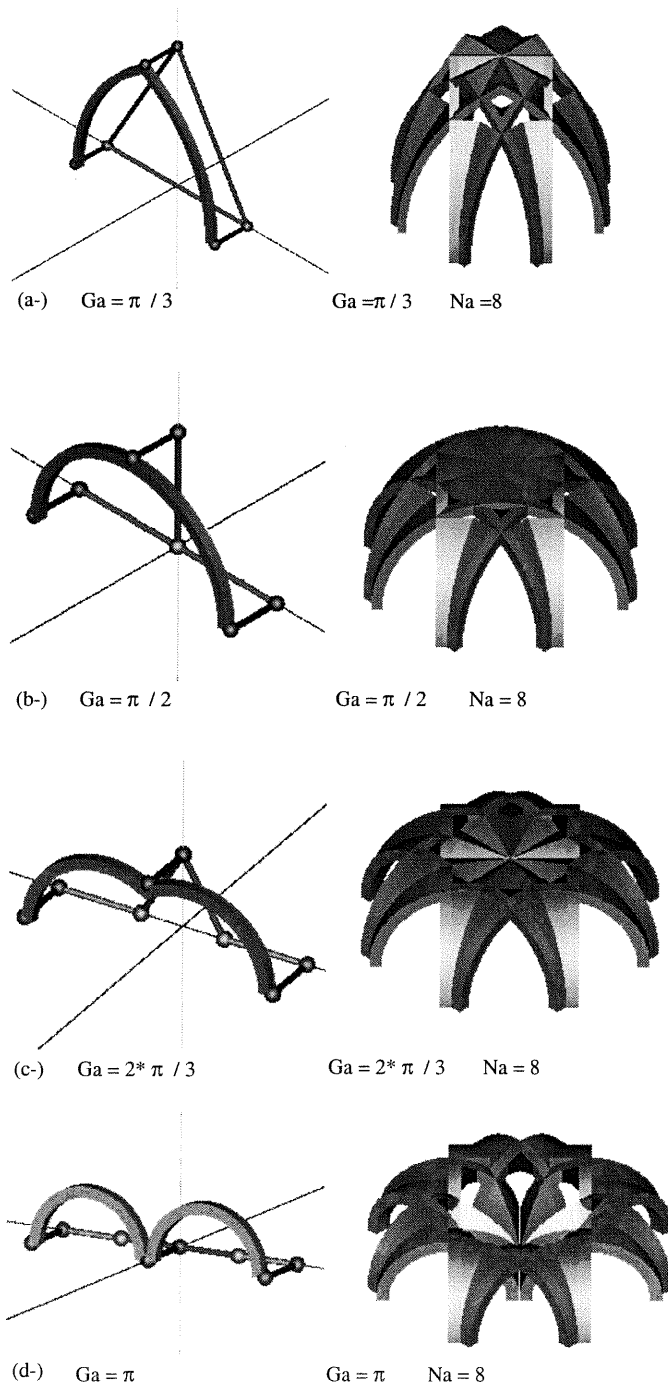
b. $(G_a = \pi/2)$: produit un arc en demi-cercle. Chacun des deux arcs de cercle formant l'arc de base est représenté par un quart de cercle dont l'angle est $\pi/2$ (figure-62-b).

c. $(\pi/2 < G_a < \pi)$: le programme produit un arc croisé ou brisé. Par exemple: $G_a = 2 * \pi/3$ le programme génère un arc croisé dont la forme géométrique est le résultat de l'intersection de deux arcs de cercle (figure-62-c).

d. $(G_a = \pi)$: l'arc obtenu est représenté par deux arcs adjacents dont le point de rencontre est un point de tangence (figure-62-d).

À cette étape le programme génère uniquement les arcs de base. S'ensuivent les dômes qui correspondent au type de ces arcs sans s'attarder au style architectural ou constructif. Tel est donc le cas général de modélisation ou de génération des dômes entrelacés.

Figure-62. différents types de dômes entrelacés dont les arcs de base sont pointus, neutres ou brisés.



Validité externe de « prog_pointu »

dans cette étape, nous chercherons à nous assurer de la validité du programme «prog_pointu» dans d'autres domaines que la géométrie (par exemple dans le domaine de l'architecture). Nous le testerons sur deux cas particuliers connus en architecture: la règle d'un arc gothique et la règle d'un arc islamique. L'angle G_a préside à l'opération de la génération de ce type d'arcs. Les deux paragraphes 2.3.A.3.I et 2.3.A.3.II chapitre-II- montrent les deux équations des deux règles gothique et islamique.

Ga de l'arc gothique: $G_a = \arccos(\tan(\pi/8) / (1 + \tan(\pi/8)))$

Ga de l'arc islamique: $G_a = \arccos(1 / (n + 1))$

La règle gothique est une fonction indépendante que l'on charge au besoin. La règle islamique est intégrée dans le programme et fait partie de son code. (Voir annexe-A, A-A2-«arc_dis.scm » F-«arc_disf »)

La fonction de filtrage du programme «prog_pointu» peut filtrer l'angle G_a et décider s'il fait partie de l'intervalle $(0 < G_a \leq \pi)$. Si l'angle G_a est reconnu par le programme, celui-ci génère l'arc de base avec l'illustration de sa règle géométrique.

(Voir annexe-A, A-A1-«geo_dis.scm » F-«geo_disf»)

À chaque fois qu'il y a une nouvelle règle: on l'analyse, on calcule l'angle G_a et on l'adopte comme un cas particulier de génération .

Le programme peut générer un seul arc à la fois, à savoir qu'il est possible de générer un arc gothique, islamique ou pointu. Dans le choix d'un style d'arc apparaissent deux paramètres: le paramètre N_{go} qui permet de générer un arc gothique et le paramètre N_{is} qui permet de générer un arc islamique .

Le paramètre N_{go} peut prendre une valeur 1 ou 0.

Dans le cas où $N_{go}=1$, le programme produit un arc selon la règle gothique (figure-63).

Dans le cas où $N_{go}=0$, le programme bloque la génération de ce type d'arc et passe en vérifiant d'autres paramètres.

Les données géométriques de la règle gothique sont: les points C_1 et C_2 sont les deux centres de l'arc. Le point D est l'intersection des deux courbes de l'arc. Les points (1,2,3,4) représentent le plan dans lequel les deux extrémités de l'arc gothique se localisent (figure-63).

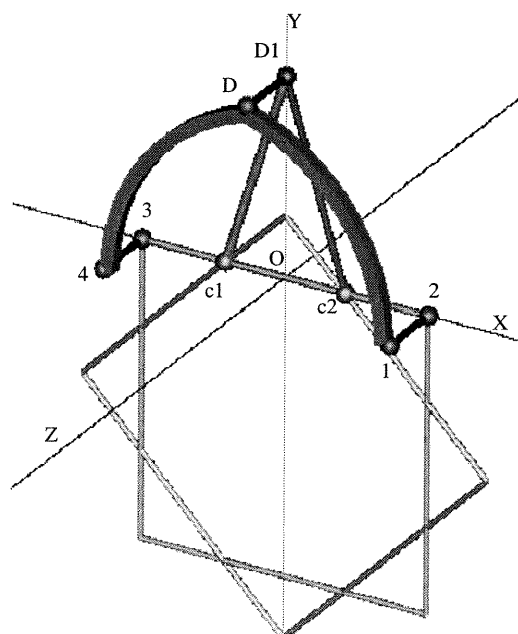
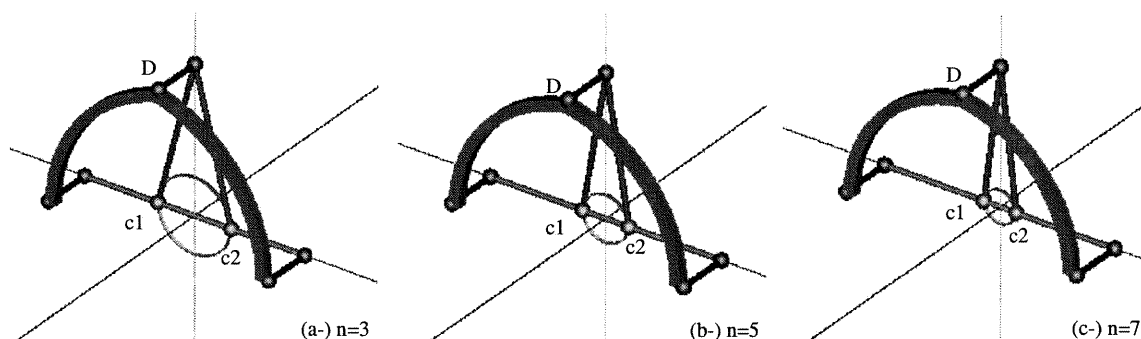


Figure-63. La génération d'un arc pointu selon la règle gothique: dans le cas où ($N_{go}=1$ et $N_{is}=0$) le programme exécute la relation $G_a = \arccos(\tan(\pi/8) / (1 + \tan(\pi/8)))$ et génère un arc gothique avec illustration de sa règle géométrique. (Voir annexe-A, A-A4-«R_gout.scm » F-«R_gout_\$f »)

Le paramètre «Nis» suit des démarches analogues à celle décrites précédemment. Si $Nis=1$, le programme génère un arc islamique (figure-64), mais si $Nis=0$, le programme bloque la génération de l'arc islamique et passe pour vérifier d'autres paramètres. Les données géométriques de l'arc islamique sont: C_1 et C_2 sont les deux centres de l'arc islamique, le point D est l'intersection des deux courbes de l'arc. Note: dans le cas où $n=\infty$, $Ga=\pi/2$, le programme produit un arc demi-cercle



(Figure-64). Génération d'un arc pointu selon la règle islamique: dans le cas où $Nis=1$ et $Ngo=0$ le programme exécute la Relation $Ga=\arccos(1/(n+1))$ et génère un arc islamique avec illustration de sa règle géométrique. La forme de l'arc généré dépend du nombre de divisions «n» (selon la valeur de n le programme génère le cas (a-), (b-) ou (c-)).

On remarque que le programme «prog_pointu» donne la priorité à la génération des deux cas particuliers. Si ces derniers sont bloqués en donnant à leurs paramètres « $Ngo=Nis=0$ », le programme passe à l'exécution du cas général de la génération d'un arc pointu. Le programme «prog_pointu» accepte un seul paramètre $Ngo=1$ ou $Nis=1$. Dans ce cas, il génère un arc selon la règle choisie, gothique ou islamique. Si les deux paramètres « $Ngo=Nis=1$ », le programme provoque une erreur donnant un message «au moins un des deux paramètres Ngo ou Nis doit être égal à 0». Dans le cas où les deux paramètres « $Ngo=Nis=0$ », le programme exécute le cas général de la génération d'un arc pointu, selon l'angle Ga déjà choisi dans le cadre des paramètres. Le programme génère aussi un dôme entrelacé selon le type de l'arc généré.

En résumé, les types d'arcs qui sont générés par ce programme sont basés sur l'intersection de deux arcs de cercle. Quel que soit le style, architectural ou constructif, de ces types, on peut les générer par ce programme. En ce qui concerne les nouvelles règles que l'on veut intégrer à la structure du programme, il nous suffit de savoir que ces nouvelles règles et le programme «prog_pointu» ont la même base géométrique. Dans l'affirmative, on est assuré que cette règle trouvera sa place dans la série des arcs enregistrés par le programme. Ensuite, on générera tous les dômes entrelacés correspondant à ce type d'arc.

2.A- Combinaison des fonctions et tâches du programme «prog_pointu»

Le programme «prog_pointu» consiste en plusieurs fonctions générant différents types d'arcs ainsi que leurs illustrations géométriques. Les fonctions se partagent les tâches ou se regroupent selon l'objet que l'on désire produire. Le programme peut ainsi exécuter deux tâches principales: la génération des différents types d'arcs pointus et l'utilisation de ces arcs comme arcs de base dans la construction des dômes entrelacés.

I-Première tâche du programme « prog_pointu »

Le programme peut générer des arcs pointus, neutres ou brisés. Il peut aussi générer un arc gothique ou islamique d'après leurs règles géométriques.

(Voir annexe-A, A-A1-«geo_dis.scm » F-«geo_disf »)

La figure-65 représente l'illustration géométrique de l'arc de base. Cette dernière est générée par un ensemble de fonctions se regroupant dans un seul algorithme « geo_disf ». Les paramètres de ces fonctions sont:

(Rc): est le rayon du dôme.

(Na): est le nombre d'arcs générés.

(Gama) ou (Ga): est l'angle de construction vertical de l'arc de base.

(Be): est l'angle de construction horizontal de l'arc de base.

(Nis et Ngo): sont les paramètres d'initialisation des deux règles islamique et gothique.

(n): est le nombre de divisions de la portée d'un arc islamique.

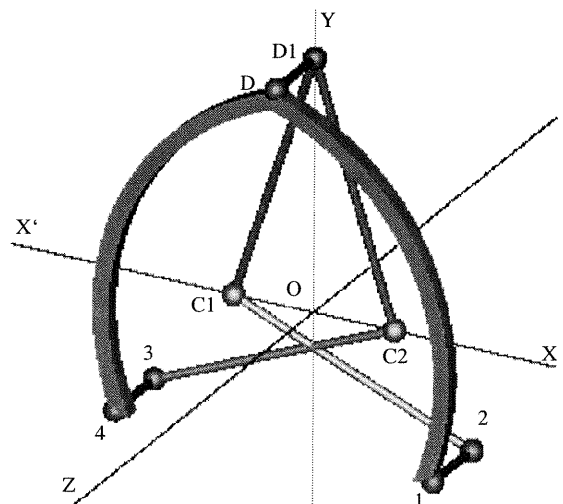
Les deux lignes de code ci-dessous représentent la fonction génératrice d'un arc pointu avec son illustration géométrique.

```
(geo_disf Rc Na Ga Be Nis n Ngo)
(geo_disf 10 8 (* (SGcst_pi) (/ 2 5)) (* (SGcst_pi) (/ -1 8)) 0 3 0)
```

Description géométrique du cas général de génération d'un arc pointu: les points (1,2,3,4) contrôlent le plan de la construction horizontale de l'arc de base. Ce plan est toujours parallèle au plan de référence XOZ. Les deux points C1 et C2 sont les deux centres de l'arc de base. Ces deux centres se déplacent sur l'axe XX'. Le point D représente le point d'inflexion (rupture) ou la tête de l'arc pointu où l'arc change sa courbure. Les points (2,3,C1,C2,O et D1) sont situés dans le plan de référence XOY et guident la construction de l'arc. Les points 1, 4 et D sont situés dans le plan de construction de l'arc lequel est toujours parallèle au plan de référence XOY (figure-65).

Les paramètres de la génération de l'arc (1,4,D) de la figure-65 sont représentés par la fonction (geo_disf Rc=10, Na=1, Ga=2π/5, Be=-π/8, Nis=0, n=1, Ngo=0). Ga et Be sont les deux angles qui contrôlent la naissance de l'arc, dont les deux points l'extrêmes 1 et 4 se situent dans le plan 1, 2, 3 et 4. Celui-ci est parallèle au plan de référence XOZ et est situé au-dessous.

Figure-65. Un arc pointu dans son cas général où l'angle Be non zéro.



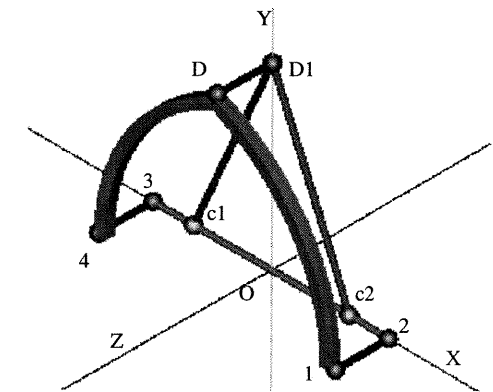
Le programme «prog_pointu» peut produire trois types d'arcs pointus représentant des points d'inflexion dans une série d'arcs générés. Selon la valeur de l'angle G_a , les deux centres C_1 et C_2 et les points de naissance 2 et 3 de l'arc se situent dans le plan de référence XOZ . Cela veut dire que le plan de construction horizontale PCH et le plan de référence XOZ sont superposés. Dans ce test, G_a est variable et $Be=0$.

1-Cas-1 $G_a=2\pi/5$, $Be=0$, (figure-66-a), l'arc généré est un arc pointu et les centres sont opposés par rapport au plan de référence ZOY .

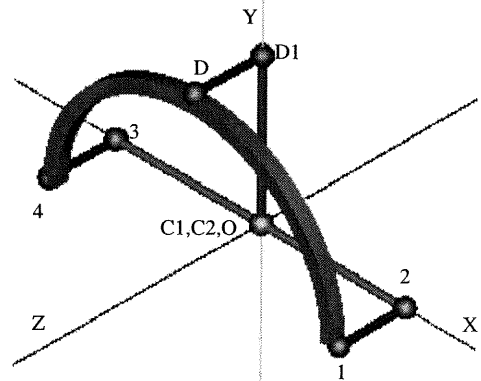
2-Cas-2 $G_a=\pi/2$, $Be=0$, (figure-66-b), l'arc généré est un arc neutre ou un demi-cercle et les deux centres C_1 et C_2 sont superposés à l'origine du dôme O .

3-Cas-3 $G_a=2\pi/3$, $Be=0$, (figure-66-c), l'arc généré est un arc brisé ou croisé et les deux centres C_1, C_2 ont changé leurs positions par rapport à l'origine O .

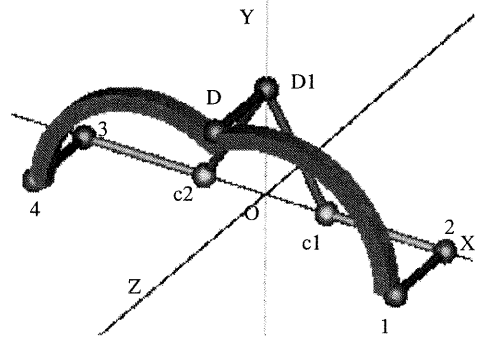
Le programme «prog_pointu» peut générer un arc pointu avec son illustration géométrique. Celle-ci représente un système géométrique tridimensionnel contrôlant les changements formels de l'arc généré. Les points de repère de chaque illustration sont, soit des points-guides installés au plan XOY (comme les points 2,3, D_1 , C_1 , C_2 et O), soit des points-orientés installés sur l'arc lui-même (comme les points 1,4 et D). Les points-guides contrôlent le déplacement des points-orientés dans un espace tridimensionnel.



(a-) Cas-1 ($G_a=2\pi/5$, $Be=0$) « arc brisé »



(b-) Cas-2 ($G_a=\pi/2$, $Be=0$) « arc neutre »



(c-) Cas-3 ($G_a=2\pi/3$, $Be=0$) « arc brisé »

(Figure-66). Les trois types d'arcs pointus générés par le programme «prog_pointu» (l'angle G_a est variable et l'angle $Be=0$).

Dans le cas où les deux angles G_a et B_e sont variés, apparaît alors un tableau de neuf types d'arcs pointus: trois types croisés, trois types en demi-cercle et trois types pointus. Chacun des deux angles G_a et B_e occupe trois intervalles.

L'angle $G_a * 3$ intervalles = trois (types_ G_a) d'arcs

L'angle $B_e * 3$ intervalles = trois (types_ B_e) d'arcs

Trois (types_ G_a) * trois (types_ B_e) = 9 types d'arcs

Par exemple: on cherche le type d'arc dont l'angle $G_a = \pi/3$ et l'angle $B_e = -\pi/8$.

$G_a = \pi/3$ est dans l'intervalle $(0 < G_a < \pi/2)$.

$B_e = -\pi/8$ est dans l'intervalle $(-G_a < B_e < 0)$

L'arc qui correspond à ces deux angles est l'arc numéro 7.

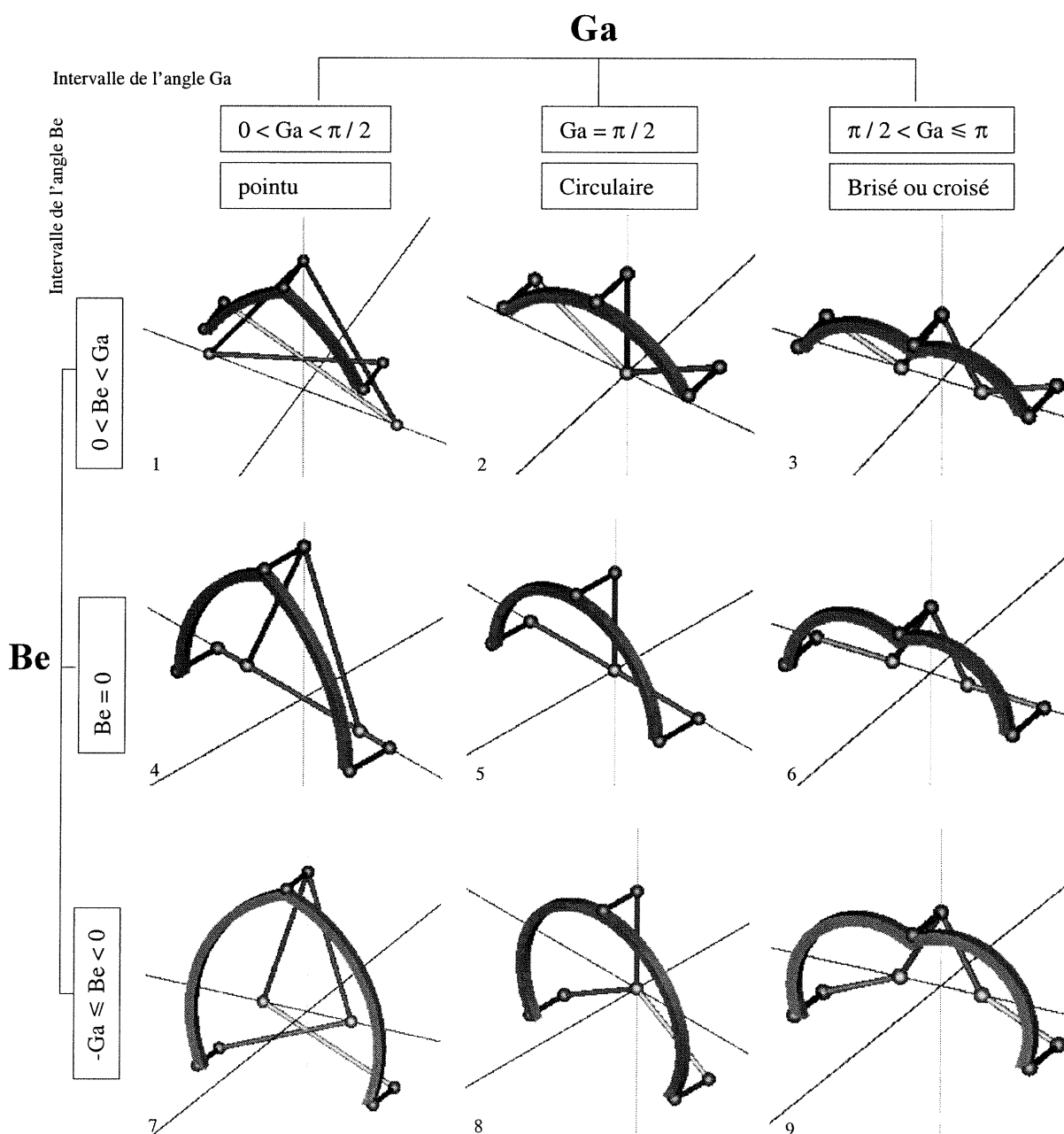


Figure-67. Un tableau de neuf types d'arcs pointus: (les angles G_a et B_e sont variables)

II-Deuxième tâche du programme « prog_pointu »

La génération d'un dôme entrelacé avec le remplissage: les fonctions responsables de cette génération sont représentées par les lignes de code suivantes:

```
1-(DLuni
2-(gen_arcpri 8 (arc_disf 10 8 (* (SGcst_pi) (/ 1 2)) (* (SGcst_pi) (/ -0 8)) 0 3 0))
3-(gen_arcpri 8 (ramlif 10 8 (* (SGcst_pi) (/ 1 2)) (* (SGcst_pi) (/ -0 8)) 0 3 0))
4-(reper 10)
(Voir annexe-A, A-A2-«arc_dis.scm » F-«arc_disf »).
```

La ligne du code 1 représente la fonction:

(arc_disf (Rc) (Na) (Gama) (Be) (Nis) (n) (Ngo)

dont les paramètres sont:

Rc=10, Na=8, Gama= $\pi/2$, Be=0, Nis=0, n=3, Ngo=0.

Cette fonction génère l'arc de base.

La ligne du code 2 représente la fonction récursive: (gen_arcpri (Na)) qui génère les arcs formant la structure du dôme entrelacé.

La ligne du code 3 consiste en deux fonctions:

1-La fonction

(ramlif (Rc) (Na) (Gama) (Be) (Nis) (n) (Ngo),

dont les paramètres sont:

Rc=10, Na=8, Gama= $\pi/2$, Be=0, Nis=0, n=3, Ngo=0.

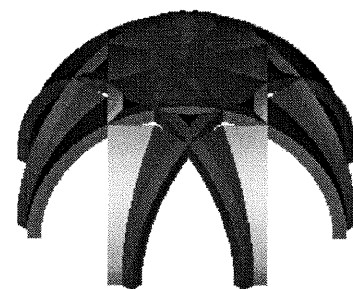
Cette fonction génère l'unité du remplissage du dôme entrelacé.

2-La fonction (gen_arcpri (Na)) génère le remplissage du dôme complet.

La ligne du code 4 représente la fonction: (reper 10) génère le système de coordonnées.

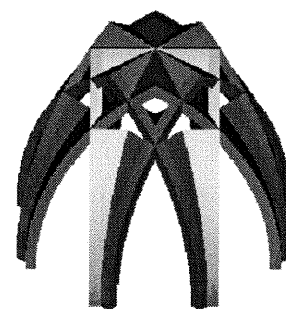
En unifiant toutes les fonctions et en utilisant la fonction DLuni de la ligne de code 1, le programme nous retourne un dôme entrelacé. Dans le cas où Ga est variable et Be=0, le programme génère trois types de dômes entrelacés: croisé, pointu et/ou demi-sphère (figure-68).

Pour avoir un dôme entrelacé selon le style architectural gothique ou islamique, il suffit d'activer un des deux paramètres Nis ou Ngo. le paramètre Ngo=1 génère un dôme entrelacé selon la règle gothique. Le paramètre Nis=1 génère un dôme entrelacé selon la règle islamique.



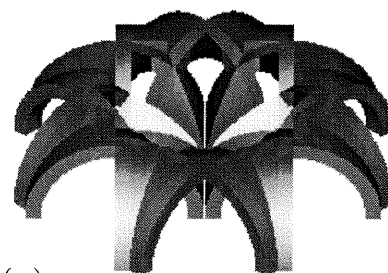
(a-)

Rc=10, Na=8, Gama= $\pi/2$, Be=0 Ngo=0, n=3, Nis=0



(b-)

Rc=10, Na=8, Gama= $\pi/3$, Be=0, Ngo=0, n=3, Nis=0



(c-)

Rc=10, Na=8, Gama= π , Be=0, Ngo=0, n=3, Nis=0

(Figure-68). La génération d'un dôme entrelacé avec le remplissage

Dans le cas où les deux paramètres G_a et Be sont variables, neuf types de dômes entrelacés peuvent être générés: trois types de dômes pointus, trois types de dômes demi-sphères et trois types de dômes croisés (figure-69). Le tableau des dômes ci-dessous correspond aux arcs de base générés dans le cadre de la première tâche (paragraphe 2.A, figure-67). Ce qui signifie que chaque type d'arc peut produire un type de dôme correspondant aux angles G_a , Be et respectant la règle d'assemblage des dômes entrelacés (mentionnée au paragraphe 2.2.C, chapitre-II).

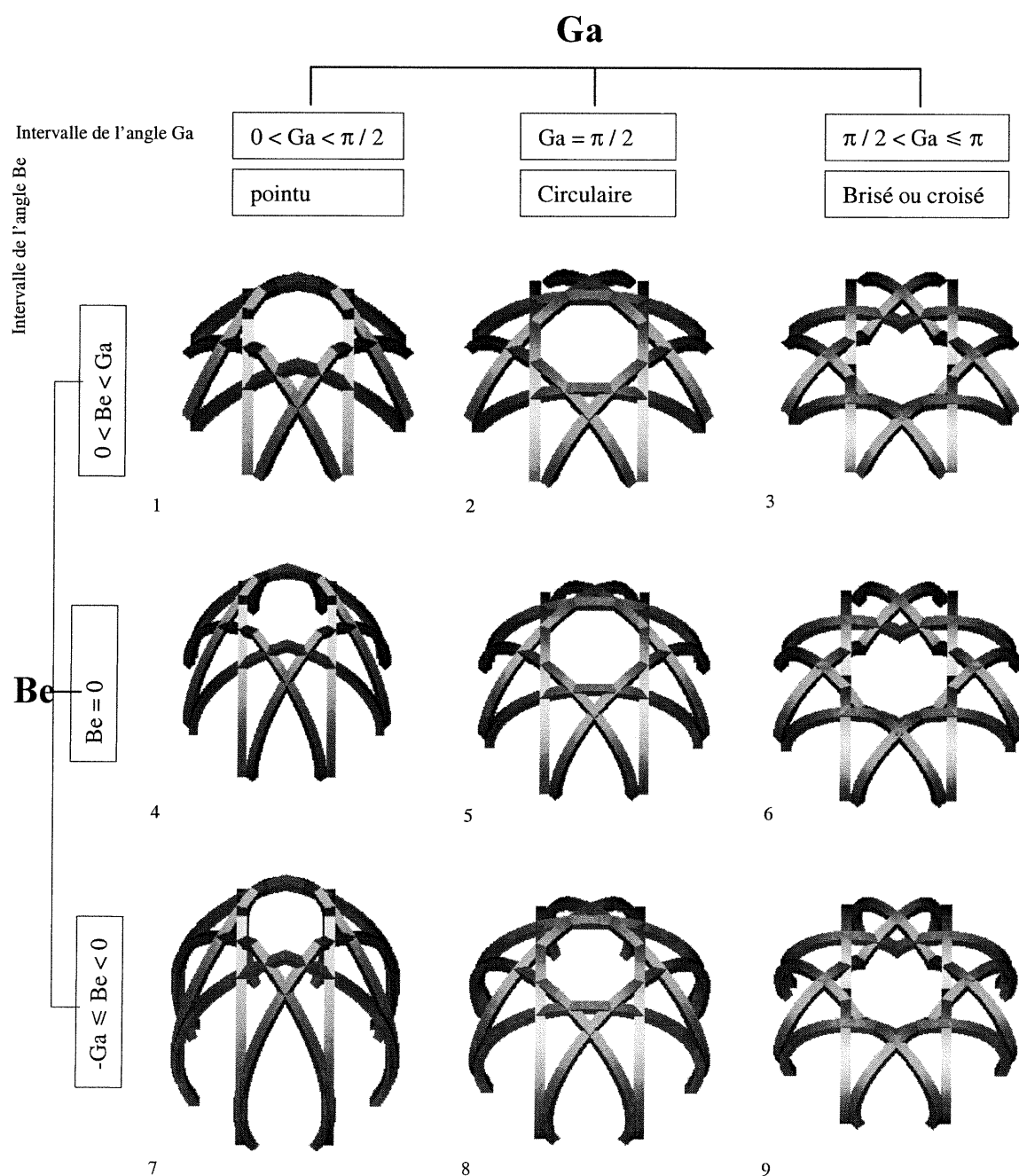


Figure-69. Les neuf types de dômes entrelacés générés selon les deux variables G_a et Be .

3-Expérimentation du programme « prog_ellipse »

Le programme «prog_ellipse» peut générer un dôme entrelacé dont la forme géométrique de l'arc de base est elliptique. Ce qui distingue ce programme du précédent, c'est la base géométrique. Le programme précédent «prog_pointu» est fondé sur des modèles mathématiques et numériques, alors que le programme «prog_ellipse» s'appuie sur des modèles géométriques et graphiques appliqués.

Les paramètres utilisés dans le programme «prog_ellipse» sont:

(Rc) est le rayon du dôme modélisé.

(Na) est le nombre d'arcs utilisés dans la construction du dôme entrelacé.

(Fa) est le facteur de la déformation de l'arc de base.

La forme géométrique de l'arc de base est contrôlée par le paramètre Fa. Celui-ci change proportionnellement le petit axe de l'ellipse par rapport au grand axe. Fa ici contrôle le point de passage P4: (vector 0 Rey 0 1) représenté dans la figure-71 par le point 2. ($Rey = Fa * Rex$) où Rex est le petit axe de l'arc ($Rex = 0, 1$) et Rey est le grand axe (2, 4). Au fur et à mesure la valeur de Fa croît, l'axe Rey diminue et le système génère un arc surbaissé.

L'intervalle du facteur Fa est ($0 < Fa < \infty$) (figure-70). À partir de cet intervalle, on détermine trois cas principaux:

a- ($1 < Fa < \infty$) produit un arc elliptique surbaissé.

b- ($Fa = 1$) produit un arc neutre ou un arc demi-cercle.

c- ($0 < Fa < 1$) produit un arc elliptique surhaussé.

Le paramètre Ga est l'angle entre l'axe XX' et le rayon OB qui trace l'ellipse, l'intervalle de l'angle Ga est ($0 \leq Ga \leq \pi$). Au fur et à mesure que Ga croît de 0 à π , le point E trace un arc elliptique (figure-71).

Le point A circule sur le cercle dont le diamètre est le petit axe 1,3

Le point B circule sur le cercle dont le diamètre est le grand axe 2,4

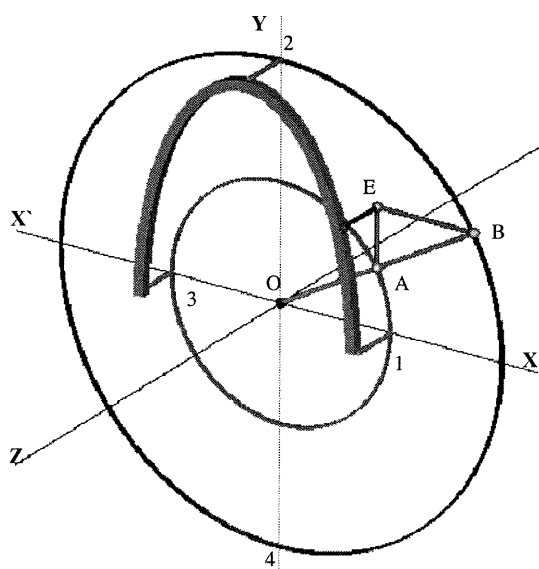


Figure-71. Tracer un arc elliptique utilisant deux cercles concentriques et un rayon

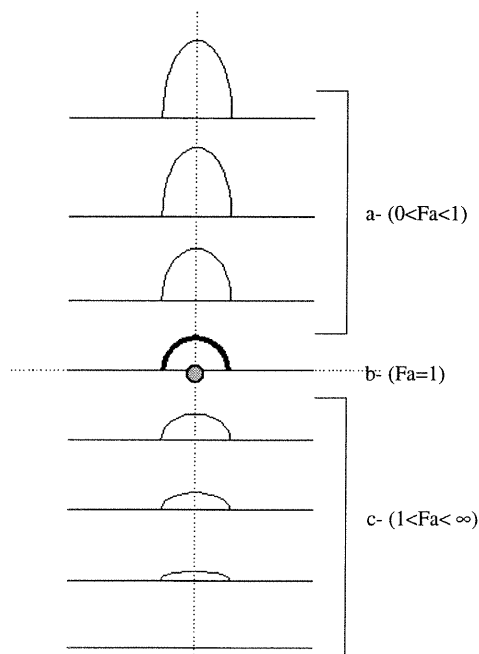


Figure-70. Intervalle du facteur Fa

L'ensemble du rayon OB, les deux lignes BE, AE et le point E forment le traceur de l'arc elliptique. Le rayon OB balaie le plan vertical xoy en tournant autour de l'axe zz' ; selon la valeur de l'angle G_a , le traceur de l'ellipse positionne le point E.

Selon la valeur de l'angle G_a , on peut déterminer deux cas généraux et trois cas particuliers. Lorsque l'angle G_a prend une valeur définie comme 0 , $\pi/2$ ou π nous sommes dans les cas particuliers. Dans les cas généraux, l'angle G_a prend sa valeur des intervalles $(0 < G_a < \pi/2)$ ou $(\pi/2 < G_a < \pi)$.

Les trois cas particuliers sont:

1- $G_a=0$: les points E et A sont superposés au point 1 et le rayon $O,1=O,A$ représente le petit axe de l'ellipse. (figure72-I)

2- $G_a=\pi/2$: les points E et B sont superposés au point 2 et le rayon $O,2=O,B$ représente le grand axe de l'ellipse (figure-72-III).

3- $G_a=\pi$: les points E et A sont superposés au point 3 et le rayon $O,3=O,A$ représente le petit axe de l'ellipse (figure72-V).

Les deux cas généraux sont:

1- $(0 < G_a < \pi/2)$: l'angle $G_a=\pi/6$ fait partie de cet intervalle (figure-72-II).

2- $(\pi/2 < G_a < \pi)$: l'angle $G_a=2\pi/3$ fait partie de cet intervalle (figure-72-IV).

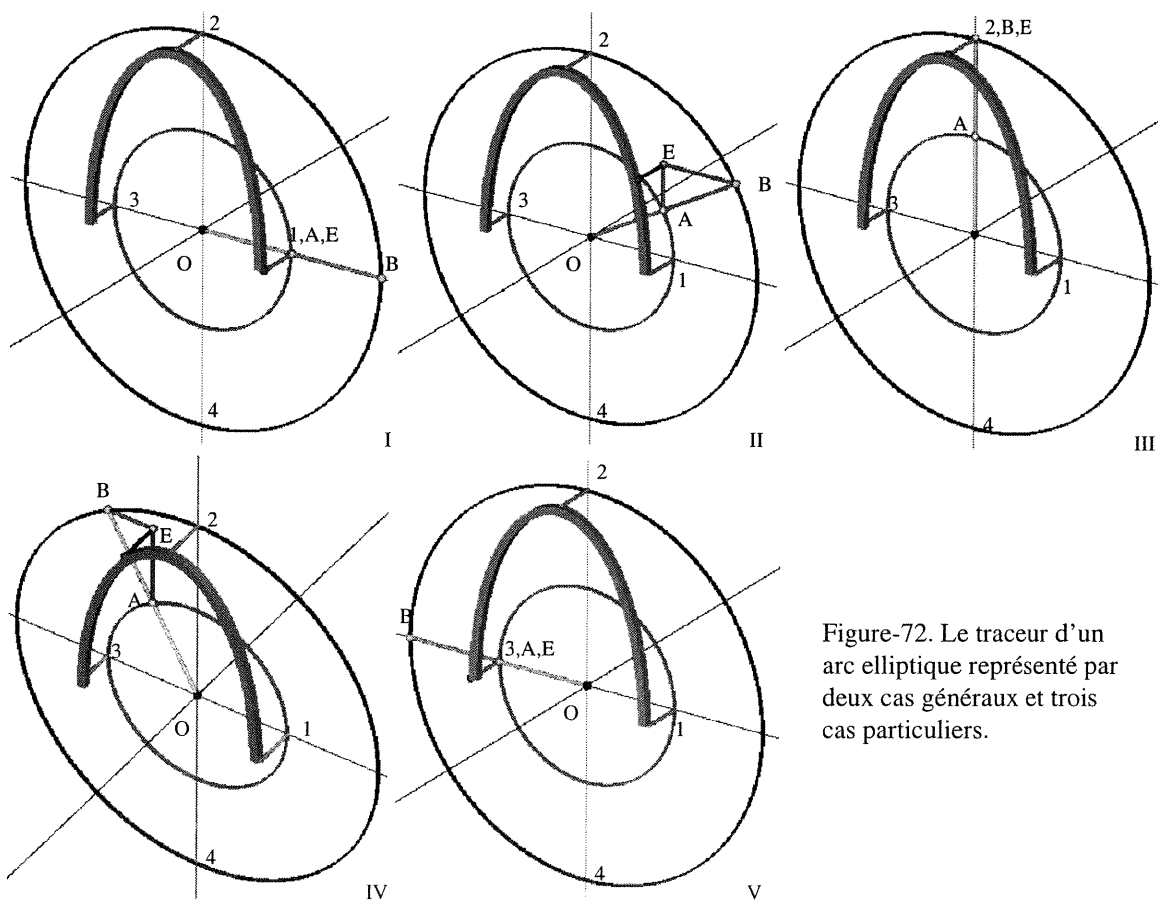


Figure-72. Le traceur d'un arc elliptique représenté par deux cas généraux et trois cas particuliers.

3.A-Combinaison des fonctions et Tâches du programme «prog_ellipse»

Le programme peut réaliser deux tâches principales: la génération des arcs elliptiques surhaussés, neutres et surbaissés et les dômes entrelacés qui correspondent à ce type d'arcs.

I-Première tâche du programme « prog_ellipse »

La génération d'un arc elliptique avec son illustration géométrique est la première tâche que l'on peut réaliser par ce programme. La fonction responsable de cette génération est:

```
(geo_ellif Rc Na Fa Ga)
(geo_ellif 10 8 (/ 3 2) (* (SGcst_pi) (/ 1 3)))
(Annexe-B, A-B1-«geo_elli.scm»F-«geo_ellif»)
```

Selon la valeur du facteur Fa on caractérise trois types d'arcs elliptiques:

Cas-1: (Fa=1/2 et Ga=π/6)

un arc elliptique surhaussé est généré selon le paramètre Fa=1/2, dans ce cas, le grand axe 2,4 égale à 2 fois le petit axe 1,3 de l'ellipse (figure-73-I).

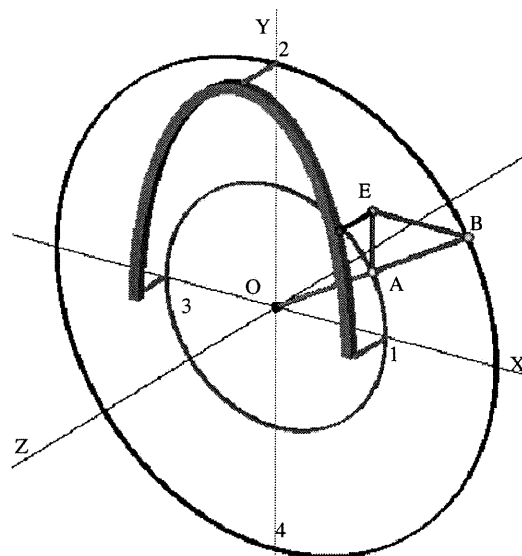
Cas-2: (Fa=1 et Ga=π/3)

un arc elliptique en demi-cercle est généré selon Fa=1. Les deux axes sont égaux et les deux cercles générateurs sont superposés (figure-73-II).

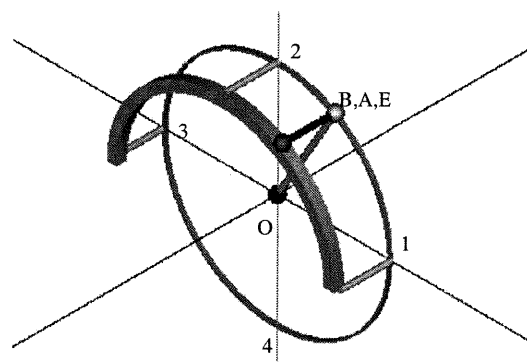
Cas-3: (Fa=2 et Ga=π/4)

un arc elliptique surbaissé est généré selon Fa=2, c'est l'inverse du cas-1, l'axe 2,4 est plus petit que l'axe 1,3 (figure-73-III).

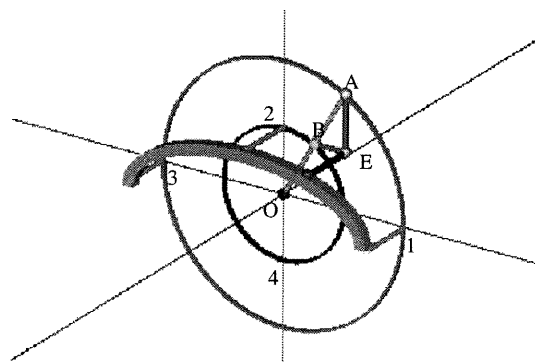
Figure-73. Le changement du type d'arcs elliptiques entre la forme surhaussée et surbaissée.



I- Cas-1- arc surhaussé $0 < Fa < 1$, $Fa = 1/2$



II- Cas-2, arc neutre un demi-cercle $Fa = 1$



III- Cas-3, arc surbaissé $1 < Fa < \infty$, $Fa = 2$

II-Deuxième tâche du programme «prog_ellipse»

Le programme «prog_ellipse» Peut générer différents types de dômes entrelacés, surhaussés, demi-sphériques et/ou surbaissés dont les arcs de base sont elliptiques. Les paramètres contrôlant cette génération sont N_a et F_a .

Le paramètre (N_a) est le nombre d'arcs utilisés dans la structure du dôme entrelacé. N_a joue un double rôle dans la génération du dôme entrelacé ellipsoïdal.

1-Il contrôle le nombre d'arcs utilisés dans la construction du dôme. (N_a) est toujours un nombre pair et plus grand que six.

2-Il contrôle l'angle T_e qui positionne l'arc de base par rapport à l'axe XX' , de sorte que la position de cet arc respecte la règle d'assemblage du dôme entrelacé (Voir chapitre-II, paragraphe 2.2.C). Selon l'analyse géométrique, l'angle $T_e = \pi/N_a$ (Voir chapitre-II, paragraphe 2.2.B).

Le paramètre (F_a) est le facteur de déformation de l'arc de base. F_a contrôle aussi la déformation du dôme. D'après la valeur de F_a , on peut engendrer un dôme ellipsoïdal surhaussé, surbaissé ou un dôme neutre en demi-sphère. En effet, la valeur du facteur $F_a=1$ représente un point d'inflexion entre les deux états de génération de dômes entrelacés ellipsoïdaux: surhaussé et surbaissé. On peut alors distinguer trois intervalles, à partir desquels on peut tirer la valeur de F_a : ($1 < F_a < \infty$), $F_a=1$ et ($0 < F_a < 1$).

La génération des dômes entrelacés ellipsoïdaux est contrôlée par les fonctions suivantes:

```
1-(DLuni
2-(gen_arcpri 6 (arc_ellif 10 6 (/ 1 2))
3-(DLdif (peauf 10 6 (/ 5 2)) (gen_arcpri 6 (arc_ellif 10 6 (/ 1 2))))
4-(repere 10))
(Voir annexe-B, A-B2-«arc_elli.scm » F-«arc_ellif»)
```

La ligne-2 génère les arcs entrelacés selon les paramètres suivants:

(Gen_arcpri N_a (arc_ellif R_c N_a F_a))

La ligne-3 génère le remplissage selon les paramètres suivants:

(DLdif (peauf R_c N_a F_a (Gen_arcpri N_a (arc_ellif R_c N_a F_a)))

On substitue à chaque paramètre sa valeur, le programme retournera un dôme ellipsoïdal correspondant à ces paramètres.

Par exemple: dans le cas où $F_a=1/2$ et $N_a=8$, le programme génère le dôme ellipsoïdal nombre-8, voir la figure-74.

D'après le tableau ci-dessous, le facteur Fa prend sa valeur des trois principaux intervalle: $(1 < Fa < \infty)$, $Fa=1$ et $(0 < Fa < 1)$.

1- $(1 < Fa < \infty)$: génère des dômes entrelacés surbaissés.

2- $Fa=1$: génère des dômes entrelacés demi-sphériques.

1- $(0 < Fa < 1)$: génère des dômes entrelacés surhaussés.

Quant au paramètre Na , il peut prendre n'importe quelle valeur paire et supérieure à six. Le remplissage des dômes est contrôlé par les mêmes paramètres Fa et Na . Ce qui permet des déformations homogènes des arcs et du remplissage.

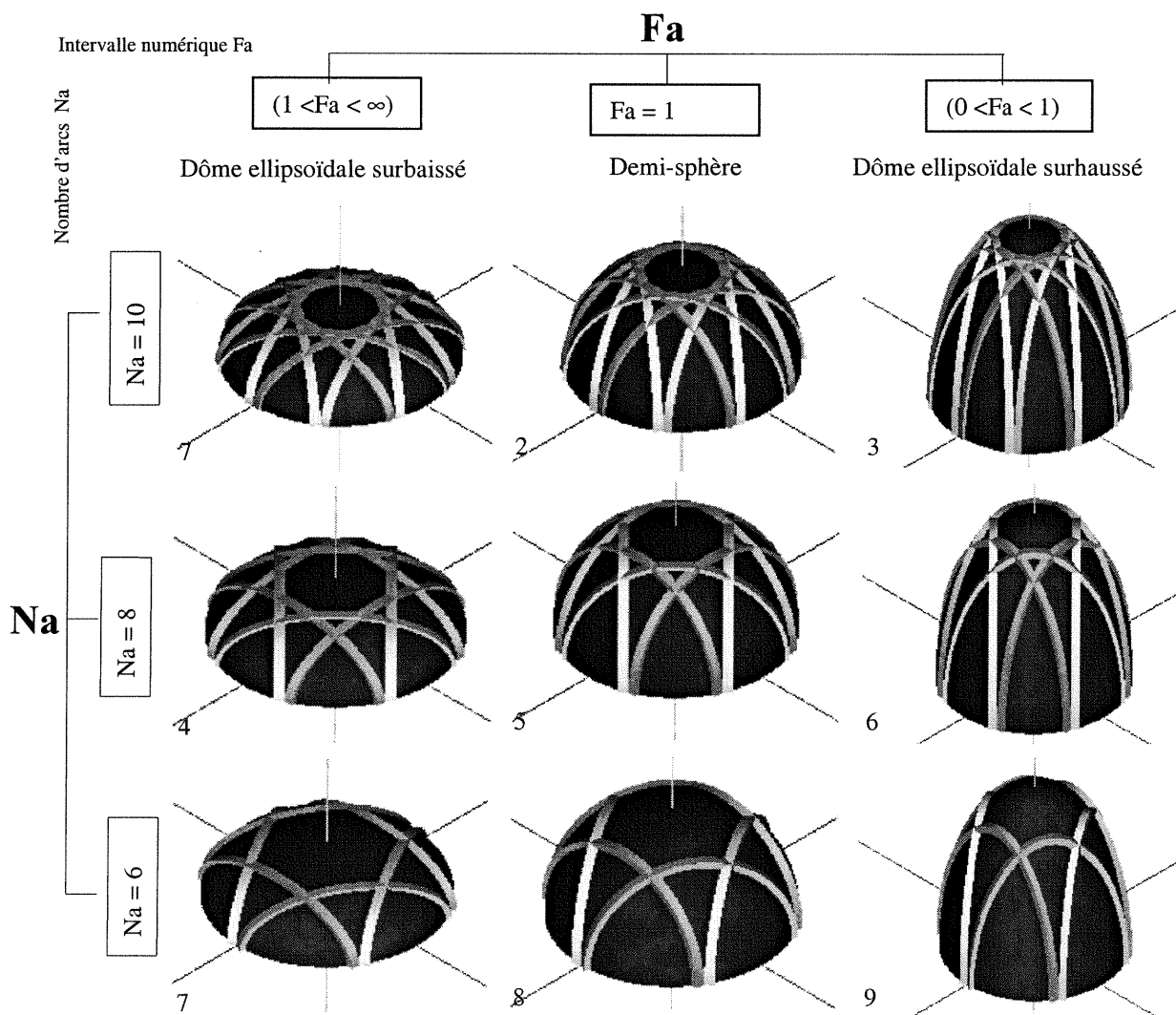


Figure-74. Les principaux types de dômes entrelacés, elliptiques générés selon les paramètres Fa et Na .

4-Insertion d'un dôme entrelacé dans un contexte architectural

Les deux programmes, «Prog_pointu» et «Prog_ellipse», peuvent générer une large série de dômes entrelacés et plusieurs types d'arcs pointus, croisés, brisés et elliptiques surhaussés et/ou surbaissés. On peut considérer ces programmes comme une bibliothèque numérique générant un ou plusieurs types d'objets géométriques ainsi que des arcs et des dômes entrelacés selon des paramètres choisis. Les arcs ou les dômes générés par ces programmes peuvent être exportés et insérés dans un nouveau contexte architectural. Celui-ci est constitué de différents éléments architecturaux dont le dôme entrelacé fait partie (figure-75). En fait l'importation et/ou l'exportation d'un objet signifie le changement de combinaison d'un groupe de fonctions représentées informatiquement dans le cadre du programme. Par exemple, nous modélisons des groupes d'éléments architecturaux représentés par un contexte architectural plus complexe. Un des éléments de ce contexte est un dôme entrelacé pointu. Celui-ci est déjà modélisé et codé dans le cadre des programmes «Prog_pointu» et «Prog_ellipse». Il suffit d'exporter ce dôme et de l'insérer dans le contexte architectural en cours de modélisation. Cette opération implique la combinaison des fonctions du dôme avec les fonctions du contexte, permettant de générer le dôme dans l'ensemble du contexte.

En résumé, les fonctions des deux programmes sont organisées dans le cadre d'une base de données (une bibliothèque numérique) permettant d'exporter et/ou d'importer des fonctions. Il suffit d'unifier les fonctions générées et de les faire exécuter par la fonction génératrice du système SGDL.

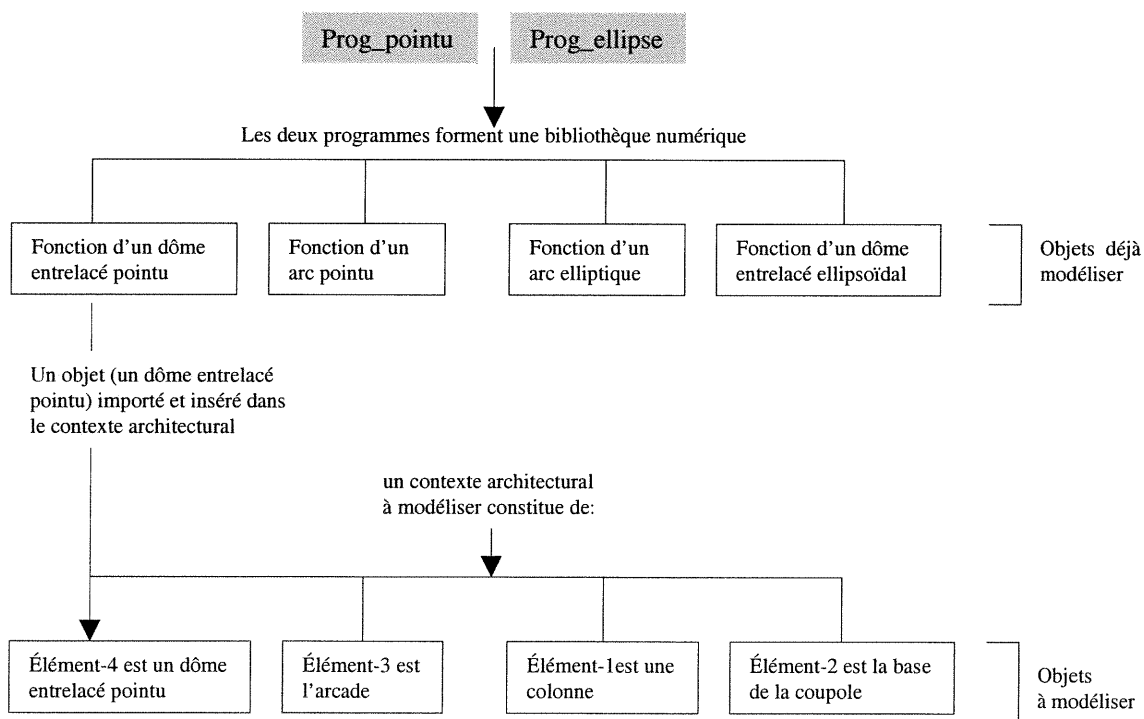


Figure-75. Rôle de la bibliothèque numérique dans la modélisation d'un objet.

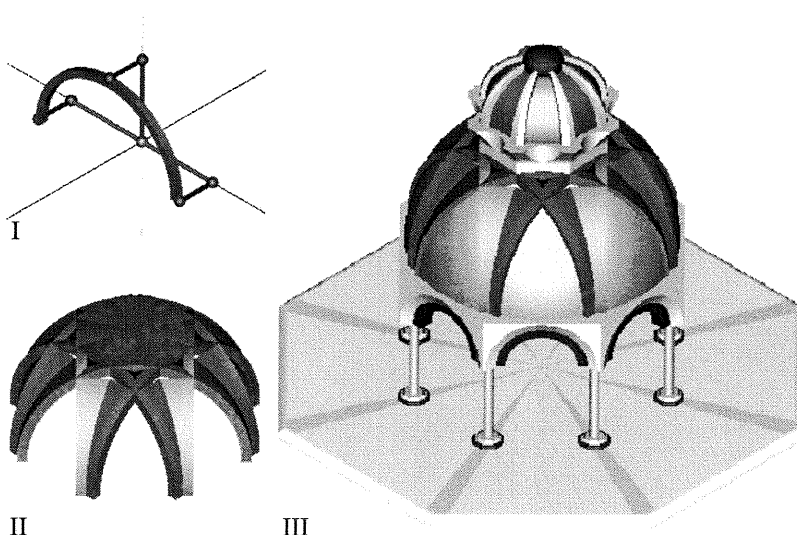
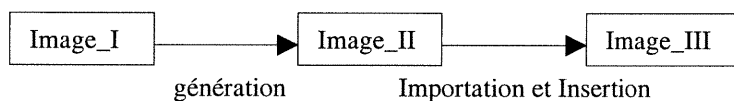
4.1-Construction du contexte architectural

Comme on l'a vu, la base de données des programmes constituent l'ensemble des fonctions dont chacune est représentée architecturalement par une unité (un arc, une unité de remplissage, etc.). Sur le plan analytique, l'unité est l'élément le plus simple. Dans notre cas, un arc généré par l'un des deux programmes précédents, représente une unité. L'ensemble des arcs est un groupe d'unités formant un objet plus complexe (un dôme entrelacé). L'ensemble de l'unité peut être identique et/ou différent selon le contexte architectural. Par exemple la figure-76-I-II-III montre comment une unité peut être intégrée dans un ensemble. L'image_I représente un arc en demi-cercle généré par le programme «prog_pointu». Celui-ci peut aussi générer l'arc de l'image_I Na fois en combinant cette dernière avec une fonction récursive afin de créer un ensemble d'arcs représentant un dôme entrelacé (voir l'image_II). Ce dôme peut être un ensemble et une unité similaire. C'est un ensemble, puisqu'il est constitué d'un groupe d'arcs. Il est une unité puisqu'il est représenté dans le cadre du programme «prog_pointu» comme une unité indépendante. Ce qui signifie que l'on peut importer ce dôme et l'insérer dans un contexte architectural (figure-76-III). Ce dernier se constitue de deux groupes d'éléments:

1-Le premier groupe sont des éléments importés. Le dôme entrelacé fait partie de ce groupe et est importé de la bibliothèque numérique, générée par le programme «prog_pointu».

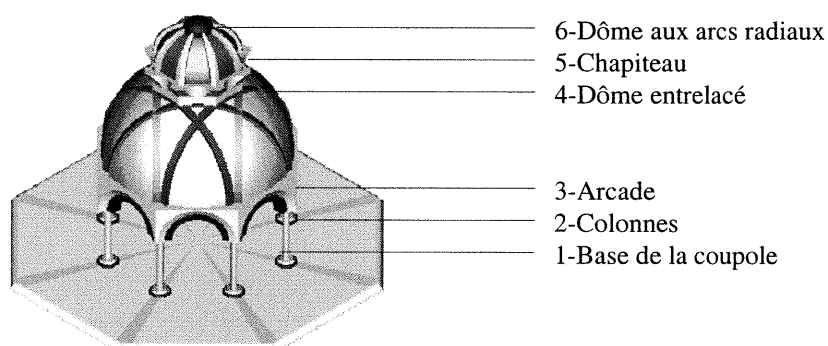
2-Le deuxième groupe est non disponible dans la bibliothèque numérique, il faudra donc le modéliser. Ce qui permet d'enrichir la bibliothèque numérique par de nouveaux éléments pouvant être utilisés à leur tour dans la construction d'autres objets architecturaux.

Figure-76. l'insertion d'une unité et ou d'un ensemble (contexte architectural)



La coupole de la grande Mosquée de Cordoue est un bon exemple de l'insertion de dômes entrelacés dans un contexte architectural. Cette coupole forme une série d'éléments architecturaux qui se superposent selon la règle géométrique du dôme entrelacé représentant une partie principale de la coupole. De par le nombre des colonnes, la forme polygonale de sa base, le nombre d'arcs du dôme du chapiteau et de la dalle sur laquelle la coupole repose, la règle du dôme est généralisée sur tous les éléments de la coupole. Cette dernière est constituée d'éléments architecturaux variés. Un de ces éléments est le dôme entrelacé formé de huit arcs reposant sur une base octogonale et couvrant un plan en forme carrée. Ce dôme peut être importé de la bibliothèque numérique du programme «prog_pointu» et inséré dans la structure de la coupole de Cordoue. Concernant les autres éléments de la coupole non-disponibles dans la bibliothèque numérique (la base, les colonnes, le chapiteau, l'arcade et le dôme aux arcs radiaux), ils seront classifiés par unité et subdivisés selon une hiérarchie facilitant la construction de leurs fonctions. La modélisation de ces éléments correspond à la règle d'assemblage des dômes entrelacés. L'Annexe-Coupole montre les démarches et les étapes de modélisation de ces éléments.

Figure-77. Les éléments du contexte architectural



Il est important de mentionner que l'exportation, l'importation et/ou l'insertion signifie: choisir et unifier les fonctions ou les algorithmes que l'on veut générer dans le cadre d'une fonction génératrice. Nous permettant de les interpréter comme résultats numériques ou images. En conclusion, le résultat de l'insertion d'un objet dans un contexte passe par la combinaison des symboles (les noms de fonctions et leurs paramètres). Ces derniers représentent des codes informatiques organisés dans le cadre d'un programme issu du même répertoire (voir annexe-C, page-xl).

L'objet inséré dans un autre contexte peut être mis à jour, à partir soit de sa fonction-mère, soit de la fonction génératrice. Grâce à la formulation homogène lambda-calcul, on effectue des calculs aussi bien sur les valeurs que sur les fonctions. Par exemple, les transformations géométriques (translations, rotations et déplacements) et les opérations booléennes appliquées à l'objet, dans le cadre de son nouveau contexte, peuvent être réalisées directement dans la fonction génératrice montrant une grande flexibilité de traitement des données. C'est dire qu'une fonction peut être intégrée comme une valeur dans les processus de calculs dans d'autres fonctions. Cette caractéristique permet une interactivité entre les fonctions et une liberté sur le plan de programmation.

4.2-Dérivation des nouvelles formes du contexte architectural modélisé

Comme on l'a vu, le contexte architectural est constitué d'éléments modélisés et importés de la bibliothèque numérique. La dérivation des nouvelles formes de coupoles dépend de l'homogénéité des paramètres et de tous les éléments modélisés et importés. Mentionnons que, lors de la modélisation des éléments des dômes entrelacés et des éléments du contexte architectural, nous avons pris en considération leur uniformité géométrique, c'est-à-dire que le système entrelacé (règle géométrique et règle d'assemblage des dômes entrelacés) est étendu à tous les éléments à modéliser. L'utilisation des différents types ou noms de paramètres n'a aucun effet sur l'opération de la génération de l'objet tant et aussi longtemps que ces paramètres gardent les mêmes sources géométriques.

Par exemple, le paramètre N_a est le nombre d'arcs utilisés dans la construction des dômes entrelacés et le paramètre T est le nombre de sommets du polygone de base de la coupole. Ces deux paramètres jouent le rôle de contrôle de l'entrelacement des arcs. N_a est équivalent à T . Tous les éléments de la coupole sont paramétrés sur les deux paramètres suivants:

R_c est le rayon du dôme entrelacé.

T est le nombre de sommets du polygone de base du dôme entrelacé.

On testera la génération de la coupole sur les données suivantes ($R_c=10$) et ($T=8, 10$ et 12). Les deux paramètres qui contrôlent la génération d'un type de l'arc de base sont:

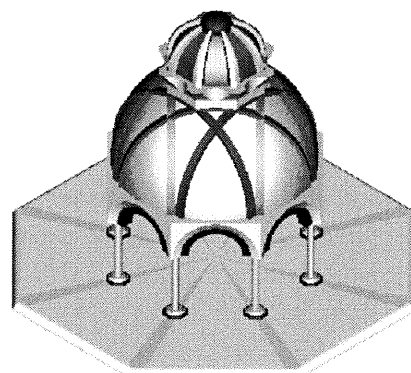
L'angle $G_a=\pi/2$ est l'angle de construction verticale de l'arc de base.

L'angle $B_e=0$ est l'angle de construction horizontale de l'arc de base.

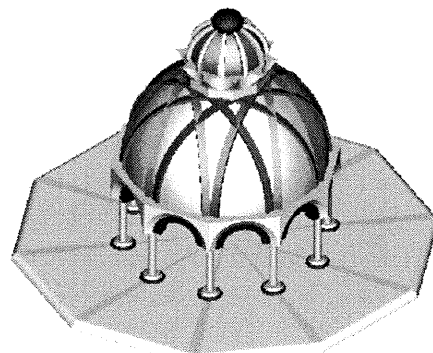
L'arc de base qui est généré, selon ces paramètres, est un demi-cercle.

On remarque que toutes les nouvelles formes ainsi générées respectent la règle géométrique de dômes entrelacés (la règle d'un dôme entrelacé est mentionnée au paragraphe 2.2.C chapitre-II).

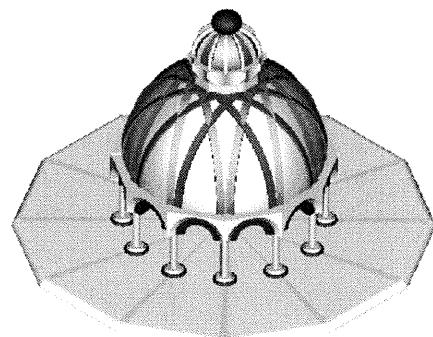
On remarque que le petit dôme aux arcs radiaux diminue, lorsque sa base polygonale diminue, ceci étant le résultat de l'entrecroisement des arcs entrelacés. Si (T) augmente graduellement, la distance entre l'arc de base et le centre du dôme diminue. Ce qui explique la diminution de la dimension du dôme aux arcs radiaux ou le chapiteau. (figure-78)



a-T = 8



b-T = 10



c-T = 12

Figure-78. La dérivation des nouvelles formes de la coupole (le contexte architectural)

4.3-Configuration surfacique du dôme entrelacé (le remplissage des dômes)

La combinaison des fonctions est une autre méthode pour générer de nouvelles formes. On s'appuie sur l'interactivité de la structure algorithmique afin de générer différentes configurations de coupoles dont les remplissages du dôme entrelacé sont de base sphérique et/ou cylindrique. L'unification des fonctions du dôme entrelacé de la coupole et du remplissage sphérique produisent la coupole I (figure-79-I). De la même façon on peut avoir une coupole dont le remplissage du dôme entrelacé est de base cylindrique II (figure-79-II). Cette variété surfacique des remplissages peut couvrir une bonne partie des choix de formes de remplissage lors de la construction des coupoles. Il est possible aussi d'utiliser un remplissage mixte, c'est-à-dire que les deux genres de surface, cylindrique et sphérique, sont utilisés dans le même dôme entrelacé. Donc, l'inventorisation possible des nouvelles configurations du remplissage des dômes entrelacés dépend des différentes fonctions des dômes et des remplissages à être regroupées.

(pour la modélisation des remplissages des dômes entrelacés voir Annexe-Coupole, paragraphe-7, page-xiv, xv et xvi).

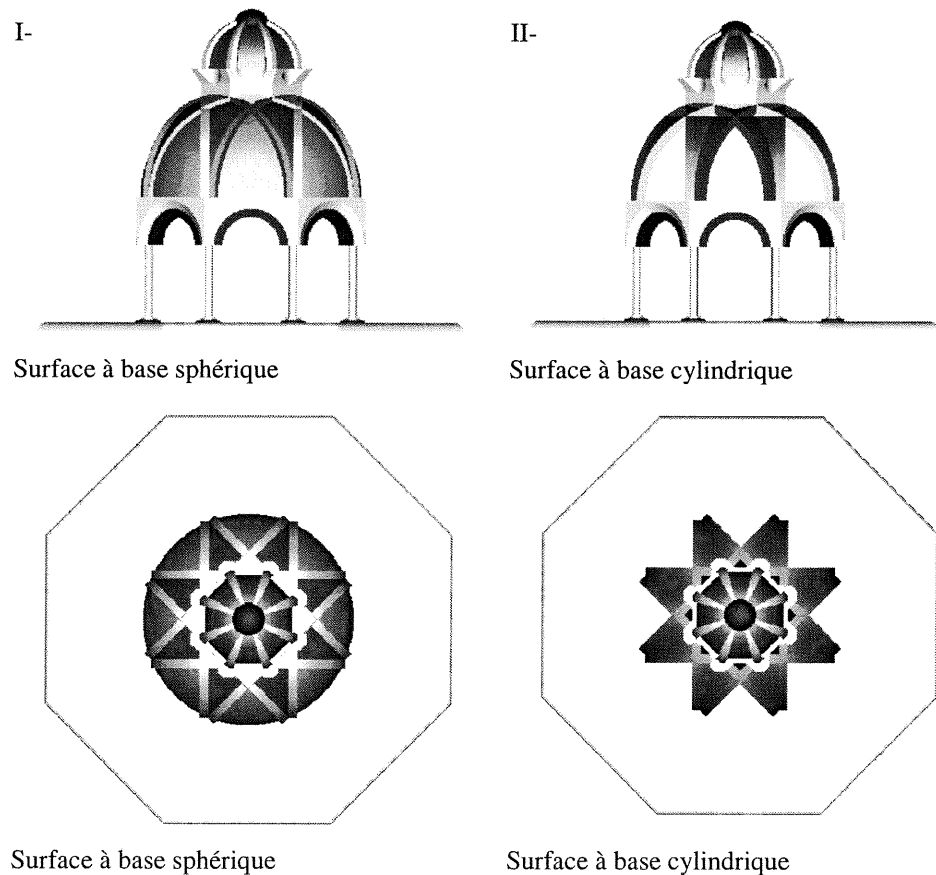


Figure-79. La configuration du remplissage de la coupole

Conclusion du chapitre-IV

La signification scientifique de l'expérimentation porte sur l'examen et la validation de l'hypothèse annoncée au début de la recherche. L'expérimentation consiste à faire ressortir les relations fonctionnelles entre les variables en assurant un contrôle systématique sur leur fonctionnement. Du point de vue géométrique, la forme spatiale de l'objet peut être comprise et communiquée à partir de sa règle géométrique. Au contraire, l'architecture donne à cette forme sa valeur et/ou son style architectural. Les travaux de modélisation réalisés sur les arcs pointus, elliptiques et sur les dômes entrelacés ont impliqué la modélisation de leurs règles géométriques ainsi que leurs règles architecturales. Ces dernières ont fait apparaître certaines restrictions à la règle géométrique. Elles suscitent des formes architecturales comme cas particulier de leurs représentations géométriques. Partant de cet énoncé deux points principaux sont examinés dans le cadre de cette expérimentation: la géométrie de l'objet modélisé et sa validité géométrique dans le domaine de l'architecture. L'expérimentation a révélé aussi que l'outil informatique peut être considéré comme l'outil principal dans l'opération de modélisation; c'est ce qui a suscité un renouvellement profond des méthodes de génération de nouvelles formes géométriques. Le système SGDL a permis d'encadrer l'expérimentation par des tests visant différentes tâches à même les programmes «Prog_pointu» et «Prog_ellipse», en vue d'assurer leur validité et de généraliser leurs applications.

L'exploration de formes enregistrées par le programme «Prog_pointu» a démontré que les formes géométriques des arcs pointus correspondent à la description mathématique d'un arc, subissant des discontinuités surfaciques et résultant de l'intersection de deux arcs de cercle. Ce résultat assure la conformité des formes géométriques des arcs pointus aux données mathématiques de ce type d'arcs apparaissant dans l'analyse géométrique. Les arcs elliptiques sont examinés, d'une part, sur le tracé géométrique et, d'autre part, sur la génération des différentes formes géométriques. Le test montre que les déformations de ces arcs sont parfaitement adaptées au facteur représenté numériquement par le rapport donnée du grand axe au petit axe des arcs générés. En effet, l'expérimentation montre aussi que, quelle que soit la forme géométrique de l'arc engendré, celui-ci peut être utilisé comme un arc de base dans la construction des dômes entrelacés, tout en respectant leurs règles de construction ou d'assemblage.

L'expérimentation réalisée sur deux cas particuliers (arc gothique et arc islamique) a permis d'assurer la validité externe de la recherche dans le domaine de l'architecture. Ce test a montré la possibilité de généraliser la méthode de construction d'un arc ayant une représentation géométrique et de l'étendre à d'autres domaines que la géométrie.

Ce dernier test a été effectué sur la base des données de deux programmes «Prog_pointu» et «Prog_ellipse». La combinaison des différentes fonctions a permis, d'une part, d'obtenir une grande variété d'arcs pointus, elliptiques et des dômes entrelacés et, d'autre part, d'assurer la flexibilité de leurs exportations vers d'autres contextes architecturaux. En résumé, trois tests sont réalisés dans le cadre de cette expérimentation: la conformité géométrique de l'objet modélisé à ses données mathématiques, la validité de l'objet dans le domaine de l'architecture et la flexibilité de la base de données des programmes.

Conclusion générale

La modélisation d'un objet signifie, en premier lieu, la représentation d'informations et/ou la description d'un phénomène, en vue de la construction d'un modèle et/ou de la simulation d'un comportement physique (transformations géométriques, déformations). Ces opérations sont considérées comme des moyens conduisant à la construction d'un objet ou d'un modèle, dans un environnement virtuel. Il s'agit en fait de décrire un phénomène ou un objet par ses caractéristiques géométriques et/ou physiques. Le modèle virtuel qui en résulte, représente quasiment la réalité et contient le processus de genèse de l'objet modélisé. La modélisation d'un objet nécessite de faire appel dès le début, au concept géométrique de l'objet et aux outils de modélisation (par exemple: des outils informatiques, dessins, maquettes, etc.). Dans la plupart des domaines, la modélisation d'un objet par outil informatique pose un problème majeur, surtout dans le design architectural et industriel. La représentation géométrique de l'objet et les outils informatiques utilisés dans les processus de modélisation ont été considérés comme des points de départ éclairant notre vision de la problématique de déformation et de générations de formes géométriques. En effet, la complexité de la modélisation d'un objet n'apparaît pas seulement lors de l'élaboration du modèle virtuel ou de la maquette numérique, mais aussi dans la coordination des divers types d'informations et d'actions dans le cadre d'une structure informatique permettant de générer des nouvelles formes géométriques à partir d'une forme de base. Le résultat obtenu, où la représentation de l'objet se transforme en une unité espace-temps, se trouve résumé dans la modélisation des actions qui figurent les différentes étapes de construction et l'enregistrement des différentes déformations formelles de l'objet.

Les informations et les données recueillies et analysées dans le cadre de la partie analytique ont formé les voies de communication de la forme géométrique de l'objet modélisé. L'analyse géométrique de l'objet s'est déroulée dans le cadre du cas général de la génération de nouvelles formes géométriques sans prendre en considération le style architectural et/ou la référence historique. Cela veut dire que nous avons procédé à des études purement géométriques contrôlées par la méthode procédurale, qui nous a permis de produire une large série de formes géométriques dans l'intervalle numérique. La perception géométrique des formes générées passe par une compréhension globale de leurs représentations mathématiques, permettant l'écart de cas particuliers et le traitement de données à partir des cas généraux: soit l'arc pointu et l'arc elliptique. La caractérisation des divers styles d'arcs architecturaux (gothique, islamique, lancé, elliptique, etc.) dans une représentation géométrique vise à généraliser cette dernière et à élaborer un concept mathématique unique qui contrôle les divers styles d'arcs. La continuité et/ou les discontinuités surfaciques des arcs représentent deux formes géométriques générales sur lesquelles est basée le concept mathématique. L'uniformisation de sources géométriques et/ou de représentations géométriques des différents styles d'arcs a facilité la coordination de leurs données géométriques dans le cadre d'un modèle mathématique (numérique). La généralité de ce dernier est traduisible par des transformations géométriques permettant d'enregistrer une large série de déformations de l'objet contrôlées par des paramètres prédéfinis. Ainsi, tout changement formel et/ou positionnel de l'objet passe par sa représentation géométrique et/ou mathématique.

La représentation de l'objet reste dans un cadre théorique. Elle ne se matérialise qu'à partir de l'utilisation d'outils de modélisation permettant la construction d'un objet virtuel. L'informatique assure le lien entre l'objet, comme concept mathématique et sa matérialisation représentée par un modèle virtuel. Grâce aux outils informatiques, une nouvelle représentation de l'objet est créée, celle de la représentation informatique par des langages de programmation. Les outils informatiques prennent leur place dans les processus de modélisation. Assurant, d'une part, la fiabilité et la rapidité du traitement des données et d'autre part la généralité de représentations des différents types d'informations dans une structure informatique ou algorithmique unique (programme). En fait, la géométrie et la programmation collaborent, dans le cadre d'un système informatique, en vue de modéliser un objet et/ou d'établir une maquette numérique. La géométrie transfère la forme de l'objet en données symboliques pouvant être intégrées et coordonnées par des langages informatiques. Ces derniers sont considérés comme langages de communication, permettant la coordination de données dans le but de résoudre un problème prédéfini ainsi que la figuration de l'objet par un dialogue logique avec l'ordinateur. Ainsi, la puissance du système informatique utilisé demeure dans son langage de programmation et son concept géométrique. Le système SGDL, avec son langage fonctionnel Scheme et son concept géométrique basé sur la géométrie projective, a permis de traduire la base théorique de l'objet (données géométriques, graphiques, structurales et/ou historiques) en deux représentations informatiques (les deux programmes «Prog_pointu» et «Prog_ellipse»), assurant ainsi la validation de l'hypothèse, c'est-à-dire l'adoption d'un système informatique permettant la modélisation d'actions et la liberté de transformations géométriques.

L'exploration des formes géométriques générées par «Prog_pointu» et «Prog_ellipse» et leur exploitation en géométrie et/ou en architecture ont impliqué, dès le début, la réalisation d'une série de tests montrant les différentes tâches pouvant être effectuées par les deux programmes. Ces tests se sont déroulés dans un contexte expérimental représenté par la modélisation informatique des arcs pointus et elliptiques, en vue de montrer leurs déformations formelles et leur adaptation au système entrelacé. En fait, ces tests ont pour but d'assurer la validité de la recherche et de montrer que la méthode utilisée a pu répondre à la question posée. Selon (T. Tidafi, 1996, p. 262), «la validité de la recherche dépend ainsi de la réponse qu'une méthode peut apporter à une question de recherche posée». Il s'agit en fait de valider l'hypothèse de la recherche se basant essentiellement sur la modélisation d'actions (les transformations géométriques de l'arc de base et les processus de construction des dômes entrelacés). L'expérimentation s'est déroulée dans le cadre d'une étude comparative examinant les résultats de la génération de différents types d'arcs pointus et elliptiques et leur utilisation dans la construction des dômes entrelacés. Les tests ont démontré l'existence de relations fonctionnelles et logiques entre les variables et les paramètres contrôlant les transformations géométriques de l'objet modélisé. Sur le plan géométrique, l'enregistrement des déformations des arcs pointus et elliptiques a montré la possibilité de la génération d'une série infinie de ces types d'arcs, selon un intervalle prédéfini dans le cadre du modèle mathématique. Il est possible, dans ce cas, d'inventorier les principaux types

d'arcs dont les valeurs représentent des points d'inflexion dans l'intervalle numérique. Ce qui permet la transparence des données numériques à partir desquelles, on peut lire et caractériser les principales formes d'arcs avant même de les générer sur l'écran de l'ordinateur. On constate que les tests réalisés sur les formes géométriques de l'objet modélisé se sont déroulés dans le cadre du cas général de la génération des arcs pointus et/ou elliptiques. C'est-à-dire que les formes géométriques de ces derniers correspondent à leurs descriptions mathématiques et graphiques représentées par le modèle numérique et/ou graphique. Il s'agit, en fait, d'assurer la validité interne de la recherche dans un contexte purement géométrique.

La validité externe de l'expérimentation porte sur la généralisation des résultats de la recherche dans des domaines autres que la géométrie. En architecture, la règle architecturale d'un arc islamique et/ou gothique représente un cas particulier de l'arc pointu. En effet, on généralise une règle géométrique (celle de l'arc pointu) en l'étendant à une règle architecturale plus rigide. (arc gothique et/ou islamique). Cette généralisation montre la flexibilité de la représentation géométrique de l'arc pointu. L'adoption de cette dernière comme modèle géométrique général implique tous les cas particuliers faisant partie de la même famille géométrique des arcs pointus.

Comme on l'a vu, les deux programmes «Prog_pointu» et «Prog_ellipse» consistent en une base de données représentée par des fonctions regroupées et combinées ayant pour but la réalisation d'une tâche particulière. Cette structure de fonctions ou d'algorithmes est considérée comme bibliothèque numérique ayant but d'exporter ses éléments en vue de les insérer dans d'autres contextes architecturaux. Partant des fonctions simples, une multitude de formes géométriques peuvent être générées par différentes combinaisons algorithmiques, en partant de fonctions simples. Cette interactivité au niveau de la base des données nous a facilité la mise à jour de la structure informatique (ajouter, supprimer et/ou combiner des fonctions et/ou des nouvelles informations). Il s'agit, en fait, d'une structure algorithmique flexible organisée par le langage fonctionnel *Scheme* et basée sur la représentation géométrique des dômes entrelacés.

En résumé, la modélisation informatique de transformations formelles et positionnelles (actions) d'un objet passe principalement par ses représentations géométriques (mathématiques) qui forment une étape intermédiaire (transitoire) entre l'objet (comme concept) et sa matérialisation (modèle virtuel). Grâce à l'informatique, nous pouvons coordonner toutes les informations (l'intelligence) de l'objet dans une structure algorithmique informatique (programme) basée sur des données géométriques (modèle mathématique) et traduisible par un langage de programmation. Enfin, la validité de l'expérimentation est assurée à même le cadre de la recherche (les transformations géométriques) et par la généralisation des résultats dans un autre domaine d'application (l'architecture).

Bibliographie

- ACLAND, James., Medieval structure, the gothic vault, University of Toronto, 1972.
- ADAM, Robert., Classical architecture, London, 1984.
- ARBOTGAST., Emmanul, Modélisation automatique d'objets non polyédriques par observation monoculaire, Institut National Poly technique de Grenoble, 1991.
- AITKE, P., C programming, Sams publishing., 1995.
- BEARZLEY, Mitchell., The world atlas of architecture, Portland House, N. Y, 1988.
- BESENVAL, Roland., Technologie de la voûte dans l'orient ancien, Paris, 1984.
- CADOZ, Claude., Les réalités virtuelles, Flammarion, Paris, 1994.
- CHAMPAGNE, François., POTVIN, Louise., DENIS, Jean. Louis., BOYL, Pierre., Savoir préparer une recherche, 1990.
- COWAN, Henry. J., Architectural structures, American Elsevier N. Y., 1971.
- DYBVIG, R.Kent., The scheme programming language, Prentice-Hall, Englewood, Cliffs, N. J., 1987.
- De PAOLI, Giovanni, Modélisation architecturale et outils informatiques entre cultures de la représentation et de savoir-faire, thèse, Université de Montréal, 2000.
- EMMRICH, Davide Georges., Course in constructive geometry, morphology, University of Washington, Seattle, 1970.
- FILON, E., An introduction to projective geometry, 1995.
- FLEURY, Michel., Graphisme et géométrie, Université du Quebec, 1986.
- FRISHMAN, Martin., Hasan, UDDIN KHAN., The Mosque, Thomas and Hudson Ltd London, 1997
- GASTON, Peter C., Geometry of spatial forms, E. Horwood; press, Chichester Sussex: New York, 1983.
- GEOTHALS, Emile., Arcs-voûtes-coupoles
- HILLENBRED, Robert., Islamic architecture, New York, Columbia University, 1994.
- HJERSMAN, Peter., Dome note, Berkeley, California, 1976.
- HOAG, John. D., Architecture islamique, Paris, 1991.
- HUFFLEN, Jean. Michel., Programmation fonctionnelle en schème, Paris Masson, 1996.
- IVANOVIC, Ivan., Mechanisms for generation of planes and curves, translated, New York, Macmillan, 1964.

- JOUVEN, George., Rythme et architecture, les tracés harmoniques, Paris, 1951.
- K.A.C.CRESWELL., Early muslim architecture, Hacker art books, New York, 1979).
- LEONLOVICH, Valerian., Frames and arches, Toronto, Mc Graw-Hill, N. Y., 1959.
- LESSER, George., Gothic cathedrals and sacred geometry, Tiranti Alec, London, 1957
- MAINSTONE., Rowland., Development in structure form, Allen Lane London, 1975.
- MANGO, Cyril., Architecture byzantine, Paris, Berger Levrault, 1981.
- PARISEL, Claude., Construction géométrique planes appliquées au dessin, Université de Montréal, 1970.
- PACCARD, André., Le Maroc et l'artisanat traditionnel islamique dans l'architecture, 1983.
- SAUDIN, Michel., Coupoles, Bibliothèque des Arts, Genève, 1989.
- SAKAROUVITCH, Joël., Épures d'architecture, de la coupe des pierres à la géométrie descriptive xvie-xixe, Basel: Birkhavser Verlag, 1998.
- SAMYNE, E., Dômes et coupoles en surfaces continues, École nationale supérieure d'architecture et des arts visuele, Bruxelles 1976.
- SCHWARTZ, Abraham., Analytic geometry and calculs, 1960, Holt, Rinehart and Winston, N. Y, 1960.
- SEIDENBERG., Lecture in projective geometry, Van nostrand Princeton, N. J, Toronto, 1962.
- SMITH, E. Baldwin., The dome, a study in the history of ideas, Princeton University press, 1971.
- STIELIN, Henri., Taschen , Architecture mondiale « Islam », Paris, 1996.
- THOMAS, George., Calculus and analytic geometry, Addison-Wesley publishing company, Reading Massachusetts, 1968
- TIDAFY, Temy., Moyen pour la communication en architecture, proposition de la modélisation d'action pour la figuration architecturale, Thèse, Université de Montréal, 1997.
- Istanbul, Congrès., domes from antiquity to the present, 1988.

ANNEXE-COUPOLE

La modélisation des éléments principaux du contexte architectural

La modélisation des éléments principaux du contexte architectural:

Le contexte architectural (la coupole de la grande Mosquée du Cordoue) est constitué d'éléments importés et modélisés. Le dôme entrelacé étant une partie essentielle de ce contexte peut être importé de la bibliothèque numérique du programme (prog_pointu). Les autres éléments de la coupole sont caractérisés, de sorte qu'on puisse les modéliser et les assembler selon la règle géométrique et architecturale de construction des dômes entrelacés (figure-c.1). Le principe de modélisation est basé sur l'utilisation de quadriques dans la construction de l'unité de base. Toutefois, d'autres primitives du système SGDL peuvent être envisagées. L'utilisation de quadriques et/ou de primitives prédéfinies dans le cadre de ce système dépendent des transformations géométriques qui peuvent être appliquées à l'élément à modéliser.

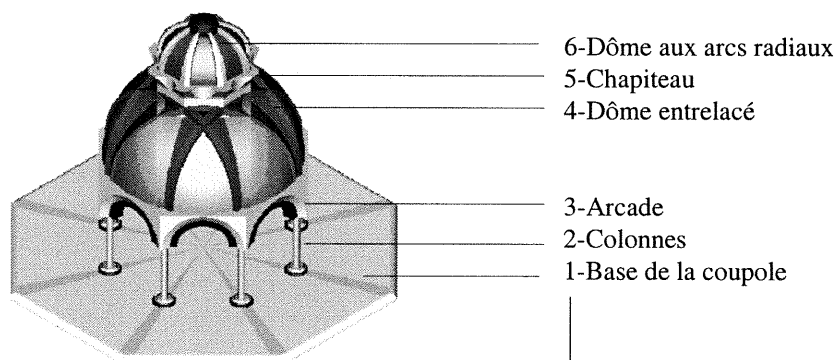
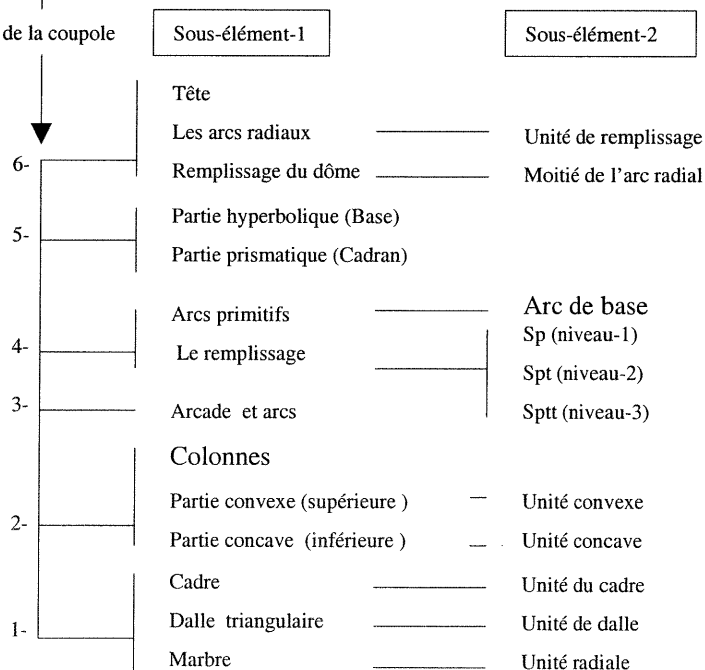


Figure-c.1. Les éléments principaux de la coupole

Les éléments de la coupole



1-La modélisation la base de la coupole

La base de la coupole est «la dalle de mosaïque» sur laquelle la coupole repose. Elle est constituée de trois éléments principaux: deux éléments radiaux et un élément périphérique. La forme de la base correspond à la base polygonale du dôme entrelacé. Les éléments radiaux sont rangés selon la division de l'angle π/Na . Le paramètre Na est le nombre d'arcs utilisés dans la construction du dôme entrelacé. Le paramètre Na correspond au paramètre T de la coupole.

Les éléments principaux de la dalle mosaïque:

- 1-Les dalles triangulaires sont représentées par la fonction (F-«dallemf»).
- 2-Les éléments radiaux entre-dalles ont des formes triangulaires, ils sont représentés par la fonction (F-«musaf»).
- 3-Le cadre ou la corniche est représenté par la fonction (F-«cornichf »). (Figure-c.2)

(pour les codes Voir l'annexe-C, A-C1, A-C2, A-C3).

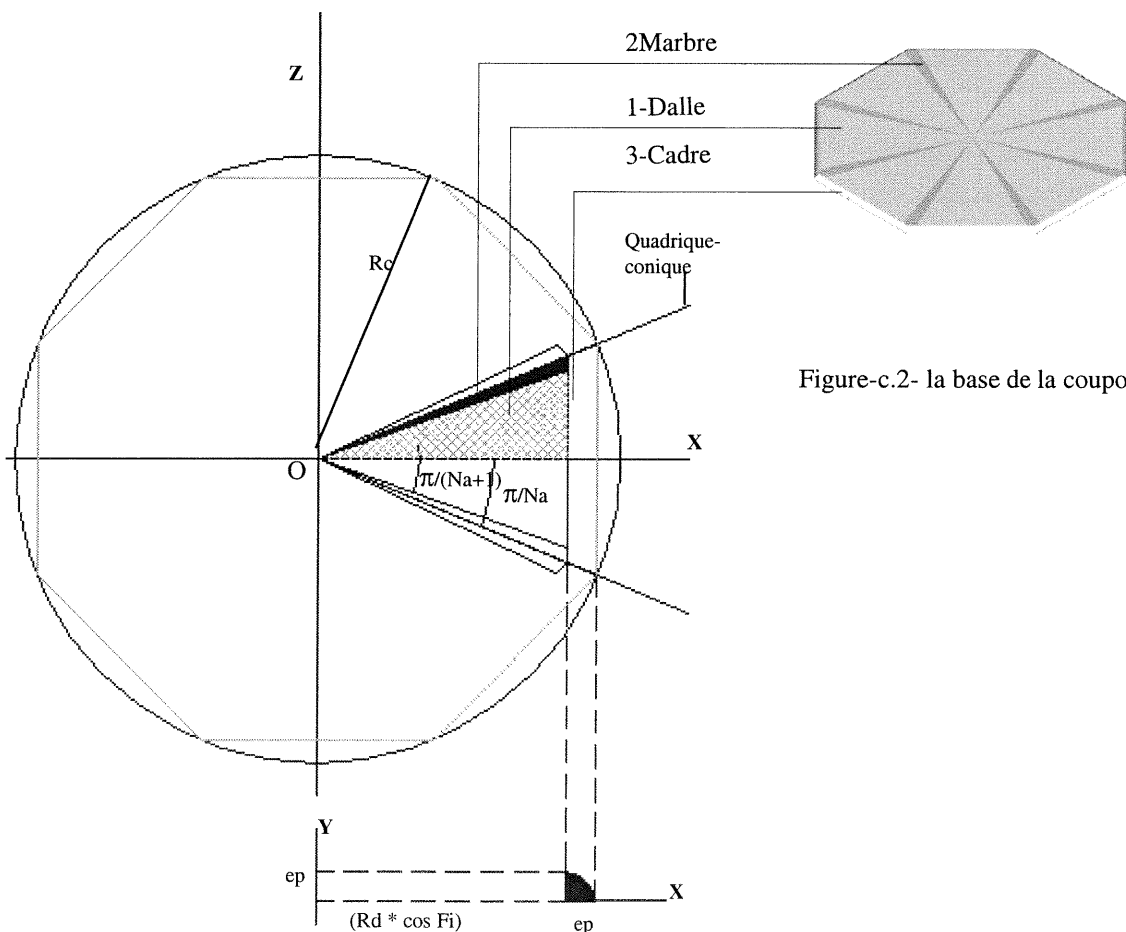


Figure-c.2- la base de la coupole

1.1-Le cadre

Afin de modéliser le cadre, nous utiliserons deux primitives du système SGDL ((GDqcyhxa) quart de cylindre + (DLformez) quadrique).

1-La primitive (GDqcyhxa) génère un quart de cylindre dont les vecteurs de son hexaèdre sont: un sommet, le centre et les trois points de fuite. (figure-c.3-a).

2-La primitive (DLformz) génère une quadrique selon des points de contrôle choisis. La quadrique obtenue prend une forme prismatique-conique. (figure-c.3-b)

L'intersection des deux primitives (1) et (2) représente l'objet modélisé ou l'unité dont les dimensions sont: (la hauteur= la largeur= ep), sa section est un quart de cercle et sa longueur L est délimitée par l'angle π/Na (figure-78).

(pour le code voir l'annexe-C, A-C1-«conrniche.scm», F-«cornichf »)

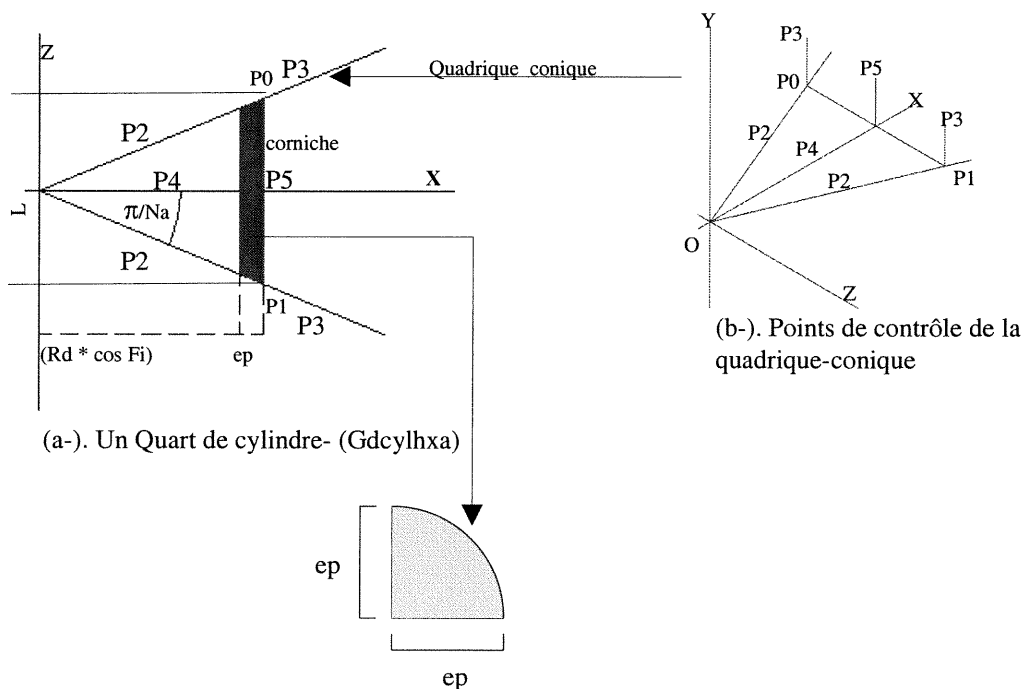
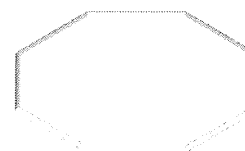


Figure-c.3. La modélisation du cadre



Primitive modélisé * matrice de la génération donne l'ensemble du cadre

1.2-La Dalle triangulaire: Cet élément est le résultat de l'intersection de trois quadriques (1, 2 et 3). (figure-c.4)

Quadrique qpl-1: (quadrique plane parallèle au plan XOZ), les points P,P0 délimitent horizontalement la quadrique sur l'axe YY`. Les points P2, P3, P4, P5 sont à l'infini.

Quadrique qpl-2: (quadrique plane parallèle au plan YOZ) les points P0, P1 délimitent verticalement la quadrique sur l'axe XX`. Les points P2, P3, P4, P5 sont à l'infini.

Quadrique conique-3: les points P0, P1 délimitent horizontalement la quadrique. Les points P2, P4 se rencontrent au point de l'origine (0 0 0) et les points P3, P5 sont à l'infini sur l'axe YY`. L'intersection des trois quadriques (DLint qpl-1 qpl-2 conique-3) donne la primitive Dalle, celle-ci est symétrique par rapport à l'axe XX`.

(Pour les codes voir l'annexe-C, A-C2-« dallem.scm » F-« dallemf »)

1.3-Marbre: est un élément radial entre-dalles. On utilise les mêmes quadriques de la fonction précédente «dallemf» avec une nouvelle configuration des points de contrôle.

(Pour les codes voir l'annexe-C, A-C3-« musa.scm » F-« musaf »)

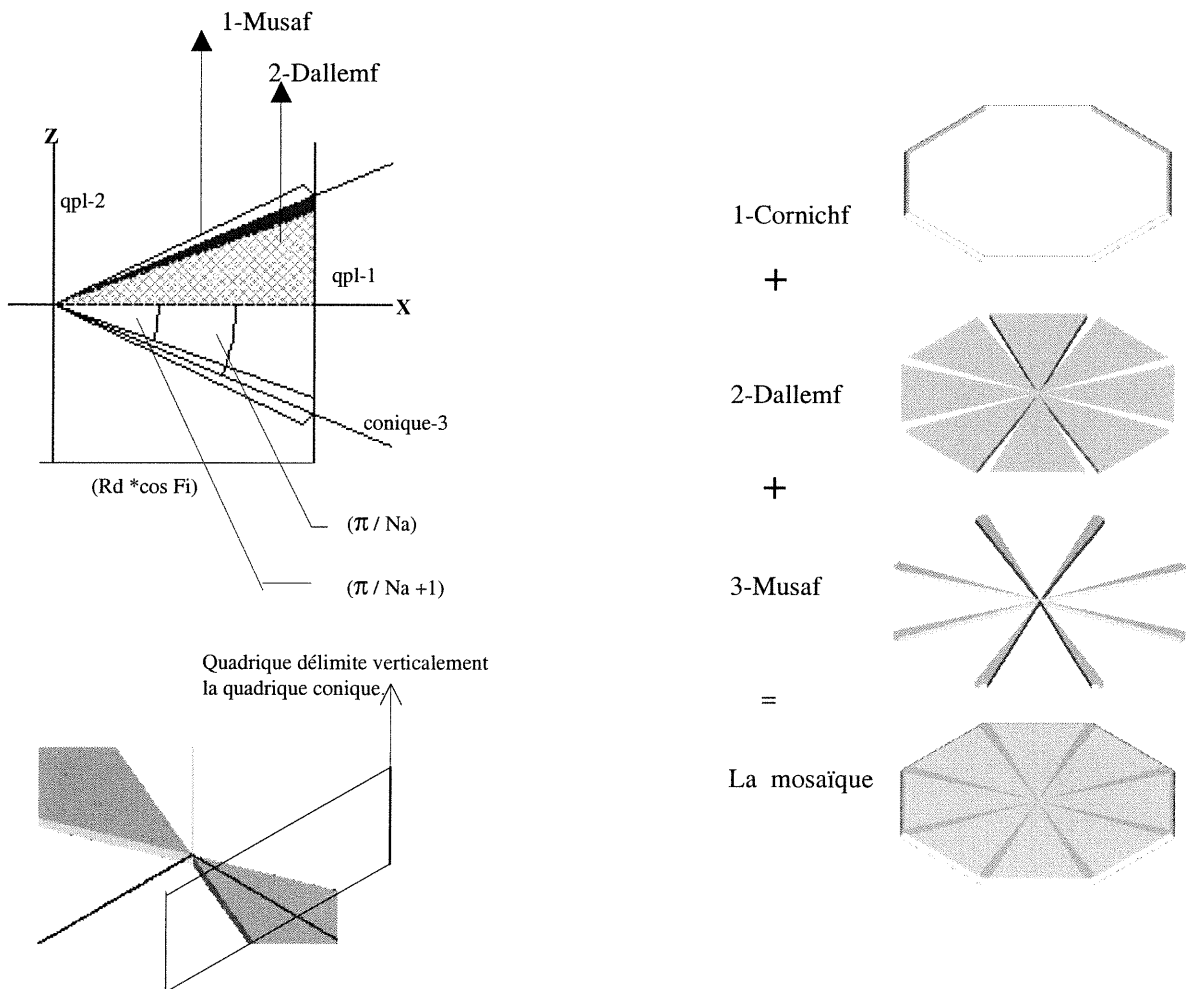


Figure-c.4. Modélisation de la dalle

L'ensemble des trois fonctions 1, 2 et 3 génère la base de la coupole ou de la mosaïque

2-Modéliser la base de la colonne

La géométrie de la surface de cet élément subit des changements dans sa courbure (figure-c.5-A). Il y a deux façons de la modéliser: ou bien on utilise l'hyperboloïde comme surface unique, ou bien on utilise un quart de cylindre et sa dualité. La deuxième méthode est préférable, puisque les deux centres et les deux rayons des deux courbes sont connus, ce qui nous permettra de contrôler la surface de la primitive. On divisera la base en deux parties: un quart de cylindre et sa dualité. (figure-c.5-C)

Partie-1: inférieure concave est représentée par un quart de cylindre.

Partie-2: supérieure convexe est représentée par la dualité d'un quart de cylindre.

La partie-1 concave:

elle est le résultat de l'intersection de la quadrique conique-q-3 et de la primitive d'un quart de cylindre (Gdcylhxa). (Dlint q-3 Gdcylhxa)

La partie-2 convexe:

est construite de deux quadriques planes q-1, q-2 et d'une quadrique conique q-3 (figure-c.5-B).

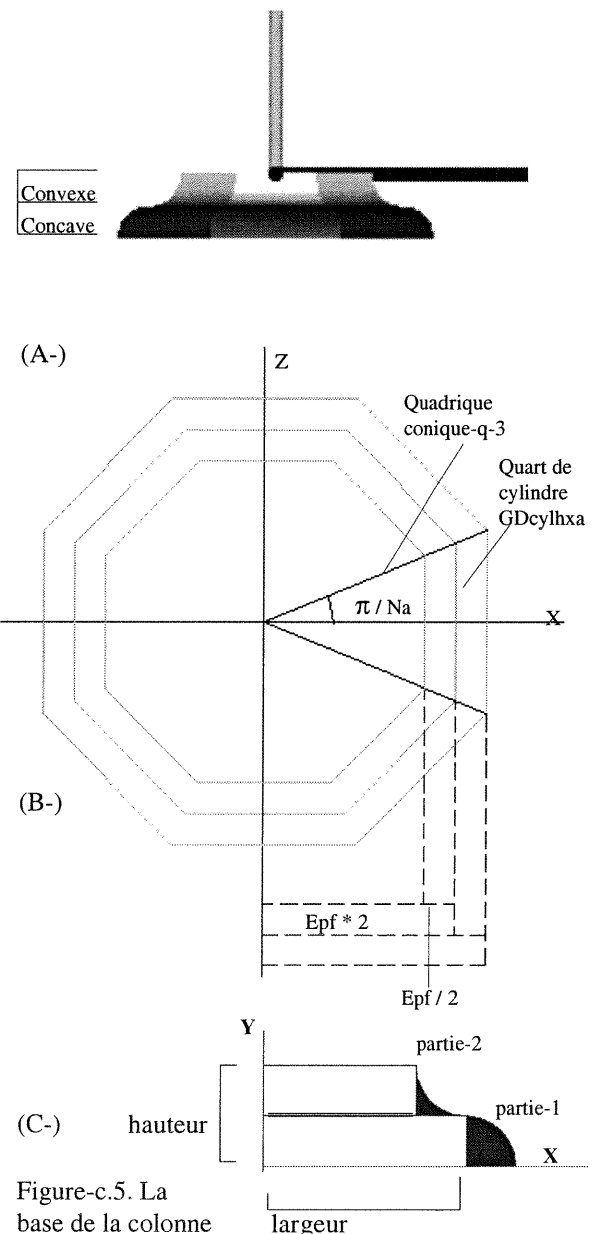
(q-1) est parallèle au plan de référence XOZ; elle définit la hauteur de la base sur l'axe YY'.

(q-2) est parallèle au plan de référence YOZ; elle définit la largeur de la base sur l'axe XX'.

(q-3) est une quadrique conique, elle est contrôlée par l'angle π/Na .

L'intersection des trois quadriques et le quart d'un cylindre produit la partie-2 de la base. Afin d'assurer la convexité de la partie-2, on enlève son coin en utilisant cette phrase de code:

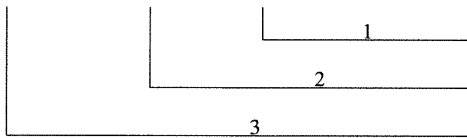
(Dlpmi (Dlint q-1 q-2 q-3 cyl)).



L'assemblage :

pour faciliter la tâche de modélisation, on construit la primitive de base au centre de la coupole; puis on produit cette primitive un nombre de fois égal au paramètre Na et on la transfère au-dessous de la colonne. Cette opération demande la création d'une fonction de construction et d'une fonction de translation. Après avoir translaté la base à sa nouvelle position, on la reproduit Na fois.

(SGmatrot (Sgmattrl (SGmatrot (primitive))))



Cette phrase de code représente la multiplication matricielle:

1-générer la primitive de base Na fois au centre du dôme pour construire la base de la colonne.

2-Translater la base à sa nouvelle position au dessous de la colonne.

3-Générer la base Na fois dans sa nouvelle position.

(Pour les codes voir l'annexe-C, A-C4-«fond.scm» F-« fondf »)

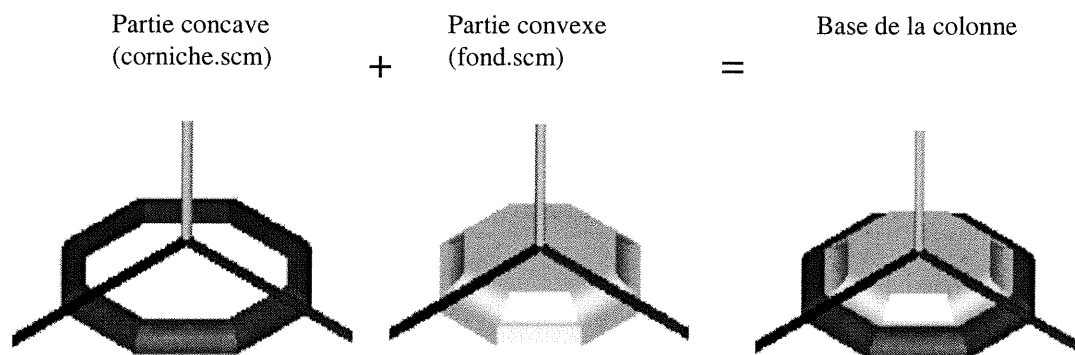


Figure-c.6. L'assemblage de la base de la colonne

3-Modéliser la colonne

La colonne est construite à partir de l'intersection de deux quadriques (DLint q-1 q-2).

1-La quadrique q-1 (cylindre): (figure-c.7) ses points de contrôle sont:

P0, P1: deux points de contact sur l'axe XX .

P2, P3: deux points de pôle, P2 est à l'infini sur l'axe YY et P3 est à l'infini sur l'axe ZZ .

P4, P5: deux points de passage, P4 point de passage sur l'axe YY et P5 point de passage sur l'axe ZZ . On peut configurer les points P4 et P5 selon la forme désirée de la colonne (figure-c.8). Par exemple:

-une colonne cylindrique: prend les points de passage suivants:

P4: (0 Y 0 0) est à l'infini sur l'axe YY , P5: (0 0 Z 1) contrôle le diamètre de la colonne sur l'axe ZZ .

-une colonne ellipsoïdale: prend les points de passage P4: (0 Y 0 1), P5: (0 0 Z 1)

-une colonne hyperboloïde: prend comme point de passage P5: (XY 0 1)

2-La quadrique q-2: représente deux plans parallèles qui tronquent la quadrique q-1 (figure-c.9). Les points de contrôle de la quadrique q-2 sont:

P0, P1: deux points de contact sur l'axe YY .

P2, P3: deux points de pôle sont à l'infini sur l'axe XX et ZZ .

P4, P5: deux points de passage sont à l'infini sur l'axe XX et ZZ .

(Pour les code voir l'annexe-C, A-C5-« colonne.scm » F-« colonref »)

Figure-c.7. (Quadrique q-1)- quadrique de la construction de la colonne (cylindre)

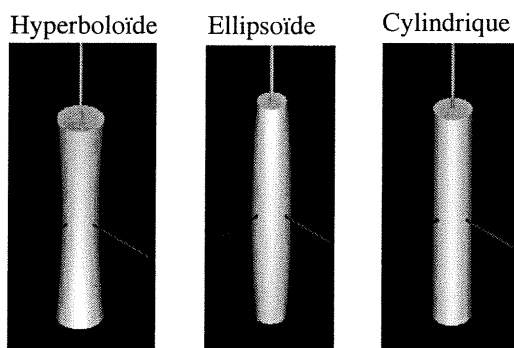
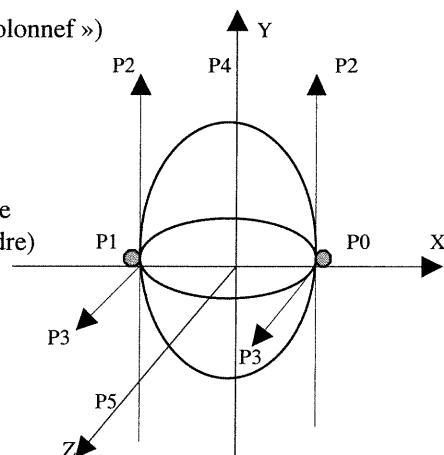


Figure-c.8. Les trois types de colonnes

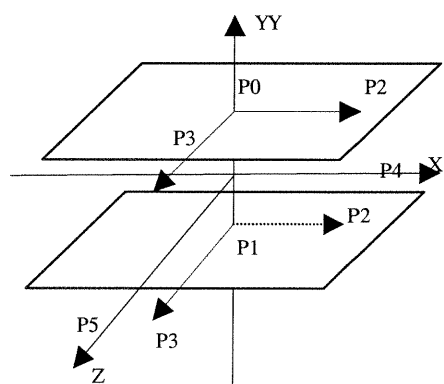


Figure-c.9. (Quadrique q-2), délimite la colonne

4-Modélisation de l'arcade

4.1-L'arcade: L'arcade est la base sur laquelle le dôme entrelacé s'installe. La projection de l'arcade sur le plan de référence XOZ, a une forme polygonale. Cette base peut être un hexagone, un octogone ou diverses autres formes polygonales régulières dont le nombre de côtés est contrôlé par le nombre des sommets du polygone de base du dôme entrelacé N_a .

La primitive de l'arcade est un prisme triangulaire. Elle est construite de deux quadriques planes q-1, q-2 et d'un quadrique conique q-3 (figure-c.10).

q-1 est parallèle au plan de référence XOZ.

q-2 est parallèle au plan de référence XOY.

q-3 est une quadrique conique.

L'intersection des trois quadriques produit un trapézoïdre. Celui-ci est vidé par un cylindre afin de créer un berceau, tronqué de l'intérieur par un cône dont le centre est le point C (figure-c.11). La phrase de code suivante résume l'opération de la construction de l'arcade:

(Dlpmi (Dlint q-1 q-2 q-3) Gdcylseg Gdconseg) (Voir l'annexe-C, A-C7-«prism.scm » F-«prismf »)

Figure-c.10. Les trois quadriques formant l'arcade

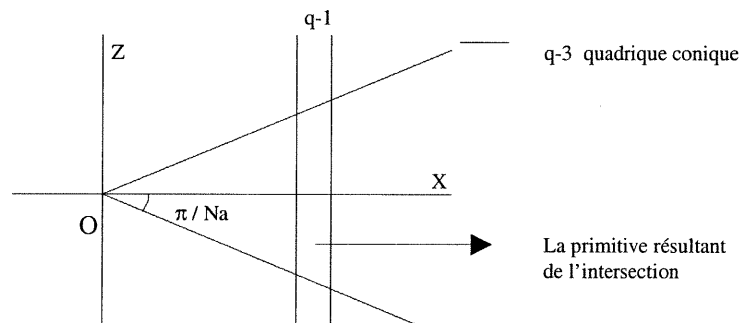
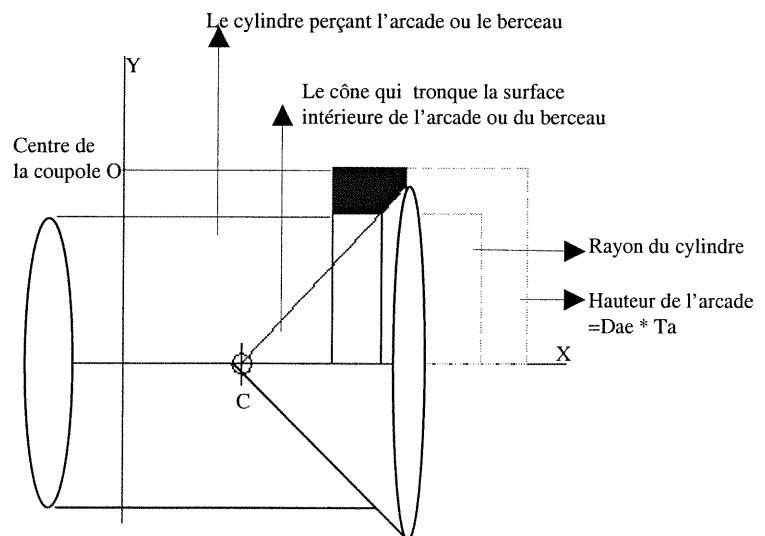


Figure-c.11. Illustration de la construction de l'arcade



La figure-c.12 montre la façon de construire la primitive prism. Le point O est le centre de la coupole. Le point C est le centre du cône qui tronque l'arcade.

4.2-L'arc de l'arcade:

L'arc est généré par la phrase du code suivant (Dlpmi Gdcylseg-E q-2 GDcylseg_-I) Gdcylseg-E est un cylindre vidé de l'intérieur par un autre cylindre Gdcylseg-I. L'anneau qui est le résultat de cette opération est coupé par la quadrique q-2 afin d'obtenir l'arc de l'arcade.

(Pour les codes voir l'annexe-C, A-C8-«arcad.scm » F-«arcadf »)

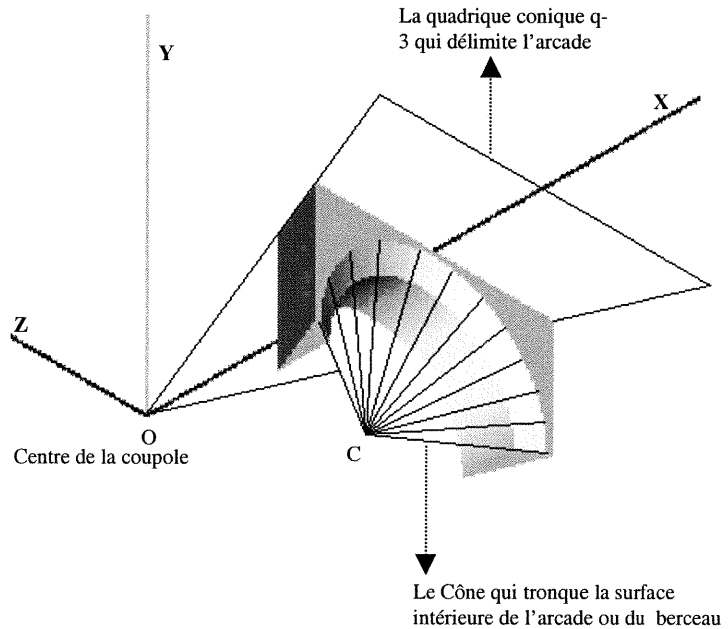
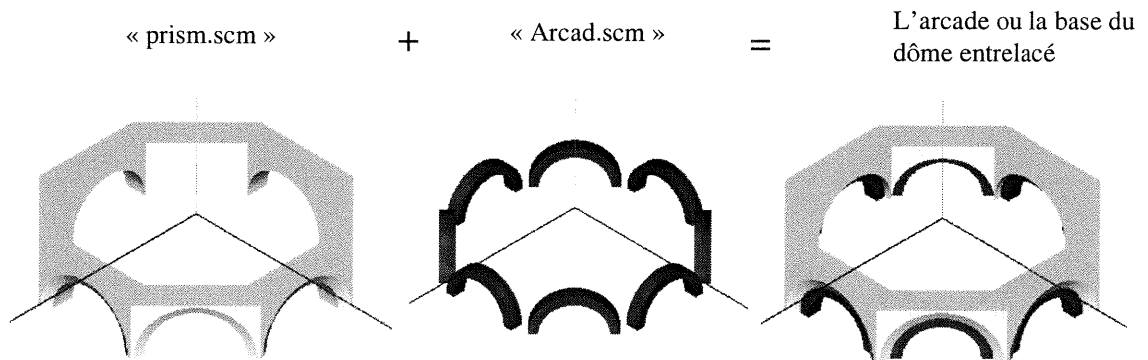


Figure-c.12. Modélisation de l'unité primitive de l'arcade

L'assemblage de l'arcade et du prisme forment la base du dôme entrelacé.



5-Modéliser le chapiteau

Le chapiteau est la base sur laquelle le petit dôme aux arcs radiaux s'installe. Il est constitué de deux parties superposées (la base et le cadre). (figure-c.13)

5.1-La base: est une hyperboloïde délimitée par quatre quadriques
 q-c est une quadrique conique contrôlée par l'angle π/Na .
 q-p est une quadrique plane parallèle au plan vertical ZOY.
 q-t est une quadrique plane parallèle au plan horizontal XOZ.
 q-h est une quadrique hyperboloïde.

(Pour les codes voir l'annexe-C, A-C9«hyper.scm » F-«hypertf »)

la partie convexe est vidée par un cylindre Cy (figure-c.14).

(Dlimp (Dlint q-c, q-p, q-t, q-h) Cy). Cette ligne de code montre l'opération d'intersection des quatre quadriques. Le volume obtenu est vidé par le cylindre « Cy » afin d'assurer la convexité de la base .

Figure-c.13 Modéliser la base du chapiteau.

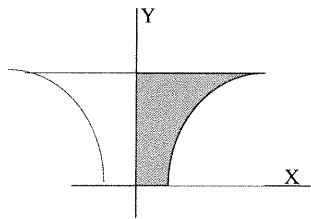
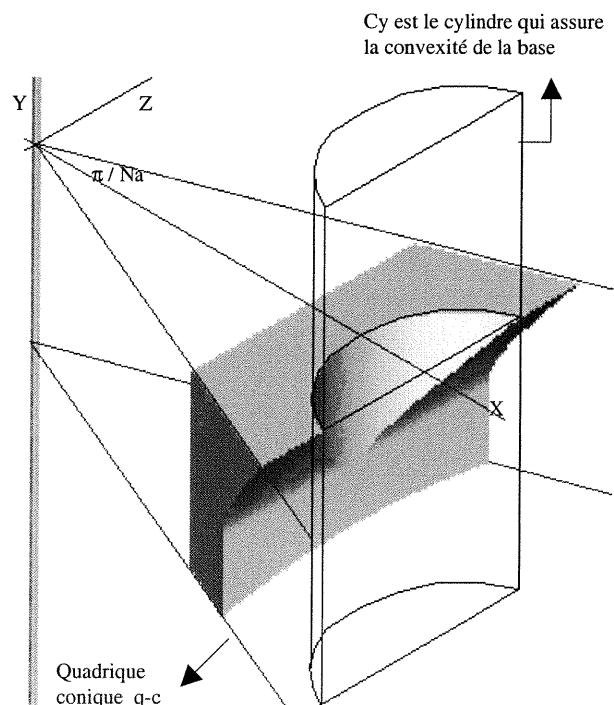


Figure-c.14. La projection latérale de la base est un quart d'un hyperboloïde



5.2-Le cadran

Il est généré par la fonction « p_base ». Les mêmes quadriques qu'on utilise dans la modélisation de la base seront réutilisés dans la modélisation du cadre ou de la p_base. La partie inférieure du cadre est vidée par l'arc de base du dôme entrelacé. (figure-c.15) (voir l'annexe-C, A-C10«p_base.scm » F-«p_basef »)

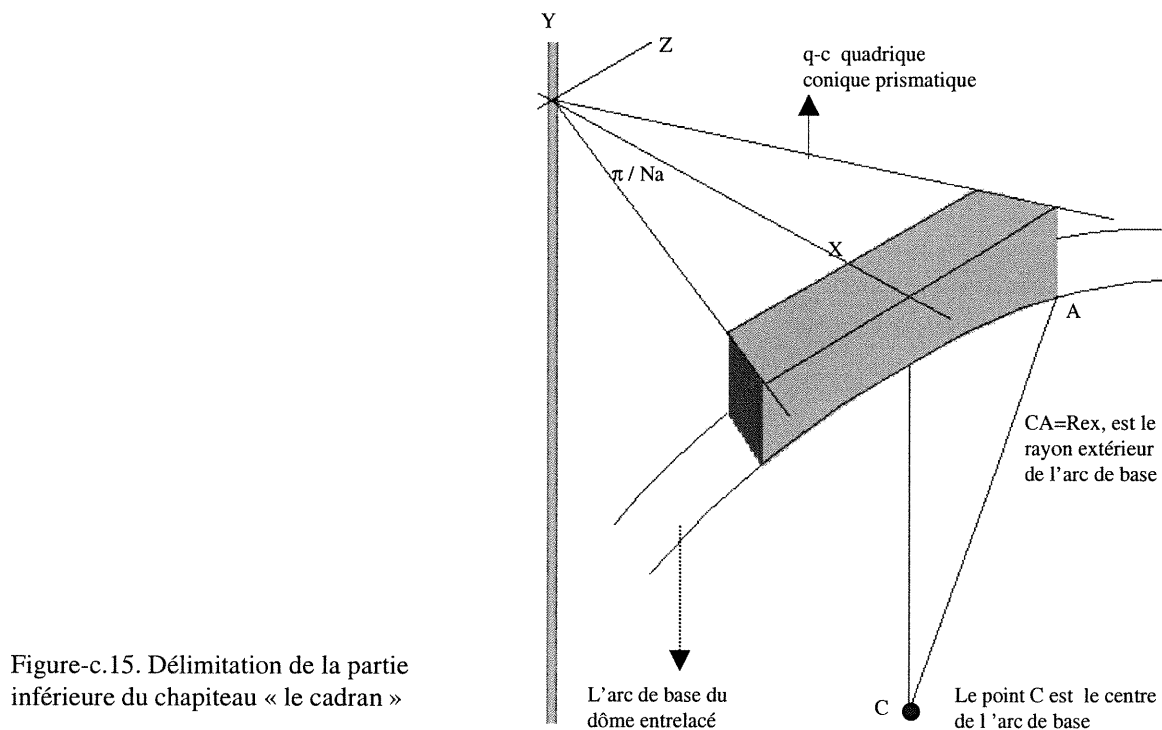
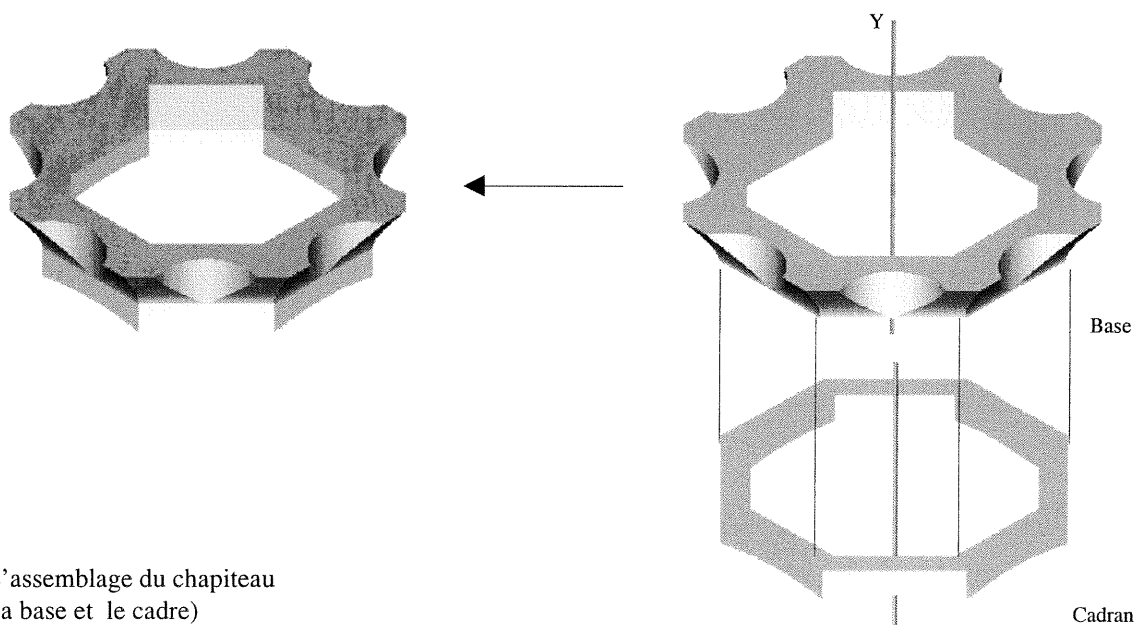


Figure-c.15. Délimitation de la partie inférieure du chapiteau « le cadran »



L'assemblage du chapiteau
(la base et le cadre)

6-Modéliser le dôme aux arcs radiaux

Le dôme aux arcs radiaux est constitué de trois éléments principaux: les arcs radiaux, le remplissage du dôme et la tête (figure-c.16).

6.1-Les arcs radiaux

L'arc de base est assemblé de deux arcs. La primitive utilisée dans la construction de cet arc est un quart de cylindre (Gdcylhxa). Le quart du cylindre est coupé par la quadrique rotative PV. Celui-ci est une quadrique plane et perpendiculaire au plan de référence XOZ (figure-c.16). Les points de contrôle de ce quadrique sont:

P0, P1: deux points de contact sur l'axe Z (0 0 Z 1).

P2, P3: deux points de pôle sont à l'infini sur l'axe XX' et YY' .

P4, P5: deux points de passage, P4 est à sur l'axe YY' et P5 est à l'infini sur l'axe XX' .

On tourne la quadrique PV autour de l'axe YY' en angle $\pi/Na+3$ et on coupe le quart du cylindre. On réalise une deuxième rotation en angle π/Na et on délimite l'arc de base. Ce dernier est vidé de l'intérieur par un cylindre dont le rayon égal EO (figure-c.17). La tête de l'arc est coupée par la quadrique de la tête de la coupole.

Les opérations précédentes permettent de construire la moitié de l'arc de base AOB. Les deux parties AOB et BOC de l'arc sont symétriques par rapport au plan BOY.

(Pour les codes voir l'annexe C, A-C11-«arcpri.scm » F-«arcpf »)

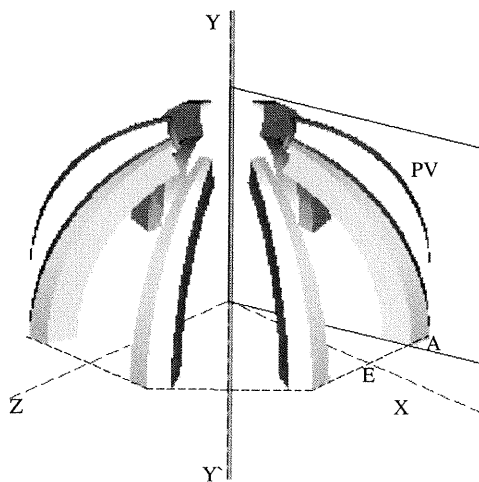


Figure-c.16. Le plan rotatif PV de la construction des arcs

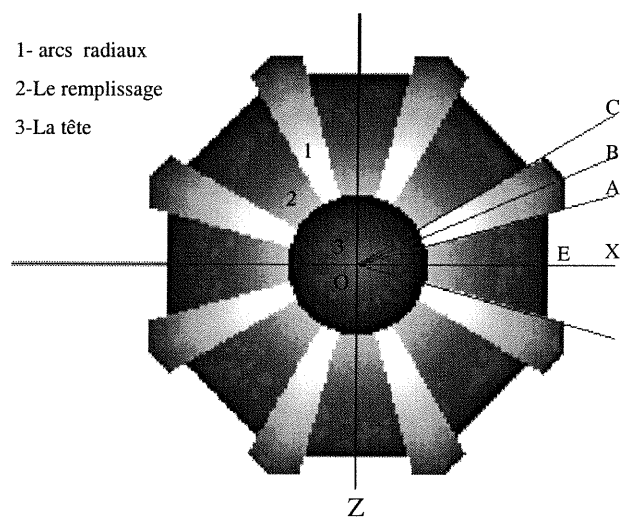


Figure-c.17. Construire les arcs du dôme

L'angle $XOA = \pi / Na + 3$

L'angle $XOB = \pi / Na$

L'angle $XOC = \pi / Na - 3$

6.2-Le remplissage: dans la construction du remplissage, on utilise un quart de cylindre (Gdcylhxa) et une quadrique conique dont les points de contrôle sont :

P0, P1: deux points de contact sont localisés sur une ligne parallèle à l'axe ZZ`.

P2, P4: Les deux vecteurs se rencontrent à l'origine (0 0 0 1).

P2 :est le point de pôle. P4: est le point de passage.

P3, P5: (P3) est le point de pôle à l'infini sur l'axe ZZ` et P5 est le point de passage à l'infini sur YY`. (figure-c.18-a)

La partie E,O,P0 représente la moitié du remplissage. Les deux parties du remplissage sont symétriques par rapport à l'axe XX`. (figure-94-b)

(Pour les codes voir l'annexe C, A-C12-«spa.scm » F-«spa»)

6.3-La tête : est une quadrique ellipsoïdale dont les points de contrôle sont:

P0 (vector X Y 0 1) point de contact.

P1 (vector X Y 0 1) point de contact.

P2 (vector 0 1 0 0) point de pôle est à l'infini sur l'axe YY`.

P3 (vector 0 0 1 0) point de pôle est à l'infini sur l'axe ZZ`.

P4 (vector 0 Y 0 1) point de passage sur l'axe YY`.

P5 (vector 0 Y Z 1) point de passage sur l'axe ZZ`.

(Pour les codes voir l'annexe C, A-C16-« tête.scm » F-« têtef»).

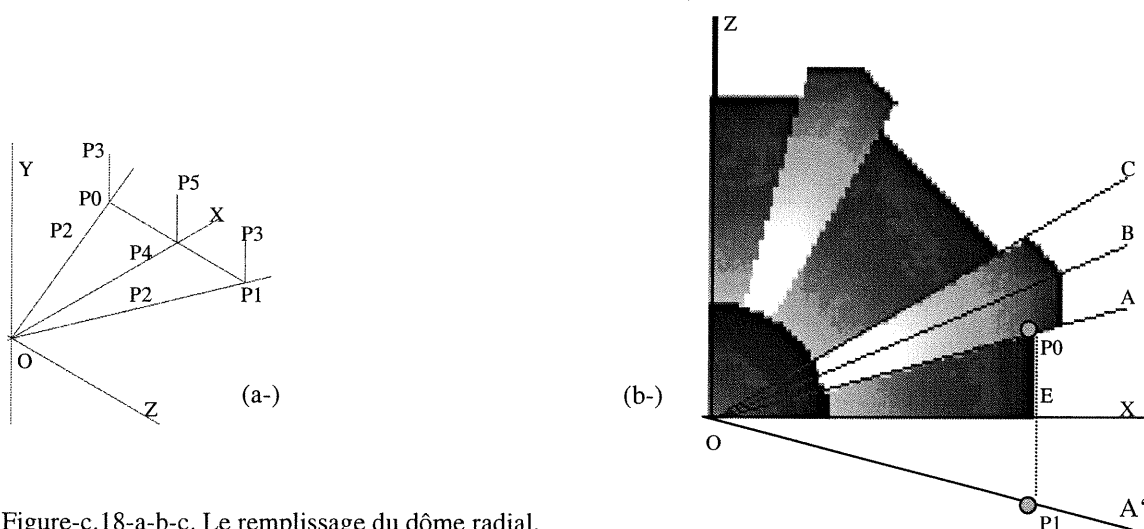
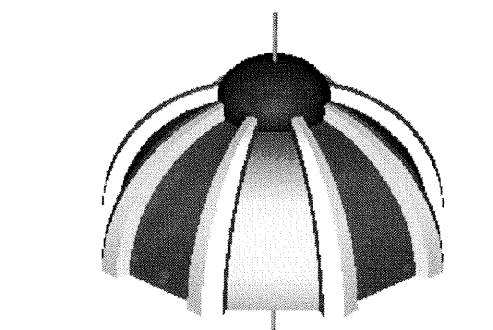


Figure-c.18-a-b-c. Le remplissage du dôme radial.

(a-) construction du remplissage

(b-) quadrique conique utilisée dans la construction



L'assemblage du dôme avec des arcs radiaux

7-Le remplissage des dômes entrelacés: nous utiliserons deux méthodes afin de remplir les espaces vides entre les arcs du dôme entrelacé. L'une basée sur l'utilisation d'une surface sphérique et l'autre basée sur l'utilisation d'une surface cylindrique.

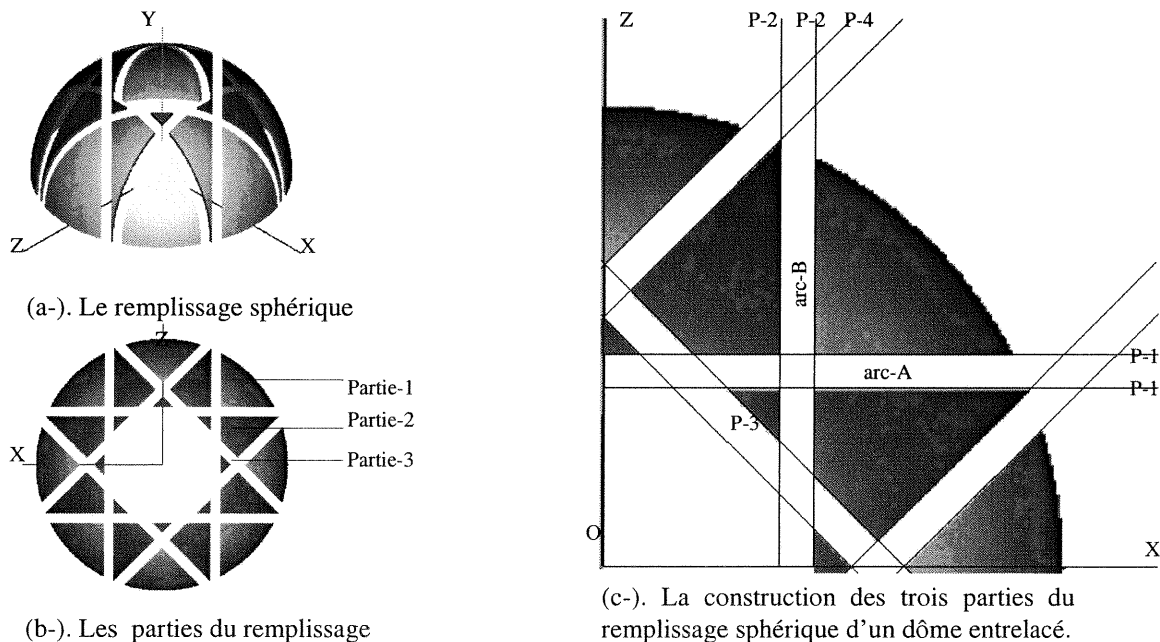
7.1-Première méthode: on utilise une demi-sphère afin de remplir les espaces entre les arcs entrelacés. Le remplissage du dôme est constitué des trois parties, lesquelles sont coupées ou taillées à partir d'une demi-sphère (figure-c.19-a-b).

Partie-1: C'est un triangle sphérique (figure-c.21). Il est construit à partir d'une demi-sphère vidée à l'intérieur. Il est coupé par deux quadriques P-1 et P-2 qui sont perpendiculaires au plan de référence XOZ. La quadrique P-1 est parallèle au plan de référence XOY et elle délimite la face externe de l'arc A. La quadrique P-2 est parallèle au plan de référence ZOY et il délimite la face externe de l'arc-B. Un autre quadrique q-h enlève la moitié inférieure de la sphère. On peut résumer cette opération par cette phrase de code: (Dlpmi Gdsphdis_ex Gdsphdis_in P-1 P-2 q-h). (Pour les codes voir l'annexe-C, A-C13-«sp.scm » F-«sp»).

Partie-2: Dans la construction de cette partie, on utilise le même principe (figure-c.19-c). Une demi-sphère vidée à l'intérieur et coupée par quatre quadriques: P-1, P-2, P-3 et P-4. Cette phrase de code résume la construction de la partie-2. (Dlpmi Gdsphdis_ex Gdsphdis_in P-1 P-2 P-3 P-4 q-h) (Pour les codes voir l'annexe-C, A-C14-«spt.scm » F-«spt»).

Partie-3: on utilise les mêmes quadriques déjà utilisés dans la construction des deux parties 1 et 2 (figure-c.19-c). Cette phrase de code résume la construction: (Dlpmi Gdsphdis_ex Gdsphdis_in P-1 P-2, P-3 q-h) (Pour les codes voir l'annexe-C, A-C15-«spt.scm »F-«spt»).

Figure-c.19. Le remplissage sphérique



7.2-Deuxième méthode: on utilise un demi-cylindre afin de remplir les espaces entre les arcs. Trois parties remplissent l'espace du dôme entrelacé. Le demi-cylindre utilisé dans la construction est une quadrique dont les points de contrôle correspondent aux points de contrôle de la quadrique de l'arc de base du dôme entrelacé (figure-c.20-a).

1-(Dlpmi
2-(Dlint qdr_arc qdr_int)
3-qdr_imp plan_cuth plan_cutv).

cette phrase de code montre la construction du demi-cylindre.

La ligne-2: représente l'intersection des deux quadriques qdr_arc et qdr_int afin d'obtenir l'arc cylindrique ABCD (figure-c.20-a).

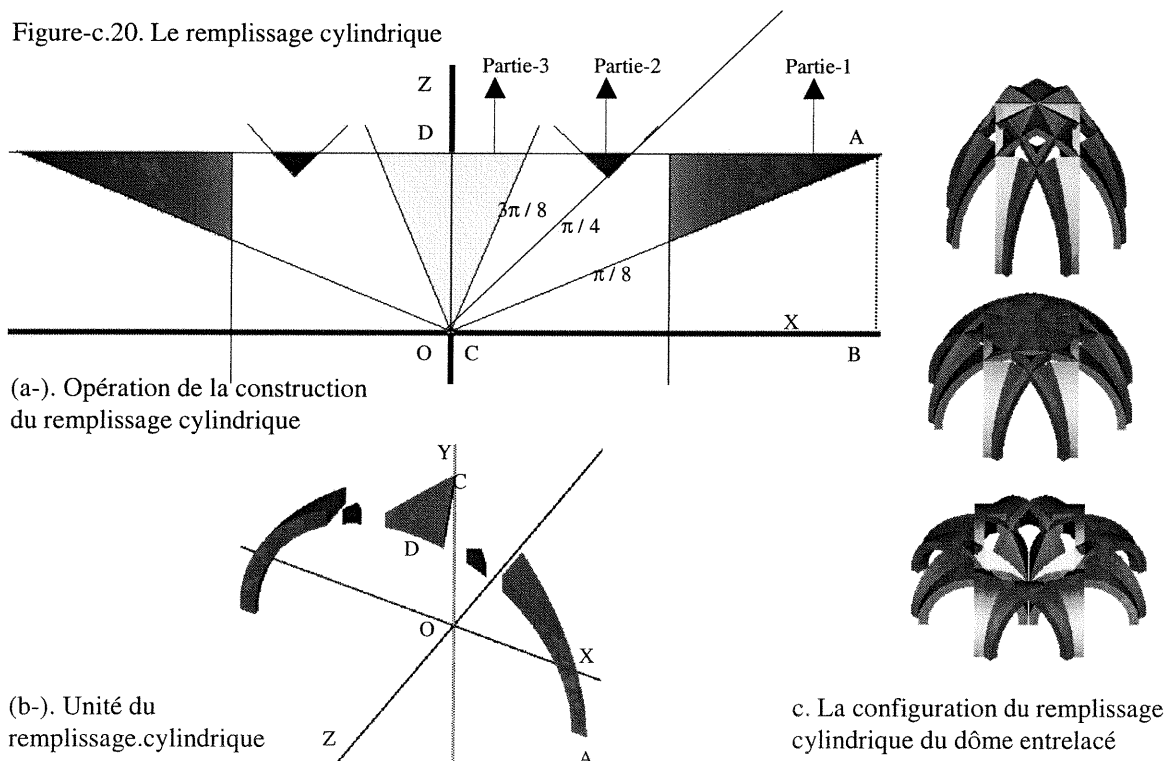
La ligne-3: représente La quadrique qdr_imp vide de l'intérieur de l'arc ABCD. Celle-ci est coupée horizontalement par le plan (plan_cuth) et verticalement par le plan (plan_cutv). (Pour les codes voir l'annexe-A, A-A3-«ramli.scm »F-«ramlif»).

L'arc ABCD représente la moitié du remplissage. Il est constitué de trois parties partie-1, partie-2 et partie-3 (figure-c.20-a). Pour construire ces parties, on coupe l'arc ABCD par un plan perpendiculaire au plan de référence XOZ. Ce plan tourne autour de l'axe YY' selon les trois angles suivants $\pi/8$, $\pi/4$ et $3\pi/8$.

Le remplissage est symétrique par rapport au plan de référence ZOY (figure-c.20-b).

Le remplissage du dôme est constitué de trois morceaux d'une quadrique qui prend la forme de la quadrique de l'arc de base du dôme entrelacé. Lorsqu'on change la forme de l'arc de base, le remplissage suit ce changement (figure-c.20-c).

(Pour les codes voir l'annexe-A, A-A3-«ramli.scm » F-«ramlif»).



ANNEXE -A

PROGRAMME « prog_pointu »

ANNEXE -A

PROGRAMME « prog_pointu »

Les fonctions des dômes entrelacés dont l'arc de base subit des discontinuités surfaciques. Ces fonctions sont regroupées dans le cadre du programme (prog_pointu).

Liste des fonctions

nom de l'algorithme	nom de la fonction
A-A1-algorithme (geo-dis.scm)	F-« géo-disf »
A-A2-algorithme (arc-dis.scm)	F-« arc-disf »
A-A3-algorithme (ramli.scm)	F-« ramlif »
A-A4-algorithme (Re_gout.scm)	F-« Re_gout_\$f »
A-A5-algorithme (descart.scm)	F-« repere »
A-A6-algorithme (mmx.scm)	F-« mmx_\$f »
A-A7-algorithme (outil.scm)	F-« rotf » F-« couperf » F-« pontf »
A-A8-algorithme (rotation.scm)	F-« gen_arcpri »

A-A1-algorithme (geo-dis.scm)

-Nom de la fonction (arc-dis)

-Autre: Jamil Moussa

-Date: septembre 2000

-Objectif : la fonction consiste en des données de base permettant de générer l'arc de base du dôme entrelacé avec son illustration géométrique.

-Liste des fonctions appelées durant l'exécution:

(Ploadr "descart.scm")

(Ploadr "rotation.scm")

(Ploadr "point.scm")

(Ploadr "rot.scm")

(Ploadr "outil.scm")

(Ploadr "mmx.scm")

(Ploadr "Re_gout.scm")

-Définition des paramètres utilisés dans la fonction :

;;Rc: :rayon de dôme

;;Na: :nombre des arcs générés dans la construction de dôme

;;Gama: angle de la délimitation verticale de l'arc de base

;;Be: angle de la délimitation horizontale de l'arc de base

;;Nis: paramètre du filtre de la règle gothique

;; n: nombre de divisions de l'arc islamique

;;Nag: paramètre du filtre de la règle islamique

-Définition des principales variables utilisées dans la fonction:

;;Te: angle de la position de l'arc de base par rapport à l'axe xx

;;Ep: épaisseur de l'arc de base (la section de l'arc est carrée)

;;Rex: le rayon externe de l'arc de base

;;Rin: le rayon interne de l'arc de base

;;Dc: la distance entre le centre du dôme et la face interne de l'arc de base

;;De: la distance entre le centre du dôme et la face externe de l'arc de base

;;======(Corps de la fonction « geo_disf »)=====

(define geo_disf

(lambda (Rc Na Gama Be Nis n Ngo)

(let* (

(Pi (* (SGcst_pi) 1))

(Te (/ Pi Na))

(Ep (/ Rc 12))

(Rex (* Rc (cos Te)))

(Rin (- Rex Ep))

(Dc (* Rc (sin Te)))

(De (+ Dc Ep))

;;------(Nis= règle islamique , Ngo= règle gothique) (filtrer l'angle Ga)

(Ga

(cond

((and (not (zero? Nis)) (not (zero? Ngo))) (display "error Nis or Ngo, have to be 0"))

((not (zero? Nis)) (acos (/ 1 (+ 1 n))))

((= Ngo 1) (acos (/ (tan (/ Pi 8)) (+ 1 (tan (/ Pi 8))))))

(else (+ Gama))))

```

;;------(illustration géométrique de la règle islamique)
(cyl_is (if (not (zero? Nis))
(DLatt (SDcolRGB (vector 20 150 180 255))
(GDcylseg (vector (/ Rex Nis) 1)
(vector 0 0 (/ Ep 6) -1)
(vector 0 0 (/ Ep 6) 1)))
Nis))
(cyl_dif (if (not (zero? Nis))
(GDcylseg (vector (- (/ Rex Nis) (/ Ep 6)) 1)
(vector 0 0 Ep -1)
(vector 0 0 Ep 1))
Nis))
(cyl_nis (DLpmi cyl_is cyl_dif))

;;------(calculer E2c)
;E2c :est la distance entre le centre de l'arc dans son cas initial ou
;(Ga=Pi/2) et et le centre du nouvel arc ou Ga prend sa valeur
;du résultat de la filtration précédente ))
;Rex-n: le rayon externe du nouvel arc
;Rin_n: le rayon interne du nouvel arc
(Rex_n (/ Rex (- 1 (cos Ga))))
(Rin_n (- Rex_n Ep))
(E2c (- Rex_n Rex))

;;------(calculer Y_cuth)
;Y_cuth: est (la coordonnée Y) du plan de la délimitation horizontale
;Y_be: est (la coordonnée Y) du plan de la délimitation horizontale dans le cas où (l'angle Ga = Pi / 2)
(Y_be (* Rex (sin Be)))
(Y_cuth (/ (* Y_be Rex_n) Rex))

;------(calculer Zmove : si la valeur de (l'angle Be n'est pas égal à zéro) l'arc de base doit se
déplacer sur l'axe Z de sorte que la règle de dôme entrelacé généré soit respectée.
;Zmove: est la distance du déplacement de l'arc de base sur l'axe Z Z.
;X: est la distance entre la projection de l'extrémité inférieure de l'arc primitif sur l'axe (xx) et le centre
de cet arc
;Xp: est la distance entre la projection de l'extrémité inférieure de l'arc primitif sur l'axe (xx) et le plan
de la délimitation horizontale .
;Rc_n: est le rayon de la nouvelle base du dôme.
;Zp: est la distance entre la projection de l'extrémité inférieure de l'arc primitif sur l'axe (zz) et sa
projection sur le plan (xoy) .

(X (* Rex_n (cos Be)))
(Xp (- X E2c))
(Rc_n (/ Xp (cos Te)))
(Zp (* Rc_n (sin Te)))
(Zmove (- Dc Xp))

;;------(la définition du plan de la symétrie (Plsym) de l'arc de base )
(Plsym
(SGx_zx3z
(vector 0 0 0 1)
(vector 0 1 0 0)
(vector 0 0 1 0)))

```

```

;;------(La quadrique de l'arc de base est un cylindre)
(qdr_arc (DLformz2
(vector
(vector Rex 0 0 1)
(vector (+ E2c Rex_n) 0 0 -1)
(vector 0 1 0 0)
(vector 0 0 1 0)
(vector (- E2c) Rex_n 0 1)
(vector 0 0 1 0)
)))

;;------(La quadrique d'enlèvement interne délimite la surface interne de l'arc de base)
(qdr_imp (DLformz2
(vector
(vector Rin 0 0 1)
(vector (+ E2c Rin_n) 0 0 -1)
(vector 0 1 0 0)
(vector 0 0 1 0)
(vector (- E2c) Rin_n 0 1)
(vector 0 0 1 0)
)))

;;------(La quadrique de l'intersection délimite l'arc sur l'axe zz)
(qdr_int (DLformz2
(vector
(vector 0 0 Zp 1)
(vector 0 0 (+ Zp Ep) 1)
(vector 1 0 0 0)
(vector 0 1 0 0)
(vector 1 0 0 0)
(vector 0 1 0 0)
)))

;;------(Le plan de la délimitation horizontale)
(plan_cuth (GDparseg
(vector 0 Y_cuth 0 1)
(vector 0 Rc 0 -0.01)
))

;;------(Le plan de la délimitation verticale)
(plan_cutv (GDparseg
(vector 0 0 0 1)
(vector Rc 0 0 -0.01)
))

;;------(opérations kleen pour délimiter l'arc de base)
(arc_p (Dlpmi (Dlint qdr_arc qdr_int) qdr_imp plan_cuth plan_cutv ))

;;------(la couleur de l'arc de base)
(arc_col (DLatt (SDcolRGB (vector 190 100 100 255))
arc_p))

```

```

;===== (mmx_$f (min max) (accélération d'affichage)
;----- (changement de la boîte englobant min-max)
(X_beta (* Rex_n (cos Be)))
(X_box
(if (< Be 0)
(/ Rex 2)
(/ (- X_beta E2c) 2)))
(S_Ga (if (> Ga (/ Pi 2))
1 (sin Ga)))
(arc_mmx (mmx_$f 0 Y_cuth (+ Zp Ep) X_box (/ (+ (* Rex_n S_Ga) Y_cuth) 2) (+ Zp (/ Ep 2)) arc_col))

;-----symétrie
(arc_sym
(DLuni
arc_mmx
(DLatt (SDmatrep (SGmatmz Plsym))
arc_mmx)))

;===== (Établir les éléments de l'illustration géométrique de l'arc de base)
;===== Les coordonnées des points de repère
(Ray_s (vector Rc 18))
(Ray_c (vector Rc 45))
(C1 (vector (- E2c) 0 0 1))
(C2 (vector E2c 0 0 1))
(H1 (vector 0 (* Rex_n (sin Ga)) 0 1))
(H2 (vector 0 (* Rex_n (sin Ga)) (+ Zp (/ Ep 2)) 1))
(Cu1 (vector Xp Y_cuth 0 1))
(Cu2 (vector (- Xp) Y_cuth 0 1))
(Cua1 (vector Xp Y_cuth (+ Zp (/ Ep 2)) 1))
(Cua2 (vector (- Xp) Y_cuth (+ Zp (/ Ep 2)) 1))

;===== Les points de repère représentés par de petites sphères
(sph_c1 (pointf 10 (- E2c) 0 0 150 100 50))
(sph_c2 (pointf 10 E2c 0 0 150 100 50))
(sph_h1 (pointf 10 0 (* Rex_n (sin Ga)) 0 100 70 120))
(sph_h2 (pointf 10 0 (* Rex_n (sin Ga)) (+ Zp (/ Ep 2)) 20 100 120))
(sph_cu1 (pointf 10 Xp Y_cuth 0 20 100 120))
(sph_cu2 (pointf 10 (- Xp) Y_cuth 0 20 100 120))
(sph_cua1 (pointf 10 Xp Y_cuth (+ Zp (/ Ep 2)) 20 100 120))
(sph_cua2 (pointf 10 (- Xp) Y_cuth (+ Zp (/ Ep 2)) 20 100 120))

;===== Les points de repère sont reliés par des cylindres représentant des bars
(cyl_c1.h1 (DLatt (SDcolRGB (vector 255 50 50 255))
(GDcylseg Ray_c C1 H1)))

(cyl_c2.h1 (DLatt (SDcolRGB (vector 255 50 50 255)) (GDcylseg Ray_c C2 H1)))
(cyl_c1.cu1 (DLatt (SDcolRGB (vector 50 150 50 255)) (GDcylseg Ray_c C1 Cu1)))
(cyl_c2.cu2 (DLatt (SDcolRGB (vector 50 150 50 255)) (GDcylseg Ray_c C2 Cu2)))
(cyl_h1.h2 (DLatt (SDcolRGB (vector 50 50 200 255)) (GDcylseg Ray_c H1 H2)))
(cyl_cu1.cua1 (DLatt (SDcolRGB (vector 50 50 200 255)) (GDcylseg Ray_c Cu1 Cua1)))
(cyl_cu2.cua2 (DLatt (SDcolRGB (vector 50 50 200 255)) (GDcylseg Ray_c Cu2 Cua2)))
;----- (L'unification de l'ensemble de sphère et de cylindres)
(en_sph (DLuni sph_c1 sph_c2 sph_h1 sph_h2 sph_cu1 sph_cu2 sph_cua1 sph_cua2 cyl_c1.h1
cyl_c2.h1 cyl_c1.cu1 cyl_c2.cu2 cyl_h1.h2 cyl_cu1.cua1 cyl_cu2.cua2 ))

```

```

;;-----Re_gout (La fonction de génération d'arc gothique)
(kk (if (= Ngo 1) (DLuni en_sph
(Re_gout_$f 10 8
(* (SGcst_pi) (/ 1 3)) (* (SGcst_pi) (/ -0 8 )) 1 1))
en_sph))
;;----- (Les conditions de génération de l'arc gothique)
(edf (cond
((= Ngo 1) (DLuni kk en_sph))
((not (zero? Nis)) (DLuni en_sph cyl_nis ))
(else en_sph)
)))
;;----- (l'arc de base)
(Dluni arc_sym ;;kk ;;sty edf
)))

;;===== (fonction de visualisation) =====
(define PIgeo
(lambda ()
(begin
(PIreset)
(PIswin (vector 640 480))
(PIsobs (vector 1 1 1.5 0))
(PIstarget (vector 0 0 0 1))
(PIsbox (vector 40 40))
(PIalign
(vector
(vector 2 1 1 0) 3))
(PIdraw->targa
(DLuni
(geo_disf 10 8 (* (SGcst_pi) (/ 1 3)) (* (SGcst_pi) (/ -1 8 )) 0 3 0)
(repere 10)
)"geo_d_is3.tga")
)))
(PIgeo)
;;=====FIN=====

```

Exemples:

La fonction (geo_disf) peut générer tous les types d'arcs pointus neutres et brisés. Elle génère aussi les deux types d'arcs islamiques et gothiques avec leurs illustrations géométriques.

Les paramètres de la fonction:

(geo_disf Rc Na Gama Be Nis n Ngo)

Exemple-1: générer un arc pointu, voir chapitre-iv, page-82, paragraphe-2.A, figure-65

(geo_disf 10 8 (* (SGcst_pi) (/ 2 5)) (* (SGcst_pi) (/ -1 8)) 0 3 0).

Exemple-2: générer un arc brisé, voir chapitre-iv, page-83, paragraphe-2.A, figure-66-c

(geo_disf 10 8 (* (SGcst_pi) (/ 2 3)) (* (SGcst_pi) (/ -0 8)) 0 3 0).

Exemple-3: générer un arc islamique, voir chapitre-iv, page-81, paragraphe-2, figure-63-a

(geo_disf 10 8 (* (SGcst_pi) (/ 2 3)) (* (SGcst_pi) (/ -0 8)) 1 3 0).

Exemple-4 générer un arc gothique, voir chapitre-iv, page-80, paragraphe-2, figure-63

(geo_disf 10 8 (* (SGcst_pi) (/ 2 3)) (* (SGcst_pi) (/ -0 8)) 0 3 1).

A-A2-algorithme(arc-dis.scm)

-Nom de la fonction; (arc-dis)

-Autre: Jamil Moussa

-Date: septembre 2000

-Objectif: la fonction consiste en des données de base permettant de générer l'arc de base et le dôme entrelacé correspondant à ce type d'arc.

-Liste des fonctions appelées durant l'exécution:

(Ploadr "descart.scm")

(Ploadr "rotation.scm")

(Ploadr "ramli.scm")

(Ploadr "mmx.scm")

-Définition de paramètres utilisées dans la fonction:

;;Rc: :rayon de dôme

;;Na: :nombre des arcs générés dans la construction de dôme

;;Gama: angle de la délimitation verticale de l'arc de base

;;Be: angle de la délimitation horizontale de l'arc de base

;;Nis: paramètre du filtre de la règle gothique

;;n: nombre de division de l'arc islamique

;;Nag: paramètre du filtre de la règle islamique

-Définition des variables utilisées dans la fonction:

;;Te: angle de la position de l'arc de base par rapport à l'axe xx

;;Ep: épaisseur de l'arc de base (la section de l'arc est carrée)

;;Rex: le rayon externe de l'arc de base

;;Rin: le rayon interne de l'arc de base

;;Dc: la distance entre le centre du dôme et la face interne de l'arc de base

;;De: la distance entre le centre du dôme et la face externe de l'arc de base

;;===== (corps de la fonction « arc_disf ») =====

(define arc_disf

(lambda (Rc Na Gama Be Nis n Ngo)

(let* (

(Pi (* (SGcst_pi) 1))

(Te (/ Pi Na))

(Ep (/ Rc 12))

(Rex (* Rc (cos Te)))

(Rin (- Rex Ep))

(Dc (* Rc (sin Te)))

(De (+ Dc Ep))

;;----- (Nis= règle islamique , Ngo= règle gothique) (filtrer l'angle Ga)

(Ga

(cond

((and (not (zero? Nis)) (not (zero? Ngo))) (display "error at least Nis or Ngo , have to be 0"))

((not (zero? Nis)) (acos (/ 1 (+ 1 n)))) ;;islamique

((= Ngo 1) (acos (/ (tan (/ Pi 8)) (+ 1 (tan (/ Pi 8))))) ;;gothique

(else (+ Gama)) ;;cas générale (sans style)))

```
;;------(Calculer E2c
;E2c: est la distance entre le centre de l'arc dans son cas initial où ;(Ga = Pi/2) et et le centre du
nouvel arc où Ga prend sa valeur ;du résultat de la filtration précédente ))
;Rex-n: le rayon externe du nouvel arc
;Rin_n: le rayon interne du nouvel arc
(Rex_n (/ Rex (- 1 (cos Ga))))
(Rin_n (- Rex_n Ep))
(E2c (- Rex_n Rex))
```

```
;;------(Calculer Y_cuth
;Y_cuth: est (la coordonnée Y) du plan de la délimitation horizontale
;Y_be:est (la coordonnée Y) du plan de la délimitation horizontale
;dans le cas où (l'angle Ga = Pi / 2)
(Y_be (* Rex (sin Be)))
(Y_cuth (/ (* Y_be Rex_n) Rex))
```

```
;;------(Calculer Zmove : si la valeur de l'angle Be n'est pas égale à zéro
;l'arc de base doit se déplacer sur l'axe Z de sorte que la règle de dôme
;entrelacé généré soit respectée.
;Zmove: est la distance du déplacement de l'arc de base sur l'axe Z de
;X: est la distance entre la projection de l'extrémité inférieure de l'arc
;primitif sur l'axe (xx) et le centre de cet arc
;Xp: est la distance entre la projection de l'extrémité inférieure de l'arc
;base sur l'axe (xx) et le plan de la délimitation horizontale .
;Rc_n: est le rayon de la nouvelle base du dôme
;Zp: est la distance entre la projection de l'extrémité inférieure de l'arc
;base sur l'axe (zz) et sa projection sur le plan (xoy) .
;toutes ces variables illustrées dans (2-3-c chapitre III)
```

```
(X (* Rex_n (cos Be)))
(Xp (- X E2c))
(Rc_n (/ Xp (cos Te)))
(Zp (* Rc_n (sin Te)))
(Zmove (- Dc Xp))
```

```
;;------(La définition du plan de symétrie (Plsym) de l'arc de base )
(Plsym
(SGx_zx3z (vector 0 0 0 1)
(vector 0 1 0 0)
(vector 0 0 1 0)
))
```

```
;;------(La quadrique de l'arc base est un cylindre)
(qdr_arc (DLformz2
(vector
(vector Rex 0 0 1)
(vector (+ E2c Rex_n) 0 0 -1)
(vector 0 1 0 0)
(vector 0 0 1 0)
(vector (- E2c) Rex_n 0 1)
(vector 0 0 1 0)
)))
```

```

;;------(La quadrique d'enlèvement interne délimite la surface interne de l'arc)
(qdr_imp (DLformz2
(vector
(vector Rin 0 0 1)
(vector (+ E2c Rin_n) 0 0 -1)
(vector 0 1 0 0)
(vector 0 0 1 0)
(vector (- E2c) Rin_n 0 1)
(vector 0 0 1 0)
))))

;;------(La quadrique de L'intersection délimite de l'arc sur l'axe zz)
(qdr_int (DLformz2
(vector
(vector 0 0 Zp 1)
(vector 0 0 (+ Zp Ep) 1)
(vector 1 0 0 0)
(vector 0 1 0 0)
(vector 1 0 0 0)
(vector 0 1 0 0)
))))

;;------(Le plan de délimitation horizontale)
(plan_cuth (GDparseg
(vector 0 Y_cuth 0 1)
(vector 0 Rc 0 -0.01)))

;;------(Le plan de la délimitation verticale)
(plan_cutv (GDparseg
(vector 0 0 0 1)
(vector Rc 0 0 -0.01)))

;;------(opérations kleen pour délimiter l'arc de base)
(arc_p (Dlpmi (Dlint qdr_arc qdr_int) qdr_imp plan_cuth plan_cutv ))

;;------(La couleur à l'arc de base)
(arc_col (DLatt (SDcolRGB (vector 190 100 100 255))
arc_p))

;===== (mmx_$f (min max) (accélération d'affichage)
;;------(changement de la boîte englobant min max)
(X_beta (* Rex_n (cos Be)))
(X_box
(if (< Be 0)
(/ Rex 2)
(/ (- X_beta E2c) 2)
))
(S_Ga (if (> Ga (/ Pi 2))
1
(sin Ga)))
(arc_mmx (mmx_$f 0 Y_cuth (+ Zp Ep) X_box (/ (+ (* Rex_n S_Ga) Y_cuth) 2) (+ Zp (/ Ep 2)) arc_col))

;;------(symétrie)
(arc_sym (Dluni arc_mmx
(DLatt (SDmatrep (SGmatmz Plsym)) arc_mmx)))
arc_sym)))

```

```

;===== (fonction de visualisation) =====
(define PIVar
(lambda ()
(begin
(PIreset)
(PISwin (vector 640 480))
(PIsobs (vector 1 1 1.5 0))
(PIstarget (vector 0 0 0 1))
(PIsbox (vector 40 40))
(PIalign
(vector
(vector 2 1 1 0)
3))
(PIDraw->targa
(DLuni
(gen_arcpri 8 (arc_disf 10 8 (* (SGcst_pi) (/ 1 2)) (* (SGcst_pi) (/ -0 8)) 0 3 0))
(gen_arcpri 8 (ramlif 10 8 (* (SGcst_pi) (/ 1 2)) (* (SGcst_pi) (/ -0 8)) 0 3 0))
(repere 10)
)"dome_p-1.3.tga"))
(PIvar)
;=====FIN=====

```

Exemple :

Les paramètres de la fonction: (Rc Na Gama Be Nis n Ngo)

(gen_arcpri Na (arc_disf Rc Na Gama Be Nis n Ngo))

La ligne de code précédente génère un dôme entrelacé sans remplissage

Les valeurs de paramètres:

Rc = 10

Na = 8

Gama = $(* (SGcst_pi) (/ 1 2)) = (Pi / 2)$

Be = $(* (SGcst_pi) (/ -0 8)) = 0$

Nis = 0

n=3 (si Nis=0 la valeur de « n » n'a aucun effet)

Ngo = 0

Substituer à chaque paramètre sa valeur

(gen_arcpri 8 (arc_disf 10 8 (* (SGcst_pi) (/ 1 2)) (* (SGcst_pi) (/ -0 8)) 0 3 0))

L'algorithme retourne un dôme entrelacé sans remplissage

Pour générer le remplissage on utilise la fonction « ramlif »

(gen_arcpri Na (ramlif Rc Na Gama Be Nis Ngo))

en lui donnant les mêmes valeurs des paramètres à la fonction (ramlif)

(gen_arcpri 8 (ramlif 10 8 (* (SGcst_pi) (/ 1 2)) (* (SGcst_pi) (/ -0 8)) 0 3 0))

On unifie les deux fonctions par (DLuni), la fonction génère un dôme entrelacé avec son remplissage.

Voir l'image générée selon les paramètres ci-dessus dans

(le chapitre-IV, page-79, paragraphe 2, figure 62-b).

On change la valeur de l'angle Gama ou Ga (les angles sont les mêmes) afin de générer d'autres types de dômes entrelacés.

A-A3-algorithme (ramli.scm)

-Nom de la fonction (arc-dis)

-Autre: Jamil Moussa

-Date: novembre 2000

-Objectif : la fonction consiste à des données de base permettant de générer le remplissage du dôme entrelacé.

-Liste des fonction appelées durant l'exécution:

(PIloadr "descart.scm")

(PIloadr "rotation.scm")

(PIloadr "couper.scm")

-Définition de paramètres utilisées dans la fonction :

;;Rc: :rayon de dôme

;;Na: :nombre des arcs générés dans la construction de dôme (Na égale à 8, il est constant).

;;Gama: angle de la délimitation verticale de l'arc de base

;;Be: angle de la délimitation horizontale de l'arc de base

;;Nis: paramètredu filtre de la règle gothique

;; n: nomdre de divisions de l'arc islamique

;;Nag: paramètre du filtre de la règle islamique

```
;;===== (corps de la fonction) =====
(define ramlif
  (lambda (Rc Na Gama Be Nis Ngo)
    (let* (
      (Pi (* (SGcst_pi) 1))
      (Te (/ Pi Na))
      (Ep (/ Rc 12))
      (Rex (* Rc (cos Te)))
      (Rin (- Rex Ep))
      (Dc (* Rc (sin Te)))
      (De (+ Dc Ep))
      ;;----- (Nis-règle islamique , Ngo-règle gothique)
      (Ga
      (cond
        ((and (not (zero? Nis)) (not (zero? Ngo))) (display "error Nis or Ngo , have to be 0"))
        ((not (zero? Nis)) (acos (/ 1 (+ 1 Nis))))
        ((= Ngo 1) (acos (/ (tan (/ Pi 8)) (+ 1 (tan (/ Pi 8))))))
        (else (+ Gama))))
      ;;----- (calculer E2c (distance entre l'origine et le centre de nouvel arc))
      (Rex_n (/ Rex (- 1 (cos Ga))))
      (Rin_n (- Rex_n Ep))
      (E2c (- Rex_n Rex))
      ;;----- (calculer Y_cut "Y" de plan de la délimitation horizontale)
      (Y_be (* Rex (sin Be)))
      (Y_cuth (/ (* Y_be Rex_n) Rex))
      ;;----- (rétablir la géométrie du dôme et déplacer l'arc base)
      (X (* Rex_n (cos Be)))
      (Xp (- X E2c))
      (Rc_n (/ Xp (cos Te)))
      (Zp (* Rc_n (sin Te)))
      (Zmove (- Dc Xp))
```

```

;;-----Box_mmx (boite englobante)
(X_beta (* Rex_n (cos Be)))
(X_box
(if (< Be 0)
(/ Rex 2)
(/ (- X_beta E2c) 2)))
(S_Ga (if (> Ga (/ Pi 2))
1
(sin Ga)))
(Box (list (vector 0 Y_cuth Zp 1)
(vector
(vector X_box (/ (+ (* Rex_n S_Ga) Y_cuth) 2) (/ Zp 2) 1)
(vector 1 0 0 0)
(vector 0 1 0 0)
(vector 0 0 1 0))))))
;;-----Plsym (plan de symétrie)
(Plsym
(SGx_zx3z (vector 0 0 0 1)
(vector 0 1 0 0)
(vector 0 0 1 0)))
;;-----(quadrique de l'arc base)
(qdr_arc (DLformz2
(vector
(vector Rex 0 0 1)
(vector (+ E2c Rex_n) 0 0 -1)
(vector 0 1 0 0)
(vector 0 0 1 0)
(vector (- E2c) Rex_n 0 1)
(vector 0 0 1 0))))))
;;-----(quadrique d'enlèvement)
(qdr_imp (DLformz2
(vector
(vector Rin 0 0 1)
(vector (+ E2c Rin_n) 0 0 -1)
(vector 0 1 0 0)
(vector 0 0 1 0)
(vector (- E2c) Rin_n 0 1)
(vector 0 0 1 0))))))
;;-----(quadrique de L'intersection)
(qdr_int (DLformz2
(vector
(vector 0 0 0 1)
(vector 0 0 Zp 1)
(vector 1 0 0 0)
(vector 0 1 0 0)
(vector 1 0 0 0)
(vector 0 1 0 0))))))
;;----- (plan de la délimitation horizontale)
(plan_cuth (GDparseg
(vector 0 Y_cuth 0 1)
(vector 0 Rc 0 -0.01)))
;;----- (plan de la délimitation verticale)
(plan_cutv (GDparseg
(vector 0 0 0 1)
(vector Rc 0 0 -0.01)))

```

```

;;------(plan de la délimitation cutz)
(plan_cutz (GDparseg
(vector 0 0 0 1)
(vector 0 0 Rc -0.01)))
;;------(plan de la délimitation selon les angles p/8 p/4 3p/8)
(plan_1arc (GDparseg
(vector De 0 0 1)
(vector Rc 0 0 -0.01)))
(plan_2arc (DLatt (SDmatrep (SGmatrot
(vector (/ Pi 4) -1)
(vector De 0 (* De (tan Te)) 1)
(vector De 1 (* De (tan Te)) 1))))
plan_1arc))
;;------(arc principal)
(arc_p (Dlpmi (Dlint qdr_arc qdr_int) qdr_imp plan_cuth plan_cutv ))
;;-----tailler part-1
(arc_1p/8 (DLpmi
arc_p
(couperf 10 (/ Pi 8))
plan_1arc))
;;-----colore
(arc_1p/8_col
(DLatt (SDcolRGB (vector 120 100 190 255))
arc_1p/8))
;;-----tailler part-2
(arc_1p/4 (DLpmi
arc_p
(couperf 10 (/ Pi 4))
plan_2arc))
(arc_1p/4_col (DLatt (SDcolRGB (vector 255 0 0 255)) arc_1p/4))
;;-----tailler part-3
(arc_3p/8 (DLpmi
arc_p
(couperf 10 (/ (* Pi 3) 8))))
(arc_3p/8_col (DLatt (SDcolRGB (vector 60 150 100 255)) arc_3p/8))
;;------(unifier les parties)
(primitive (DLuni arc_1p/8_col arc_1p/4_col arc_3p/8_col ))
;;-----mmx(boite englobant)
(arc_mmx
;; (DLuni
;; (apply Exmechxa (append Box (list (vector 1 4))))
(DLmmx
(DLformz2 (apply SGqdrhxa Box)
primitive) );;)
;;------(créer la symétrie des parties construites)
(peau
(DLuni
arc_mmx
(DLatt (SDmatrep (SGmatrmz Plsym)) arc_mmx)))
peau)))
;=====
(Fonction de visualisation)
;=====FiN=====

```

A-A4-algorithme (Re_gout.scm)

-Nom de la fonction (arc-dis)

-Autre: Jamil Moussa

-Date: octobre 2000

-Objectif : la fonction permet de générer un arc selon la règle gothique avec son illustration géométrique.

-Liste des fonction appelées durant l'exécution:

(Ploadr "descart.scm")

(Ploadr "outil.scm")

(Ploadr "mmx.scm")

-Définition des paramètres utilisées dans la fonction :

;;Rc: :rayon de dôme

;;Na: :nombre des arcs générés dans la construction de dôme (dans ce paramètre est égale à (8), il est constant.

;;Gama: angle de la délimitation verticale de l'arc de base

;;Be: angle de la délimitation horizontale de l'arc de base

;;Nag: paramètre du filtre de la règle islamique

```

;;===== (corps de la fonction) =====
(define Re_gout_$f
  (lambda (Rc Na Gama Be Ngo)
    (let* (
      (Pi (* (SGcst_pi) 1))
      (Te (/ Pi Na))
      (Ep (/ Rc 12))
      (Rex (* Rc (cos Te)))
      (Rin (- Rex Ep))
      (Dc (* Rc (sin Te)))
      (De (+ Dc Ep))
      ;;-----L'intersection entre l'arc et la ligne qui passe la tête de l'arc et est parallèle à l'axe YY
      (qdr_poi (Sgqdrarray
        (vector 0 0 0 1)
        (vector 0 1 0 1)
        (vector
          (vector Rex 0 0 1)
          (vector (+ Rex (* Rex (tan (/ Pi 8)))) 0 0 -1)
          (vector 0 1 0 0)
          (vector 0 0 1 0)
          (vector (tan (/ Pi -8)) (+ Rex (* Rex (tan (/ Pi 8)))) 0 1)
          (vector 0 0 1 0))))
      ;;-----retourner le point qdr_poi et y positionner une sphère
      (poi_y (vector-ref qdr_poi 1))
      (sph_y (DLatt (SDcolRGB (vector 20 10 255 255))
        (GDspdis (vector Rc 13) poi_y)))
      ;;-----installer une petite sphère au point qui est représenté par le vecteur
      (vector (* Rex (tan (/ Pi -8))) 0 0 1)
      (sph_c2 (DLatt (SDcolRGB (vector 0 255 0 255))
        (GDspdis (vector Rc 13)
          (vector (* Rex (tan (/ Pi -8))) 0 0 1))))
      ;;-----construire une cylindre
      (cylr (DLatt (SDcolRGB (vector 20 150 180 255))
        (GDcylseg (vector Rc 55)
          (vector Rex 0 0 1)
          (vector Rex 0 0 -1))))

```



```

;;-----définir les deux lignes D1, D2 par la fonction (Sgy_coo)
(D1 (SGy__coo (vector 0 0 0 1) (vector 1 0 0 0)))
(D2 (SGy__coo (vector (* Rex (tan (/ Pi -8))) 0 0 1) poi_y))
;;-----retourner le cos de l'angle entre les deux lignes D1, D2
(co (SGcos_2y D1 D2))
;;-----installer un point de repère (petite sphère)
(sph_yy (DLatt (SDcolRGB (vector 0 255 0 255))
(GDsphdis (vector Rc 13)
(vector 0 (* (sin (acos co)) (+ Rex (* Rex (tan (/ Pi 8)))) 0 1))))
;;-----construire une cylindre
(cyl_cen/poiy (DLatt (SDcolRGB (vector 20 150 180 255))
(GDcylseg (vector Rc 35)
(vector (* Rex (tan (/ Pi -8))) 0 0 1)
(vector 0 (* (sin (acos co)) (+ Rex (* Rex (tan (/ Pi 8)))) 0 1)
)))
;;-----construire l'illustration géométrique de l'arc gothique
(reg (DLuni cylr (rotf 10 8 1 0 Rex 0 -1 0 0 1 0 cylr)))
(reg_1 (DLuni reg (rotf 10 8 2 0 Rex 0 -1 0 0 1 0 reg)))
(reg_2 (DLuni reg_1 (rotf 10 8 4 0 Rex 0 -1 0 0 1 0 reg_1)))
;=====mmx_$f (boite englobant
(D_carre (mmx_$f (* Rex (sqrt 2)) (- (* Rex (sqrt 2)) Rex) (/ Rex 55) 0 (- Rex) 0 reg_2))
) D-carre )))
;====Note: La fonction (Re_gout_$f) était déjà testée dans le cadre de la fonction (geo_disf)
;(fonction de la visualisation)
=====FIN=====

```

A-A5 -algorithme (descart.scm)

```

;cette fonction génère le système de coordonnées
;=====-(corps de la fonction)=====
(define repere
(lambda (Rc)
(let*
((Dia (/ Rc 200)))
;-----trois axes (X Y Z)
(DLuni
(Dlatt (SDcolRGB (vector 255 0 0 255)) ;;Axe X
(GDcylseg
(vector Dia 1)
(vector Rc 0 0 -0.01)
(vector Rc 0 0 0.01)))
(Dlatt (SDcolRGB (vector 0 255 0 255)) ;;Axe Y
(GDcylseg
(vector Dia 1)
(vector 0 Rc 0 -0.01)
(vector 0 Rc 0 0.01)))
(Dlatt (SDcolRGB (vector 0 0 255 255)) ;;Axe Z
(GDcylseg
(vector Dia 1)
(vector 0 0 Rc -0.01)
(vector 0 0 Rc 0.01)))
))))
;=====FIN=====

```

A-A6-algorithme (mmx.scm)

```

Fonction de la boite englobant ( min max)
;=====le corps de la fonction=====
(define mmx_$f
(lambda ( Xs Ys Zs Xc Yc Zc vol)
(let*
(
(Box (list (vector Xs Ys Zs 1)
(vector
(vector Xc Yc Zc 1)
(vector 1 0 0 0)
(vector 0 1 0 0)
(vector 0 0 1 0)
)))
(vol_mmx
;; (DLuni
;; (apply EXmechxa (append Box (list (vector 1
4))))
(DLmmx
(DLformz2 (apply SGqdrhxa Box))
vol)
);;)
)
vol_mmx
)))
;=====FIN=====

```

A-A7-algorithme (outil.scm)

Objectif: Cette fonction permet de repositionner une cylindre dans l'espace

```
=====première fonction=====
;;Les paramètres sont:
;;Rc: rayon du dôme
;;Na: nombre d'arcs
;;An: angle de rotation
;;x y z f les coordonnées du point de départ de
cylindre
;;x1 y1 z1 f1: les coordonnées du point
d'arrivée de cylindre
;;vol: cylindre
```

```
(define rotf
(lambda (Rc Na An x y z f x1 y1 z1 f1 vol)
(let* (
(Pi (* (SGcst_pi) 1))
(Te (/ Pi Na))
(Ep (/ Rc 12))
(Rex (* Rc (cos Te)))
(Rin (- Rex Ep))
(Dc (* Rc (sin Te)))
(De (+ Dc Ep))
(RR (DLatt (SDmatrep (SGmatrot
(vector (* (SGcst_pi) (/ 1 An)) 1)
(vector x y z f)
(vector x1 y1 z1 f1))) vol))
)
RR
)))
```

```
=====deuxième fonction=====
;;objectif: Cette fonction contrôle le plan de la
construction de l'arc de base
(define couperf
(lambda (Rc A_rot)
(let* (
(plan_cutz
(GDparseg
(vector 0 0 0 1)
(vector 0 0 Rc -0.01)))
(plan_rot (DLatt (SDmatrep (SGmatrot
(vector A_rot -1)
(vector 0 0 0 1)
(vector 0 1 0 1)))plan_cutz)))
plan_rot
)))
```

```
=====troisième fonction=====
;;objectif: cette fonction contrôle le
positionnement d'un point dans l'espace
(define pointf
(lambda (Rc X Y Z R G B)
(let* (
(bull (DLatt (SDcolRGB (vector R G B 255))
(GDsphdis
(vector Rc 18)
(vector X Y Z 1))))))
bull
)))
=====FIN=====
```

A-A8-algorithme (rotation.scm)

La fonction de génération polaire

```
===== (fonction de la préparation de list) =====
(define liste_theta
(lambda (n)
(let f ((acc n))
(if (= acc 0) (list)
(cons (vector (* acc 2 (SGcst_pi)) n) (f (- acc 1))))))
```

```
===== (fonction de génération) =====
;;Paramètres
;;n nombre de volume générée
;;vol volume
```

```
(define gen_arcpri
(lambda (n vol)
(apply
DLuni
(map
(lambda (v)
(DLatt
(SDmatrep
(SGmatrot v (vector 0 0 0 1) (vector 0 1 0 1))
)
vol))
(liste_theta n)
)))
=====FIN=====
```

ANNEXE -B

PROGRAMME « prog_ellipse »

ANNEXE -B

PROGRAMME « prog_ellipse »

Les fonctions des dômes entrelacés dont l'arc de base est continu . Ces fonctions sont regroupées dans le cadre du programme (prog_ellipse).

Liste des fonctions du programme (prog_ellipse)

A-B1-algorithme (geo-elli.scm)	F-« géo-ellif »
A-B2-algorithme(arc-elli.scm)	F-« arc-ellif »
A-B3-algorithme (peau.scm)	F-« peauf »
A-B6-algorithme (pointell.scm)	F-« cylin_\$f » F-« point_\$f »
A-B4-algorithme (descart.scm)	F-« repere »
A-B5-algorithme (mmx.scm)	F-«mmx_\$f »
A-B7-algorithme (rotation.scm)	F-«gen_arcpri »

A-B1-algorithme (geo_elli.scm)

-Nom de la fonction (geo_ellif)

-Autre: Jamil Moussa

-Date: octobre 2000

-Objectif : la fonction permet de générer un arc elliptique avec son illustration géométrique.

-Liste de fonctions appelées durant l'exécution :

(load "rotation")

(load "descart")

(load "mmx")

(load "point")

-Définition des paramètres utilisées dans la fonction :

;Rc: rayon de dôme

;Na: nombre des arcs générés dans la construction de dôme

;Fa: facteur de la déformation verticale de l'arc primitif

;Ga: l'angle entre l'axe xx et le rayon qui trace l'arc primitif

-Définition des variables utilisées dans la fonction :

;Te: angle de position de l'arc de base par rapport à l'axe xx

;Rex: le rayon externe de l'anneau dont le diamètre est le petit axe de l'ellipse

;Rex_in: le rayon interne de l'anneau dont le diamètre est le petit axe de l'ellipse

;Rey: le rayon externe de l'anneau dont le diamètre est le grand axe de l'ellipse

;Rey_in: le rayon interne de l'anneau dont le diamètre est le grand axe de l'ellipse

;;Dc: la distance entre le centre du dôme et la face interne de l'arc de base

;;De: la distance entre le centre du dôme et la face externe de l'arc de base

;;===== corps de la fonction=====

```
(define geo_ellif
```

```
(lambda (Rc Na Fa Ga)
```

```
(let*
```

```
(
```

```
(Pi (* (SGcst_pi) 1))
```

```
(Te (/ Pi Na))
```

```
(Rex (* Rc (cos Te)))
```

```
(Rex_in (- Rex (/ Rc 50)))
```

```
(Rey (* Rex Fa))
```

```
(Rey_in (- Rey (/ Rc 50)))
```

```
(Ep (/ Rc 12))
```

```
(Dc (* Rc (sin Te)))
```

```
(De (+ Dc Ep))
```

;;-----l'anneau dont le diamètre est le petit axe

```
(qdr_x (vector
```

```
(vector Rex 0 0 -1)
```

```
(vector Rex 0 0 1)
```

```
(vector 0 1 0 0)
```

```
(vector 0 0 1 0)
```

```
(vector 0 Rex 0 1)
```

```
(vector 0 0 1 0)))
```

```

;;-----Le quadrique d'enlèvement du matière de qdr_x
(qdr_x_dif (vector
(vector Rex_in 0 0 -1)
(vector Rex_in 0 0 1)
(vector 0 1 0 0)
(vector 0 0 1 0)
(vector 0 Rex_in 0 1)
(vector 0 0 1 0)))
;-----
(qdr_y (vector
(vector Rey 0 0 -1)
(vector Rey 0 0 1)
(vector 0 1 0 0)
(vector 0 0 1 0)
(vector 0 Rey 0 1)
(vector 0 0 1 0)))
;-----
(qdr_y_dif (vector
(vector Rey_in 0 0 -1)
(vector Rey_in 0 0 1)
(vector 0 1 0 0)
(vector 0 0 1 0)
(vector 0 Rey_in 0 1)
(vector 0 0 1 0)))
;;-----L'intersection du rayon traçant l'ellipse avec l'anneau du petit axe (retourne deux
points)
(qdr_ray_x (SGqdrarray
(vector 0 0 0 1)
(vector (* Rey (cos Ga)) (* Rey (sin Ga)) 0 1)
qdr_x))
;;-----L'intersection du rayon traçant l'ellipse avec l'anneau du grand axe (retourne
deux points)
(qdr_ray_y (SGqdrarray
(vector 0 0 0 1)
(vector (* Rex (cos Ga)) (* Rex (sin Ga)) 0 1)
qdr_y))
;;-----Les deux points retournés sont:
(po_x (vector-ref qdr_ray_x 1))
(po_y (vector-ref qdr_ray_y 1))
;;-----Le point d'intersection des deux droites. Ce point fait partie de l'ellipse tracé .
(po_el (SGx___4x po_x (vector 0 1 0 0)
po_y (vector 1 0 0 0)))
;-----
(po_el_arc (SGx___dis (vector (+ Dc (/ Ep 2)) 1) po_el
(vector 0 0 1 0)))

;;-----Repérer les points principaux par de petites sphères ;(point_$f est la fonction qui
positionne les points
(sph_x (point_$f 10 po_x 150 200 0 ))
(sph_y (point_$f 10 po_y 0 255 0 ))
(sph_el (point_$f 10 po_el 80 80 180))
(sph_o (point_$f 10 (vector 0 0 0 1) 255 0 0))
(sph_el_arc (point_$f 10 po_el_arc 80 180 30))

```

```

;;-----Les bars reliant les points de repère (cylin$f est la fonction générant et positionnant les bars)
(c_o_pox (cylin_$f 10 (vector 0 0 0 1) po_x 220 200 0))
(c_pox_poy (cylin_$f 10 po_x po_y 50 220 0))
(c_pox_poel (cylin_$f 10 po_x po_el 50 220 0))
(c_poy_poel (cylin_$f 10 po_y po_el 50 220 0))
(c_poel_poelarc (cylin_$f 10 po_el po_el_arc 255 0 0))

;;-----
(c_pel_pelar (cylin_$f 10 (vector 0 Rey 0 1)
(vector 0 Rey (+ Dc (/ Ep 2)) 1) 180 220 10))
(c_x+ (cylin_$f 10 (vector Rex 0 0 1)
(vector Rex 0 (+ Dc (/ Ep 2)) 1) 180 220 10))
(c_x- (cylin_$f 10 (vector (- Rex) 0 0 1)
(vector (- Rex) 0 (+ Dc (/ Ep 2)) 1) 180 220 10))
;;-----anneau_x (petit axe)
(cadre_x (DLpmi
(DLformz2 qdr_x)
(DLformz2 qdr_x_dif)
))
;;-----anneau_y (grand axe)
(cadre_y (DLpmi
(DLformz2 qdr_y)
(DLformz2 qdr_y_dif)
))
;;-----pl_xy_int
(pl_xy_int (GDparseg (vector 0 0 (/ Ep -5) 1)
(vector 0 0 (/ Ep 5) 1)))
;;-----
(guid_x (DLatt (SDcolRGB (vector 20 200 20 255))
(DLint cadre_x pl_xy_int)))

;;-----(boite englobant de l'anneau-x)
(guid_x_mx (mmx_$f Rex Rex (/ Ep 2) 0 0 0 guid_x))
;;-----
(guid_y (DLatt (SDcolRGB (vector 200 20 20 255))
(DLint
cadre_y
pl_xy_int)))
;;-----(boite englobant de l'anneau-y)
(guid_y_mx (mmx_$f Rey Rey (/ Ep 2) 0 0 0 guid_y))
;;-----
(guid_xy (DLuni guid_x_mx guid_y_mx))
(sph (DLuni sph_x sph_y sph_el sph_o sph_el_arc c_pel_pelar c_x- c_x+))
(cyl
(cond
((and (> Fa 1) (or (= Ga 0) (= Ga (/ Pi 2)) (= Ga Pi))))
(DLuni c_o_pox c_pox_poy c_poel_poelarc))
((and (or (< Fa 1) (= Fa 1)) (or (= Ga 0) (= Ga (/ Pi 2)) (= Ga Pi))))
(DLuni c_o_pox c_poel_poelarc))
(else (DLuni c_o_pox c_pox_poy c_pox_poel c_poy_poel c_poel_poelarc))))
(sys (DLuni sph cyl guid_xy c_poel_poelarc)))
sys
)))

```

```

=====fonction de visualisation=====
(define PGeo_elli
(lambda ()
(begin
(Preset)
(PIsobs (vector 1 1.5 1.5 0))
(PIstarget (vector -1 0 0 2))
(PIsbox (vector 46.1 46.1))
(PIswin (vector 640 480))
(PIswincol (vector .9 1 .9 1))
(PIsightpow 0 0.7)
(PIalight
(vector(vector -1 3 1 0) 0.9))
(PIdraw (DLuni
(geo_ellif 10 8 (/ 1 .5) (* (SGcst_pi) (/ 2 3))) ;(Rc Na Fa Ga)
(repere 10)))
)))
(PGeo_elli)
=====FIN=====

```

Exemple:

```
(geo_ellif Rc Na Fa Ga)
```

```
(geo_ellif 10 8 (2) (* (SGcst_pi) (/ 2 3)))
```

L'arc généré selon les paramètres ci-dessus est un arc elliptique surbaissé.

(voir l'image, page-89, paragraphe-1.2.A, figure-73-III-cas-3)

Dans le cas où $Fa=1/2$ l'arc généré devient elliptique surhaussé

(voir l'image, page-89, paragraphe-1.2.A, figure-73-I-cas-1)

A-B2-algorithme (arc_elli.scm))

-Nom de la fonction (geo_ellif)

-Autre: Jamil Moussa

-Date: novembre 2000

-Objectif : la fonction permet de générer un dôme entrelacé dont l'arc de base est une demi-ellipse .

-Liste des fonction appelées durant l'exécution :

(Ploadr "rotation.scm")

(Ploadr "descart.scm")

(Ploadr "geo_elli.scm")

(Ploadr "peau.scm")

(Ploadr "mmx.scm")

-Définition de paramètres utilisées dans la fonction :

;Rc: rayon de dôme

;Na: nombre des arcs générés dans la construction de dôme

;Fa: :facteur de la déformation verticale de l'arc primitif

-Définition des variables utilisées dans la fonction :

;Te: angle de position de l'arc de base par apport à l'axe xx

;Rex: le petit axe de l'ellipse (face externe)

;Rex_in: le petit axe de l'ellipse (face interne)

;Rey: le grand axe de l'ellipse (face externe)

;Rey_in: le grand axe de l'ellipse (face interne)

;Dc: la distance entre le centre du dôme et la face interne de l'arc de base

;De: la distance entre le centre du dôme et la face externe de l'arc de base

;;=====corps de la fonction=====

```
(define arc_ellif
(lambda (Rc Na Fa)
(let* (
(Pi (* (SGcst_pi) 1))
(Te (/ Pi Na))
(Rex (* Rc (cos Te)))
(Ep (/ Rc 12))
(Rex_in (- Rex Ep))
(Rey (* Rex Fa))
(Rey_in (- Rey Ep))
(Dc (* Rc (sin Te)))
(De (+ Dc Ep))
```

;;-----quadrique (cylindre en section ellipsoïdale)

```
(qdr_el
(DLformz2 (vector
(vector Rex 0 0 1)
(vector Rex 0 0 -1)
(vector 0 1 0 0)
(vector 0 0 1 0)
(vector 0 Rey 0 1)
(vector 0 0 1 0))))
```

```

;;-----quadrique négative
(qdr_el_dif
(DLformz2 (vector
(vector Rex_in 0 0 1)
(vector Rex_in 0 0 -1)
(vector 0 1 0 0)
(vector 0 0 1 0)
(vector 0 Rey_in 0 1)
(vector 0 0 1 0))))

;;-----plans de construction de l'arc de base
(pl_cut (GDparseg (vector 0 0 0 1)
(vector 0 Rc 0 -0.01)))
(pl_int (GDparseg (vector 0 0 Dc 1)
(vector 0 0 De 1)))
;;-----arc de base
(arc_el
(DLint (Dlpmi qdr_el qdr_el_dif pl_cut pl_int))
;;-----color
(arc_col
(DLatt (SDcolRGB (vector 80 200 160 255))
arc_el))
;;=====mmx_$f (boite englobant)
(arc_mmx (mmx_$f Rex 0 De 0 (/ Rey 2) (+ Dc (/ Ep 2)) arc_col)))
arc_mmx
)))
;;=====fonction de visualisation =====
(define varc_elli
(lambda ()
(begin
(PIreset)
(PIsobs (vector 1 1.5 1.5 0))
(PIstarget (vector -1 0 0 2))
(PIsbox (vector 46.1 46.1))
(PIswin (vector 640 480))
(PIswincol (vector .9 1 .9 1))
(PIsightpow 0 0.7)
(PIalight
(vector(vector -1 3 1 0) 0.9))
(PIdraw
(DLuni
(gen_arcpri 6 (arc_ellif 10 6 (/ 3 2));;)
(DLdif (peauf 10 6 (/ 3 2)) (gen_arcpri 6 (arc_ellif 10 6 (/ 3 2))))
(repere 10)
)))
(Pivarc_elli)
;;=====FIN=====

```

Exemple:

```
(gen_arcpri 6 (arc_ellif 10 6 (/ 3 2));;)
```

```
(DLdif (peauf 10 6 (/ 3 2)) (gen_arcpri 6 (arc_ellif 10 6 (/ 3 2))))
```

Selon ces paramètres la fonction génère un dôme ellipsoïdal surhaussé en 6 arcs
(voir l'image, page-91, paragraphe-A.3, figure-74, image-9)

A-B3-algorithme (peau.scm)

```

Objectif: générer le remplissage du dôme
(Ploadr "mmx.scm")
;=====corps de la fonction=====
(define peauf
(lambda (Rc Na Fa)
(let* (
(Pi (* (SGcst_pi) 1))
(Te (/ Pi Na))
(Ry (* Rc Fa))
(Rex (* Rc (cos Te)))
(Ep (/ Rc 12))
(Rex_in (- Rex (/ Ep 3)))
(Ry_in (- Ry (/ Ep 2)))
(Rc_in (- Rc (/ Ep 2)))
(Dc (* Rc (sin Te)))
(De (+ Dc Ep))

;-----quadrique de la construction du
remplissage du dôme
(qdr_pe (DLformz2 (vector
(vector Rc 0 0 1)
(vector Rc 0 0 -1)
(vector 0 1 0 0)
(vector 0 0 1 0)
(vector 0 Ry 0 1)
(vector 0 0 Rc 1) )))
(qdr_dif (DLformz2 (vector
(vector Rc_in 0 0 1)
(vector Rc_in 0 0 -1)
(vector 0 1 0 0)
(vector 0 0 1 0)
(vector 0 Ry_in 0 1)
(vector 0 0 Rc_in 1))))

(pl_cut (GDparseg (vector 0 0 0 1)
(vector 0 Rc 0 -0.01)))

;-----
(habi (DLpmi qdr_pe qdr_dif pl_cut))
(habi_col (DLatt (SDcolRGB (vector 200 20
20 255)) habi))
(Sp_mmx (mmx_$f Rc 0 Rc 0 (/ Ry 2) 0
habi_col))
)
Sp_mmx
)))

;=====fonction de la vualisation=====
(define PIpeau
(lambda ()
(begin
;=====FIN=====

```

A-B4-algorithme (pointell.scm)

```

;=====corps de la fonction=====
Objectif: Cette fonction permet de construire
un point de repère (petite sphère) et le
positionner dans l'espace dans le centre v
(define point_$f
(lambda (Rc v R G B)
(let*(
(bull (DLatt (SDcolRGB (vector R G B
255))
(GDsphdis
(vector Rc 25)
;; (vector X Y Z 1)
v)
))
)
bull
)))

;=====corps de la fonction=====
Cette fonction permet de construire un
cylindre et le positionner dans l'espace selon
les deux vecteur v1 v2
(define cylin_$f
(lambda (Rc v1 v2 R G B)
(let*(
(cyl (DLatt (SDcolRGB (vector R G B
255))
(Gdcylseg
(vector Rc 55)
v1
v2)))
)
cyl
)))

;=====FIN=====

```

Pour les autres trois algorithmes
A-B4-algorithme (descart.scm)
A-B5-algorithme (mmx.scm)
A-B7-algorithme (rotation.scm)
Voir l'annexe-A-

ANNEXE -C

PROGRAMME « prog_cordoue»

ANNEXE -C

PROGRAMME « prog_cordoue »

La modélisation du contexte architectural (la coupole)

Nom du programme: (prog_cordoue)

Autre: Jamil Moussa

Date: juin 1999

Objectif: générer l'ensemble du contexte architectural selon les deux paramètres:

Rc: est le rayon du dôme entrelacé

T: est le nombre de sommets du polygone de la base du dôme entrelacé. T est également le nombre d'arcs utilisés dans la construction du dôme entrelacé.

Le programme: (prog_cordoue)

Les dix-sept fonctions regroupées dans le cadre du programme (prog_cordoue). Celui-ci permet de générer le contexte architectural (la coupole) selon les deux paramètres précédents Rc et T.

Liste des fonctions

A-C1- algorithme (corniche.scn)

A-C2- algorithme (dalle.scn)

A-C3- algorithme (musa.scn)

A-C4- algorithme (fonf.scn)

A-C5- algorithme (Trans.scn)

A-C6- algorithme (colonne.scn)

A-C7- algorithme (prism.scn)

A-C8- algorithme (arcade.scn)

A-C9- algorithme (hyper.scn)

A-C10- algorithme (p_base.scn)

A-C11- algorithmes (arcpri.scn)

A-C12- algorithme (spa.scn)

A-C13- algorithme (sp.scn)

A-C14- algorithme (spt.scn)

A-C15- algorithme (sptt.scn)

A-C16- algorithme (tête.scn)

A-C17- algorithme (resultat.scn)

A-C1- Algorithmme (corniche.scm)

Objectif: la fonction génère le cadre de la base de la coupole.

(voir image: page-iv, annexe-coupole, figure-c.3)

```

;=====corps de la fonction=====
(define cornichf
(lambda (T Rc)
(let* (
(Fi (/ (SGcst_pi) T))
(Ta (tan Fi))
(Rex (* Rc (cos Fi)))
(Dc (* Rc (sin Fi)))
(ep (/ Rex 12))
(epf (* ep (/ 3 5)))
(Rin (- Rex ep))
(De (+ Dc ep))
(Rci (- Rc (/ Rc 8)))
(Rce (+ Rc (/ Rc 8)))
(Dae (* Rce (cos Fi)))
(Dai (* Rci (cos Fi)))
(ep_arca (/ (+ Dae Dai) 2))
(Rarca_e (* Ta Dai))
(Rarca_i (- Rarca_e (* (/ 2 3) ep_arca)))

;-----boite englobant
(Box (list (vector (* epf 2) (- (/ epf 2)) (* epf (/ 5 2)
Ta) 1)
(vector
(vector (+ (* epf 2) (/ epf 4)) (- (/ (* 3 epf) 4)) 0 1)
(vector 1 0 0 0)
(vector 0 1 0 0)
(vector 0 0 1 0))))))

;------(quart de cylindre)
(primitive
(DLatt
(SDcolRGB (vector 100 55 50 255))
(DLint
(GDqcyhxa
(vector (+ (* epf 2) (/ epf 2)) (- (/ epf 2)) (+ epf (/
epf 2)) 1)
(vector
(vector (+ (* epf 2) (/ epf 4)) (- (/ (* 3 epf) 4)) 0 1)
(vector 1 0 0 0)
(vector 0 1 0 0)
(vector 0 0 1 0))))))
(DLformz2 ;-----quadrique conique
(vector
(vector (* epf 2) 0 (* epf 2) Ta) 1)
(vector (* epf 2) 0 (- (* epf 2) Ta) 1)
(vector 0 0 0 1)
(vector 0 1 0 0)
(vector 0 0 0 1)
(vector 0 1 0 0))))))

```

```

;------(Box- min max)
;;(DLuni (apply EXmechxa (append Box (list
(vector 1 20))))))
(DLmmx
(DLformz2 (apply SGqdrhxa Box))
primitive) ;; )
)))
;=====FIN=====

```

A-C2- algorithmme (dalle.scm)

Objectif: générer les dalles triangulaires de la base de la coupole.

(voir image: page-v, annexe-coupole, figure-c.4)

```

;=====corps de la fonction=====
(define dallemf
(lambda (T Rc)
(let* ( (Fi (/ (SGcst_pi) T))
(F9 (/ (SGcst_pi) 9))
(T9 (tan F9))
(Ta (tan Fi))
(S_p/8 (sin Fi))
(C_p/8 (cos Fi))
(Rd (* Rc (/ 4 2)))
(Rex (* Rc (cos Fi)))
(Dc (* Rc (sin Fi)))
(ep (/ Rex 12))
(epf (* ep (/ 3 5)))
(Rci (- Rc (/ Rc 8)))
(Rce (+ Rc (/ Rc 8)))
(Dae (* Rce (cos Fi)))
(Dai (* Rci (cos Fi)))
(Yf (* Dae Ta (/ 5 2)))
(Y (+ epf Yf))
(Rin (- Rex ep))
(De (+ Dc ep))
(ep_arca (/ (+ Dae Dai) 2))
(Rarca_e (* Ta Dai))
(Box (list (vector (* Rd C_p/8) (- Y) (* Rd S_p/8) 1)
(vector
(vector (/ (* Rd C_p/8) 2) (- (+ Y (/ ep 2))) 0 1)
(vector 1 0 0 0)
(vector 0 1 0 0)
(vector 0 0 1 0))))))

;------(les trois quadriques qui construisent la base de
la coupole) (primitive (DLatt
(SDcolRGB (vector 255 255 180 255))
(DLint
(DLformz2 (vector
(vector 0 Y 0 -1)
(vector 0 (+ Y ep) 0 -1)
(vector 1 0 0 0)
(vector 0 0 1 0)
(vector 1 0 0 0)
(vector 0 0 1 0))))))

```

```

(DLformz2 (vector
(vector (* Rd C_p/8) 0 0 1)
(vector 0 0 0 1)
(vector 0 1 0 0)
(vector 0 0 1 0)
(vector 0 1 0 0)
(vector 0 0 1 0)))

(DLformz2 (vector
(vector (* Rd C_p/8) 0 (* Rd C_p/8 T9) 1)
(vector (* Rd C_p/8) 0 (- (* Rd C_p/8 T9)) 1)
(vector 0 0 0 1)
(vector 0 1 0 0)
(vector 0 0 0 1)
(vector 0 1 0 0))) )))
;;------(Box)
;;(DLuni (apply EXmechxa (append Box (list (vector
1 10))))))
(DLmmx (DLformz2 (apply SGqdrhxa Box)
primitive);;)))
;;=====FIN=====

```

A-C3- algorithme musa.scm

Objectif: générer les éléments radiaux de la base de la
(voir image: page-v, annexe-coupole, figure-c.4)

;;=====corps de la fonction=====

```

(define musaf
(lambda (T Rc)
(let* ((Fi (/ (SGcst_pi) T))
(F9 (/ (SGcst_pi) (+ T 1)))
(T9 (tan F9))
(Ta (tan Fi))
(S_p/8 (sin Fi))
(C_p/8 (cos Fi))
(S_p/9 (sin F9))
(C_p/9 (cos F9))
(Rd (* Rc (/ 4 2)))
(epd (/ Rc 200))
(Rex (* Rc (cos Fi)))
(Dc (* Rc (sin Fi)))
(ep (/ Rex 12))
(epf (* ep (/ 3 5)))
(Rci (- Rc (/ Rc 8)))
(Rce (+ Rc (/ Rc 8)))
(Dae (* Rce (cos Fi)))
(Dai (* Rci (cos Fi)))
(Yf (* Dae Ta (/ 5 2)))
(Y (+ epf Yf))
(Rin (- Rex ep))
(De (+ Dc ep))
(ep_arca (/ (+ Dae Dai) 2))
(Rarca_e (* Ta Dai))

```

```

(Pls (SGx_zx3z
(vector 0 0 0 1)
(vector 0 Rex 0 1)
(vector (* Rd C_p/8) 0 (* Rd S_p/8) 1)))

(Box (list (vector Rd (- Y) 0 1)
(vector
(vector (/ Rd 2) (- (+ Y (/ ep 2))) (- (* Rd T9)) 1)
(vector 1 0 0 0)
(vector 0 1 0 0)
(vector 0 0 1 0))))))
(P0 (vector 0 0 0 1))
(Px (vector 1 0 0 0))
(Py (vector 0 1 0 0))
(Pz (vector 0 0 1 0))

;------(les quadriques de la base)
(primitive
(DLlatt
(SDcolRGB (vector 150 255 100 255))
(Dlint

```

```

(DLformz2
(vector
(vector 0 Y 0 -1)
(vector 0 (+ Y ep) 0 -1)
(vector 1 0 0 0)
(vector 0 0 1 0)
(vector 1 0 0 0)
(vector 0 0 1 0)))

```

```

(DLformz2
(vector
(vector (* Rd C_p/8) 0 0 1)
(vector 0 0 0 1)
(vector 0 1 0 0)
(vector 0 0 1 0)
(vector 0 1 0 0)
(vector 0 0 1 0)))

```

```

(DLformz2
(vector
(vector (* Rd C_p/8) 0 (* Rd S_p/8) 1)
(vector (* Rd C_p/9) 0 (* Rd S_p/9) 1)
(vector 0 0 0 1)
(vector 0 1 0 0)
(vector 0 0 0 1)
(vector 0 1 0 0))))))

```

```

;;------(Box)
(DLuni
primitive (Dlatt (SDmatrep (SGmatismz Pls))
primitive ))))
;;=====FIN=====

```

A-C4- algorithme (fonf.scm)

Objectif: générer la base du colonne
(voir image: page-vi, annexe-coupole, figure-c.5)

```

;=====corps de la fonction=====
(define fondf
  (lambda (T Rc)
    (let* ( (Fi (/ (SGcst_pi) T))
            (Ta (tan Fi))
            (Rex (* Rc (cos Fi)))
            (Dc (* Rc (sin Fi)))
            (ep (/ Rex 12))
            (epf (* ep (/ 3 5)))
            (De (+ Dc ep))
            (Rci (- Rc (/ Rc 8)))
            (Rce (+ Rc (/ Rc 8)))
            (Dae (* Rce (cos Fi)))
            (Dai (* Rci (cos Fi)))

            (Box (list (vector (* epf 2) 0 (* epf 2) Ta) 1)
                  (vector
                   (vector epf (- (/ epf 2)) 0 1)
                   (vector 1 0 0 0)
                   (vector 0 1 0 0)
                   (vector 0 0 1 0))))

            ;;------(les trois quadriques de la
            base)
            (primitive (Dlpmi (DLint
                              (DLformz2 (vector
                                         (vector 0 0 0 1)
                                         (vector 0 epf 0 -1)
                                         (vector 1 0 0 0)
                                         (vector 0 0 1 0)
                                         (vector 1 0 0 0)
                                         (vector 0 0 1 0)))

                              (DLformz2 (vector
                                         (vector (* epf 2) 0 0 1)
                                         (vector 0 0 0 1)
                                         (vector 0 1 0 0)
                                         (vector 0 0 1 0)
                                         (vector 0 1 0 0)
                                         (vector 0 0 1 0)))

                              (DLformz2 (vector
                                         (vector (* epf 2) 0 (* epf 2) Ta) 1)
                                         (vector (* epf 2) 0 (- (* epf 2) Ta) 1)
                                         (vector 0 0 0 1)
                                         (vector 0 1 0 0)
                                         (vector 0 0 0 1)
                                         (vector 0 1 0 0))))

            (DLmmx (DLformz2 (apply SGqdrhxa Box)
                              primitive) ;; )
            )))
;=====FIN=====

```

```

(GDcylseg (vector (/ epf 2) 1) ;un quad de cylindre
(vector (* epf 2) 0 (* epf 2) 1)
(vector (* epf 2) 0 (- (* epf 2) 1)))) )
;------(Box- min max)
;;(DLuni (apply EXmechxa (append Box (list (vector
1 20))))
(DLmmx (DLformz2 (apply SGqdrhxa Box)
primitive) ;; )
)))
;=====FIN=====

```

A-C5- algorithme (Trans.scm)

Objectif: la fonction de la translation permet de transférer l'ensemble de la base de la colonne à sa place au-dessous de la colonne.

```

(Ploadr "repere.scm")
(Ploadr "rotate.scm")
(Ploadr "fond.scm")
(Ploadr "rot_trf.scm")
(Ploadr "cornich.scm")

;=====-(fonction de translation)=====
(define Transf
  (lambda (T Rc)
    (let* ( (Fi (/ (SGcst_pi) T))
            (Ta (tan Fi))
            (Rex (* Rc (cos Fi)))
            (Dc (* Rc (sin Fi)))
            (ep (/ Rex 12))
            (epf (* ep (/ 3 5)))
            (De (+ Dc ep))
            (Rci (- Rc (/ Rc 8)))
            (Rce (+ Rc (/ Rc 8)))
            (Dae (* Rce (cos Fi)))
            (Dai (* Rci (cos Fi)))

            ;;-----Matrices de translation
            (DLuni
            (DLatt
            (SDmatrep (SGmattrl (vector (+ Dai (/ (- Dae
Dai) 2))
(- (* Dae Ta (/ 5 2)))
(* (+ Dai (/ (- Dae Dai) 2)) Ta) 1)))
(fondf T Rc))
            (Dlatt (SDmatrep (SGmattrl (vector (+ Dai (/
(- Dae Dai) 2))
(- (* Dae Ta (/ 5 2)))
(* (+ Dai (/ (- Dae Dai) 2)) Ta) 1)))
(cornichf T Rc))
            )))
;=====FIN=====

```


A-C6-L'algorithmme (colonne.scm)

Objectif: générer différentes types de colonnes.
(voir image: page-viii, annexe-coupole, figure-c.8)

```

;=====corps de la fonction=====
(define colonnef
  (lambda (T Rc)
    (let* (
      (Fi (/ (SGcst_pi) T))
      (Ta (tan Fi))
      (Rex (* Rc (cos Fi)))
      (Dc (* Rc (sin Fi)))
      (ep (/ Rex 12))
      (Rin (- Rex ep))
      (De (+ Dc ep))
      (Rci (- Rc (/ Rc 8)))
      (Rce (+ Rc (/ Rc 8)))
      (Dae (* Rce (cos Fi)))
      (Dai (* Rci (cos Fi)))
      (Box (list (vector (/ ep 2) (- (* Dae Ta)) (/ ep 2) 1)
        (vector
          (vector 0 (- (* Dae Ta (/ 7 4))) 0 1)
          (vector 1 0 0 0)
          (vector 0 1 0 0)
          (vector 0 0 1 0))))))
      ;;------(primitive)
      (primitive (DLint
        (DLformz2 (vector
          (vector (/ ep 2) (+ (* Dae Ta) (/ (* Dae Ta) 2)) 0 -1)
          (vector (/ ep 2) (+ (* Dae Ta) (/ (* Dae Ta) 2)) 0 1)
          (vector 0 1 0 0)
          (vector 0 0 1 0)
          (vector 0 (/ ep 2) 0 0)
          (vector 0 0 (/ ep 2) 1)))
        (DLformz2
          (vector
            (vector 0 (* Dae Ta) 0 -1)
            (vector 0 (* Dae Ta (/ 5 2)) 0 -1)
            (vector 1 0 0 0)
            (vector 0 0 1 0)
            (vector 1 0 0 0)
            (vector 0 0 1 0))))))
      ;;------(translation)
      (DLatt (SDmatrep (SGmattrl (vector (+ Dai (/ (-
        Dae Dai) 2) 0
        (* (+ Dai (/ (- Dae Dai) 2)) Ta) 1))))
      ;;------(Box)
      ;;(DLuni (apply EXmechxa (append Box (list (vector 1
        6))))
      (DLmmx (DLformz2 (apply SGqdrhxa Box))
        primitive))
      ;; )
      )))
;=====FIN=====

```

A-C7-Algorithmme (prism.scm)

Objectif: générer la base du dôme entrelacé
(voir image: page-x, annexe-coupole, figure-c.12)

```

;=====corps de la fonction=====
(define prismf
  (lambda (T Rc)
    (let* (
      (Fi (/ (SGcst_pi) T))
      (Ta (tan Fi))
      (Rex (* Rc (cos Fi)))
      (Dc (* Rc (sin Fi)))
      (ep (/ Rex 12))
      (Rin (- Rex ep))
      (De (+ Dc ep))
      (Rci (- Rc (/ Rc 8)))
      (Rce (+ Rc (/ Rc 8)))
      (Dae (* Rce (cos Fi)))
      (Dai (* Rci (cos Fi)))
      (ep_arca (/ (+ Dae Dai) 2))
      (Rarca_e (* Ta Dai))
      (Rarca_i (- Rarca_e (* (/ 2 3) ep_arca)))
      (Box (list (vector Dae 0 (* Dae Ta) 1)
        (vector
          (vector (+ Dai (/ (- Dae Dai) 2)) (- (/ (* Dae Ta) 2))
            0 1)
          (vector 1 0 0 0)
          (vector 0 1 0 0)
          (vector 0 0 1 0))))))
      ;;------(les quadriques de la base)
      (primitive (Dlint (DLformz2 (vector
        (vector 0 0 0 1)
        (vector 0 (* Dae Ta) 0 -1)
        (vector 1 0 0 0)
        (vector 0 0 1 0)
        (vector 1 0 0 0)
        (vector 0 0 1 0))))
      (DLformz2 (vector
        (vector Dai 0 0 1)
        (vector Dae 0 0 1)
        (vector 0 1 0 0)
        (vector 0 0 1 0)
        (vector 1 0 0 0)
        (vector 0 1 0 0)
        (vector 0 0 1 0))))
      (DLformz2 (vector
        (vector Dae 0 (* Dae Ta) 1)
        (vector Dae 0 (- (* Dae Ta) 1)
          (vector 0 0 0 1)
          (vector 0 1 0 0)
          (vector 0 0 0 1)
          (vector 0 1 0 0))))))

```

```

(prisme2
;;------(pmi .. cylindre et cone)
(DLatt (SDcolRGB (vector 170 255 170 255))
(DLpmi primitive
(GDcylseg
(vector Rarca_e 1)
(vector 0 (- (* Dae Ta)) 0 1)
(vector Rce (- (* Dae Ta)) 0 1))
(GDconseg
(vector (* (+ Rce (/ Rce 70)) (sin Fi)) 1)
(vector (* (+ Rce (/ Rce 70)) (cos Fi)) (- (* Dae
Ta)) 0 1)
(vector Dai (- (* Dae Ta)) 0 1))))))
;;------(Box)
;;(DLuni (apply EXmechxa (append Box (list
(vector 1 4))))
(DLmmx (DLformz2 (apply SGqdrhxa Box)
prisme2)
;;)
)))
;;=====FIN=====

```

A-C8-Algorithm (arcade.scm)

Objectif: générer l'arc qui forme l'arcade de la base du dôme entrelacé. (voir image: page-x, annexe-coupole, figure-c.12)

```

;;=====corps de la fonction=====
(define arcadf
(lambda (T Rc)
(let* ( (Fi (/ (SGcst_pi) T))
(Ta (tan Fi))
(Rex (* Rc (cos Fi)))
(Dc (* Rc (sin Fi)))
(ep (/ Rex 12))
(Rin (- Rex ep))
(De (+ Dc ep))
(Dae (* (+ Rc (/ Rc 8)) (cos Fi)))
(Dai (* (- Rc (/ Rc 8)) (cos Fi)))
(ep_arca (/ (- Dae Dai) 2))
(Rarca_e (* Ta Dai))
(Rarca_i (- Rarca_e (* (/ 2 3) ep_arca)))
(Pls (SGx_zx3z
(vector 0 0 0 1)
(vector 0 Rex 0 1)
(vector Rex 0 (- Dc) 1)))

```

```

(Box (list (vector (+ Dai (/ (- Dae Dai) 4)) (-Rarca_e
(* Dae Ta) (* (+ Dai (/ (- Dae Dai) 4)) Ta) 1)
(vector
(vector (+ Dai (/ (- Dae Dai) 2)) (- (/ Rarca_e 2) (*
Dae Ta)) 0 1)
(vector 1 0 0 0)
(vector 0 1 0 0)
(vector 0 0 1 0))))))

```

```

;;------(structure objet)
(primitive (DLatt (SDcolRGB (vector 200 25 30
255))
(DLpmi(GDcylseg
(vector Rarca_e 1)
(vector (+ Dai (/ (- Dae Dai) 4)) (- (* Dae Ta)) 0 1)
(vector (- Dae (/ (- Dae Dai) 4)) (- (* Dae Ta)) 0 1))

```

```

(GDcylseg
(vector Rarca_i 1)
(vector 0 (- (* Dae Ta)) 0 1)
(vector (* Rc 2) (- (* Dae Ta)) 0 1))

```

```

(DLformz2 (vector
(vector 0 (* Dae Ta) 0 -1)
(vector 0 Rc 0 -1)
(vector 1 0 0 0)
(vector 0 0 1 0)
(vector 1 0 0 0)
(vector 0 0 1 0))))))

```

```

;;------(Box)
;;(DLuni
;;(apply EXmechxa (append Box (list (vector 1 4))))
(DLmmx (DLformz2 (apply SGqdrhxa Box)
primitive)
;;)
)))
;;=====FIN=====

```

A-C9-Algorithm (hyper.scm)

Objectif: générer la partie postérieure du chapiteau
(voir image: page-xi, annexe-coupole, figure-c.13)

```
;;=====corps de la fonction=====
```

```
(define hypertf
(lambda (T Rc)
(let* ( (Fi (/ (SGcst_pi) T))
(Ta (tan Fi))
(Rex (* Rc (cos Fi)))
(Dc (* Rc (sin Fi)))
(ep (/ Rex 12))
(Rep (+ Rex ep))
(Rin (- Rex ep))
(De (+ Dc ep))
(P0 (vector 0 0 0 1))
(Px (vector 1 0 0 0))
(Py (vector 0 1 0 0))
(Pz (vector 0 0 1 0))

(Box (list (vector (+ Dc (* ep 2)) Rep (* (+ Dc (*
ep 2)) Ta) 1)
(vector
(vector (+ Dc ep) (+ Rep ep) 0 1)
(vector 1 0 0 0)
(vector 0 1 0 0)
(vector 0 0 1 0))))))
```

```
(Quad_hyper
(DLformz2
(vector
(vector Dc Rep 0 1)
(vector De Rep 0 1)
(vector 0 1 0 0)
(vector 0 0 1 0)
(vector (+ Dc (* ep 6)) (+ Rep (/ ep 2)) 0 0)
(vector 0 Rep (/ ep 2) 0))))
```

```
(Quad_xz (DLformz2 (vector
(vector 0 (+ Rex ep) 0 1)
(vector 0 (+ Rex (* ep 3)) 0 1)
(vector 1 0 0 0)
(vector 0 0 1 0)
(vector 1 0 0 0)
(vector 0 0 1 0))))
```

```
(Quad_zy (DLformz2 (vector
(vector De 0 (* De Ta) 1)
(vector De 0 (- (* De Ta)) 1)
(vector 0 0 0 1)
(vector 0 1 0 0)
(vector 0 0 0 1)
(vector 0 1 0 0))))
```

```
(Quad_cut (DLformz2 (vector
(vector Dc 0 0 1)
(vector Dc 0 0 -1)
(vector 0 1 0 0)
(vector 0 0 1 0)
(vector 0 1 0 0)
(vector 0 0 1 0))))
```

```
(Cruse (GDcylseg
(vector (* ep 2) 1)
(vector (+ De (* 2 ep)) 0 0 1)
(vector (+ De (* 2 ep)) (* 2 Rex) 0 1)))
```

```
;;------(les quadriques de la
base)
```

```
(primitive (DLatt (SDcolRGB (vector 255 200
255 255))
(DLpmi (DLint Quad_hyper Quad_xz
Quad_zy) Cruse Quad_cut))))
```

```
;;------(Box
```

```
; (DLuni (apply EXmechxa (append Box (list
(vector 1 4))))
(DLmmx (DLformz2 (apply SGqdrhxa Box))
primitive ) ;
)))
```

```
;;=====FIN=====
```

A-C10-Algorithm (p_base.scm)

Objectif: générer la partie inférieure du
chapiteau. (voir image: page-xii, annexe-
coupole, figure-c.15)

```
;;=====corps de la fonction=====
```

```
(define p_basef
(lambda (T Rc)
(let* ( (Fi (/ (SGcst_pi) T))
(Ta (tan Fi))
(Rex (* Rc (cos Fi)))
(Dc (* Rc (sin Fi)))
(ep (/ Rex 12))
(Rin (- Rex ep))
(De (+ Dc ep))
(P0 (vector 0 0 0 1))
(Px (vector 1 0 0 0))
(Py (vector 0 1 0 0))
(Pz (vector 0 0 1 0))
```

```

(Box (list (vector De (+ Rex ep) (* De Ta) 1)
(vector
(vector (+ Dc (/ ep 2)) (+ Rex (/ ep 3)) 0 1)
(vector 1 0 0 0)
(vector 0 1 0 0)
(vector 0 0 1 0))))

;;------(les quadriques de
la base)
(primitive
(Dlatt (SDcolRGB (vector 255 200 255
255))
(Dlpmi (Dlint

(DLformz2 (vector
(vector 0 (+ Rex ep) 0 1)
(vector 0 (- Rex (* ep 2)) 0 1)
(vector 1 0 0 0)
(vector 0 0 1 0)
(vector 1 0 0 0)
(vector 0 0 1 0)))

(DLformz2 (vector
(vector Dc 0 0 1)
(vector De 0 0 1)
(vector 0 1 0 0)
(vector 0 0 1 0)
(vector 0 1 0 0)
(vector 0 0 1 0)))

(DLformz2 (vector
(vector De 0 (* De Ta) 1)
(vector De 0 (- (* De Ta)) 1)
(vector 0 0 0 1)
(vector 0 1 0 0)
(vector 0 0 0 1)
(vector 0 1 0 0)))

(GDcylseg
(vector Rex 1)
(vector 0 0 0 1)
(vector Rc 0 0 1))))))
;;------(Box

;;(DLuni (apply EXmechxa (append Box (list
(vector 1 12))))
(DLmmx (DLformz2 (apply SGqdrhxa
Box)) primitive) ;)
)))
;;=====FIN=====

A-C11-Algorithmes (arcpri-scm)
Objectif: générer l'arc de base
(voir image: page-xiii, annexe-coupole,
figure-c.17)
;;=====corps de la fonction=====
(define arcpf
(lambda (T Rc)
(let* ( (Fi (/ (SGcst_pi) T))
(Rex (* Rc (cos Fi)))
(Dc (* Rc (sin Fi)))
(ep (/ Rex 12))
(De (+ Dc ep))
(X (* De (tan Fi)))
(Pls (SGx_zx3z
(vector 0 0 0 1)
(vector 0 Rex 0 1)
(vector Rex 0 (- Dc) 1)))
(Box (list (vector De (+ Rex ep) (- X) 1)
(vector
(vector (/ De 2) (+ Rex ep (/ De 2)) (- (/ X 2))
1)
(vector 1 0 0 0)
(vector 0 1 0 0)
(vector 0 0 1 0))))
(P0 (vector 0 0 0 1))
(Px (vector 1 0 0 0))
(Py (vector 0 1 0 0))
(Pz (vector 0 0 1 0))
;;------(structure objet)
(primitive (Dlpmi (DLatt
(SDmatrep (SGmattrl (vector 0 (+ Rex ep) (-
De) 1)))
(Dlpmi (GDqcyhxa
(vector De De De 1)
(vector (vector De De De 2) Px Py Pz))

(Gdcylseg
(vector Dc 1)
(vector 0 0 (- (* 2 Dc)) 1)
(vector 0 0 (* 2 Dc) 1) ))

;;-----(2 plans verticaux sur xz couper le quart
du cylindre)
(Dlatt (SDmatrep (SGmatrot (vector
(SGcst_pi) 12)
(vector 0 0 0 1) (vector 0 1 0 1)))

(DLformz2
(vector P0 (vector 0 0 Rex 1) Py Px Py Px)) )
(DLatt (SDmatrep (SGmatrot (vector
(SGcst_pi) 8)
(vector 0 0 0 1) (vector 0 1 0 1)))

(DLformz2 (vector P0 (vector 0 0 (- Rex) 1)
Py Px Py Px )) )

```

```

;---(quadrique pour couper la tete de l'arc)
(DLformz2 (vector
(vector (- (/ De 3))(+ (* (/ 3 2) ep)(- De ep) Rex) 0 1)
(vector (/ De 3)(+ (* (/ 3 2) ep) (- De ep) Rex) 0 1)
(vector 0 1 0 0)
(vector 0 0 1 0)
(vector 0 (+ (/ De 3) ep (- De ep) Rex) 0 1)
(vector 0 (+ ep (- De ep) Rex) (/ De 3) 1))) )))

;------(Box)
(DLatt (SDmatrep (SGmatt1 (vector 0 (* 2 ep) 0
1)))
(DLuni
;; (DLuni
;; (apply EXmechxa (append Box (list (vector 1 4))))

(DLmmx (DLformz2 (apply SGqdrhxa Box)
primitive)
;; )
;------(matrice symetrique)
(DLuni primitive (DLatt (SDmatrep (SGmatmz
Pls)) primitive )))
)))
;=====FIN=====

```

A-C12-algorithme de la remplissage (spa.scm)

Objectif: le remplissage du dôme aux arcs radiaux
(voir image: page-xiv, annexe-coupoles, figure-c.18)

```

;=====corps de la fonction=====
define spa
(lambda (T Rc)
(let* (
(Fi (/ (SGcst_pi) T))
(Fd (/ (SGcst_pi) 12))
(T12 (tan Fd))
(Rex (* Rc (cos Fi)))
(Dc (* Rc (sin Fi)))
(ep (/ Rex 12))
(Dcp (+ Dc (/ ep 2)))
(De (+ Dc ep))
(X (* De (tan Fi)))

(Pls (SGx_zx3z
(vector 0 0 0 1)
(vector 0 Rex 0 1)
(vector Rex 0 (- Dc) 1)))

```

```

(Box (list (vector Dcp (+ Rex ep) (* Dcp T12) 1)
(vector
(vector (* (/ 1 2) De) (+ Rex ep (/ De 2)) 0 1)
(vector 1 0 0 0)
(vector 0 1 0 0)
(vector 0 0 1 0))))

(P0 (vector 0 0 0 1))
(Px (vector 1 0 0 0))
(Py (vector 0 1 0 0))
(Pz (vector 0 0 1 0))

(Qcy (GDqcyhxa
(vector Dcp (+ Rex ep Dcp) (* Dcp T12) 1)
(vector (vector (/ Dcp 2) (+ Rex ep (/ Dcp 2)) 0 1)
Px Py Pz)))

;------(structure objet)
(primitive
(Dlatt (SDcolRGB (vector 80 140 35 255))
(Dlint (DLuni (Dlpmi Qcy (GDcylseg
(vector (+ Dc (/ ep 4)) 1)
(vector 0 (+ Rex ep) Rc 1)
(vector 0 (+ Rex ep) (- Rc) 1))

(DLformz2
(vector
(vector (- (/ De 3)) (+ (* (/ 3 2) ep) (- De ep) Rex) 0
1)
(vector (/ De 3) (+ (* (/ 3 2) ep) (- De ep) Rex) 0 1)
(vector 0 1 0 0)
(vector 0 0 1 0)
(vector 0 (+ (/ De 3) ep (- De ep) Rex) 0 1)
(vector 0 (+ ep (- De ep) Rex) (/ De 3) 1))))))

(DLformz2
(vector
(vector Dcp 0 (* Dcp T12) 1)
(vector Dcp 0 (- (* Dcp T12)) 1)
(vector 0 0 0 1)
(vector 0 1 0 0)
(vector 0 0 0 1)
(vector 0 1 0 0))) )))

(DLatt
(SDmatrep (SGmatt1 (vector 0 (* ep 2) 0 1)))
;;(DLuni (apply EXmechxa (append Box (list (vector
1 4))))
(DLmmx (DLformz2 (apply SGqdrhxa Box)
primitive))
;; )
)))
;=====FIN=====

```

A-C13-Algorithmme sp.scm

Objectif: générer le remplissage du dôme entrelacé,
partie-1

(voir image: page-xv, annexe-coupole, figure-c.19.c)

;;=====corps de la fonction=====

(Ploadr "repere.scm")

(Ploadr "rotate.scm")

(Ploadr "spt.scm")

(Ploadr "sptt.scm")

;;=====

la peau)

(define sp

(lambda (T Rc)

(let* (

(Fi (/ (SGcst_pi) 8))

(Rex (* Rc (cos Fi)))

(Dc (* Rc (sin Fi)))

(ep (/ Rex 12))

(De (+ Dc ep)

(Rin (- Rex ep))

(Box (list (vector De 0 De 1)

(vector

(vector (+ (/ (- Rex De) 2) De) (* Rex (/ 2 5)) (+ (/ (-

Rex De) 2) De) 1)

vector 1 0 0 0)

(vector 0 1 0 0)

(vector 0 0 1 0))))

;;------(structures objet)

(ph_ex (GDsphdis

(vector Rc 1)

(vector 0 0 0 1))))

(p_2 (DLformz2 (vector

(vector De 0 0 1)

(vector (- Rc) 0 0 1)

(vector 0 1 0 0)

(vector 0 0 1 0)

(vector 0 1 0 0)

(vector 0 0 1 0))))

(p_1 (DLformz2 (vector

(vector 0 0 De 1)

(vector 0 0 (- Rc) 1)

(vector 0 1 0 0)

(vector 1 0 0 0)

(vector 0 1 0 0)

(vector 1 0 0 0))))

(ph_in (GDsphdis

(vector (- Rc (/ ep 4)) 1)

(vector 0 0 0 1))))

(q_h (DLformz2 (vector

(vector 0 Rc 0 -1)

(vector 0 0 0 1)

(vector 1 0 0 0)

(vector 0 0 1 0)

(vector 1 0 0 0)

(vector 0 0 1 0))))

(objet (DLpmi ph_ex ph_in p_1 p_2 q_h))

(primitive (DLatt

(SDcolRGB (vector 255 150 0 255)) objet))

;;------(Box)

(final

;; (DLuni (apply EXmechxa (append Box (list

(vector 1 4))))

(DLmmx

(DLformz2 (apply SGqdrhxa Box)) primitive) ;;)

))

final

)))

;;-----FIN-----

A-C14-Algorithmme spt.scm

Objectif: générer le remplissage du dôme
entrelacé, partie-2. (voir image: page-xv,
annexe-coupole, figure-c.19.c)

```
(Ploadr "repere.scm")
```

```
(Ploadr "rotate.scm")
```

```
;;=====corps de la fonction=====
```

```
(define spt
```

```
(lambda (T Rc)
```

```
(let* (
```

```
(Fi (/ (SGcst_pi) 8))
```

```
(Fiq (/ (SGcst_pi) 4))
```

```
(Rex (* Rc (cos Fi)))
```

```
(Dc (* Rc (sin Fi)))
```

```
(ep (/ Rex 12))
```

```
(De (+ Dc ep))
```

```
(Rin (- Rex ep))
```

```
(Box (list (vector De 0 Dc 1)
```

```
(vector
```

```
(vector (+ (/ (- Rex De) 2) De) (/ Rex 2) (/ (+
```

```
Dc (/ ep 2)) 2) 1)
```

```
(vector 1 0 0 0)
```

```
(vector 0 1 0 0)
```

```
(vector 0 0 1 0))))
```

```
(Quad11 (DLformz2 (vector
```

```
(vector De 0 0 1)
```

```
(vector (- Rc) 0 0 1)
```

```
(vector 0 1 0 0)
```

```
(vector 0 0 1 0)
```

```
(vector 0 1 0 0)
```

```
(vector 0 0 1 0))))
```

```
(Quad1 (Dlatt
```

```
(SDmatrep (SGmatrot (vector Fiq -1)
```

```
(vector 0 0 0 1)
```

```
(vector 0 1 0 0))) Quad11))
```

```
(Quad22 (DLformz2 (vector
```

```
(vector 0 0 Dc 1)
```

```
(vector 0 0 Rc 1)
```

```
(vector 0 1 0 0)
```

```
(vector 1 0 0 0)
```

```
(vector 0 1 0 0)
```

```
(vector 1 0 0 0))))
```

```
(Quad2 (Dlatt
```

```
(SDmatrep (SGmatrot (vector (* (SGcst_pi) (/
```

```
3 4)) 1)
```

```
(vector 0 0 0 1)
```

```
(vector 0 1 0 0))) Quad22))
```

```
(ph_ex (Gdsphdis
```

```
(vector Rc 1)
```

```
(vector 0 0 0 1)))
```

```
(ph_in (GDsphdis
```

```
(vector (- Rc (/ ep 4)) 1)
```

```
(vector 0 0 0 1)))
```

```
(q_h (DLformz2 (vector
```

```
(vector 0 Rc 0 -1)
```

```
(vector 0 0 0 1)
```

```
(vector 1 0 0 0)
```

```
(vector 0 0 1 0)
```

```
(vector 1 0 0 0)
```

```
(vector 0 0 1 0))))
```

```
(objet (DLpmi ph_ex ph_in q_h Quad11
```

```
Quad22 Quad1 Quad2))
```

```
(primitive (DLatt (SDcolRGB (vector
```

```
110 110 35 255)) objet))
```

```
;;------(Box)
```

```
(final ;;(DLuni (apply EXmechxa
```

```
(append Box (list (vector 1 4))))
```

```
(DLmmx (DLformz2 (apply SGqdrhxa
```

```
Box)) primitive) ;;)
```

```
))
```

```
final)))
```

```
;;=====FIN=====
```

A-C15-algorithme sptt.scm

Objectif: générer le remplissage du dôme
entrelacé, partie-3. (voir image: page-xv,
annexe-coupole, figure-c.19.c)

```
(Ploadr "repere.scm")
(Ploadr "rotate.scm")
;=====corps de la fonction=====
(define sptt
(lambda (T Rc)
(let* (
(Fi (/ (SGcst_pi) 8))
(Ta (tan Fi))
(Fiq (/ (SGcst_pi) 4))
(Rex (* Rc (cos Fi)))
(Dc (* Rc (sin Fi)))
(ep (/ Rex 12))
(De (+ Dc ep))
(Rin (- Rex ep))
(Box (list (vector Dc (* Rex (/ 5 2)) (* Dc Ta)
1)
(vector
(vector Dc (+ (* Rex (/ 5 2)) (/ Dc 2)) 0 1)
(vector 1 0 0 0)
(vector 0 1 0 0)
(vector 0 0 1 0))))))
(Quad (DLformz2 (vector
(vector De 0 0 1)
(vector Rc 0 0 1)
(vector 0 1 0 0)
(vector 0 0 1 0)
(vector 0 1 0 0)
(vector 0 0 1 0))))))
(Quad1 (DLatt
(SDmatrep (SGmatrot (vector Fiq -1)
(vector 0 0 0 1)
(vector 0 1 0 0))) Quad))
(Quad2 (Dlatt
(SDmatrep (SGmatrot (vector Fiq 1)
(vector 0 0 0 1)
(vector 0 1 0 0))) Quad))
(ph_ex (GDsphdis
(vector Rc 1)
(vector 0 0 0 1)))
(p_2 (DLformz2 (vector
(vector De 0 0 1)
(vector (- Rc) 0 0 1)
(vector 0 1 0 0)
(vector 0 0 1 0)
(vector 0 1 0 0)
(vector 0 0 1 0))))))
```

```
(ph_in (GDsphdis
(vector (- Rc (/ ep 4)) 1)
(vector 0 0 0 1)))
(q_h (DLformz2 (vector
(vector 0 Rc 0 -1)
(vector 0 0 0 1)
(vector 1 0 0 0)
(vector 0 0 1 0)
(vector 1 0 0 0)
(vector 0 0 1 0))))
(objet (DLpmi ph_ex ph_in p_2 q_h Quad1
Quad2))
(primitive (DLatt (SDcolRGB (vector 10 110 45
255)) objet))
;=====END=====
```

A-C16-Algorithm (tête.scm)

Objectif: générer l'ellipsoïde qui couvre le dôme
aux arcs radiaux

```
;=====corps de la fonction=====
(define tetef
(lambda (Rc)
(let* (
(Rex (* Rc (cos (/ (SGcst_pi) 8))))
(Dc (* Rc (sin (/ (SGcst_pi) 8))))
(ep (/ Rex 12))
(De (+ Dc ep))
(P0 (vector 0 0 0 1))
(Px (vector 1 0 0 0))
(Py (vector 0 1 0 0))
(Pz (vector 0 0 1 0))
(prim_tete
(DLatt
(SDcolRGB (vector 100 55 50 255))
(DLformz2 (vector
(vector (- (/ De 3)) (+ (* (/ 3 2) ep) (- De ep) Rex)
0 1)
(vector (/ De 3) (+ (* (/ 3 2) ep) (- De ep) Rex) 0
1)
(vector 0 1 0 0)
(vector 0 0 1 0)
(vector 0 (+ (/ De 3) ep) (- De ep) Rex) 0 1)
(vector 0 (+ ep (- De ep) Rex) (/ De 3) 1)
))))))
(Dlatt
(SDmatrep (SGmattrl (vector 0 (* ep 2) 0 1)))
prim_tete)
)))
;=====FIN=====
```


A-C17-Algorithmme (resultat.scm) Structure algorithmique globale (regrouper toutes les fonctions)

```

;;=====corps de la fonction=====
(define glob
(lambda (T Rc) (DLuni
(gen_arcpri T (spa T Rc)) ;le remplissage du dôme aux arcs radiaux
;(gen_arcpri T (sp T Rc))
;(sp , spt , sptt , Trois algorithmes génèrent les surfaces qui
remplissent entre-arcs dans le dôme inter-lassé

;(gen_arcpri T (spt T Rc))
;(gen_arcpri T (sptt T Rc))
;(gen_arcpri T (coupf T Rc)) ;Génère les arcs entrelacés
(gen_arcpri T (p_basef T Rc)) ;Génère la base du dôme radial
(gen_arcpri T (arcpf T Rc)) ;Génère les arcs du dôme radial
(gen_arcpri T (hypertf T Rc)) ;Génère la partie hyperbolique de la base du dôme radial
(gen_arcpri T (prismf T Rc)) ;Génère la partie supérieur de l'arcade
(gen_arcpri T (arcadf T Rc)) ;Génère l'arcade
(gen_arcpri T (gen_trf T T Rc (Transf T Rc)))
;(génère la base de la colonne

(gen_arcpri T (colonnef T Rc)) ;Génère les colonnes
(gen_arcpri T (dallemf T Rc)) ;génère la base de la coupole
(gen_arcpri T (finishf T Rc)) ;Génère le cadre de la base de la coupole
(gen_arcpri T (musaf T Rc)) ;Génère la mosaïque
(tetef Rc ))) ;Génère la tête de la coupole croisée

;;=====fonction de visualisation=====
(define vresultat
(lambda ()
(begin
(PIreset)
(PIswin (vector 640 480))
(PIsobs (vector 0 0 1 0))
(PIstarget (vector 0 0 0 1))
(PIsbox (vector 45 45))
(PIalign
(vector (vector 2 3 1 0) .6 ))
(PIdraw->targa (DLuni (glob 8 10) ;(la fonction regroupe toutes les fonction de la coupole
(gen_arcpri 8 (arc_disf 10 8 (* (SGcst_pi) (/ 1 2)) (* (SGcst_pi) (/ -0 8)) 0 0))
;la ligne de code ci-dessous génère le dôme entrelacé importé de la bibliothèque numérique du
programme (prog_pointu)
)"fa-co-pea.tga« )))
(vresultat)
;;=====FIN=====

```

Exemple:

La fonction (DLuni (glob 8«T» 10«Rc») génère la coupole

La fonction (gen_arcpri 8 (arc_disf 10«Rc» 8 «Na» (* (SGcst_pi) (/ 1 2)) (* (SGcst_pi) (/ -0 8)) 0 0))
génère le dôme entrelacé importé de la bibliothèque numérique du programme (prog_pointu).

Note: le paramètre T=Na et Rc ext le rayon de la coupole.

Dans le cas ou T=8 et Rc=10 (voir l'image au chapitre-iv, page-95, figure-78-a)

T=10 et Rc=10 (voir l'image au chapitre-iv, page-95, figure-78-b)

T=12 et Rc=10 (voir l'image au chapitre-iv, page-95, figure-78-c)

ANNEXE -LISP

PROGRAMME « prog_lisp»: les fonctions de transformations géométriques

ANNEXE -lisp

PROGRAMME « prog_lisp »

Lisp-1

Objectif: générer une surface à partir de deux primitives (cercle se développe sur un autre cercle)

```

;=====corps de la fonction=====
(defun C:FORME-2 () Px Py Pz / Cp Ra Nr Nc
Te Fi

(setq Cp (getpoint "\nCentre de dôme:"))
(setq Ra (getdist Cp "\nRayon de dôme:"))
(setq Nr (getint "\nNombre de ranges:"))
(setq Nc (getint "\nNombre de colonnes:"))
;(setq Lc (list))

(setq Fi 0)
(while
(< Fi (/ pi 2))
(setq Te 0)
(while
(< Te (* pi 2))
(setq Px (* (sin Fi) (cos Te) Ra ))
(setq Py (* (sin Fi) (sin Te) Ra ))
(setq Pz (* (cos Fi) (sin Fi) Ra))
(command "point" (list Px Py Pz))
;(setq Lp (cons (list Px Py Pz) Lp))

(setq Te (+ (/ (* pi 2) Nc) Te))
)
(setq Fi (+ (/ pi (* 2 Nr)) Fi))
)
;(setq Lp (reverse Lp))
)
;=====FIN=====

```

Exemple: lisp-1
 Les paramètres: Cp=0,0,0 Ra=100 Nr=40 Nc=160
 Génère un tour (voir, page-14, figur-10-c)

Lisp-2

Objectif: générer un surface ellipsoïdale

```

;-----
;Cp ;(centre de dôme)
;Ra ;(Rayon de d,ome)
;Nr ;(Nombre de ranges)
;Nc ;(Nombre des colonnes)
;Te ;(Angle horizontale)
;Fi ;(Angle verticale)
;Px Py Pz ;(coordonnées de point)
;=====corps de la fonction=====
(defun C:FORME-1 () Px Py Pz / Cp Ra Nr
Nc Te Fi

(setq Cp (getpoint "\nCentre de dôme:"))
(setq Ra (getdist Cp "\nRayon de dôme:"))
(setq Nr (getint "\nNombre des ranges:"))
(setq Nc (getint "\nNombre des colonnes:"))
;(setq Lp (list))

(setq Fi 0)
(while
(< Fi (/ pi 2))
(setq Te 0)
(while
(< Te (* pi 2))
(setq Px (* (sin (* 2 Fi)) (cos Te) Ra ))
(setq Py (* (sin Fi) (sin Te) Ra ))
(setq Pz (* (cos Fi) Ra))
(command "point" (list Px Py Pz))
;(setq Lp (cons (list Px Py Pz) Lp))

(setq Te (+ (/ (* pi 2) Nc) Te))
)
(setq Fi (+ (/ pi (* 2 Nr)) Fi))
)
;(setq Lp (reverse Lp))
)
;=====FIN=====

```

Exemple: lisp-2

Un ellipsoïde commence du point 0,0,z et se développe en changeant l'orientation de son axe de X à Y. Les paramètres :Cp=0,0,0 Ra=100 Nr=100 Nc=200.
 (voir, page-14, figure-10-b)

